Orthographic support for passing the reading hurdle in Japanese

A thesis presented
by

Lars Yencken

to
The Department of Computer Science and Software Engineering
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

University of Melbourne
Melbourne, Australia
April 2010

Thesis advisor(s)
Timothy Baldwin

Author
Lars Yencken

Orthographic support for passing the reading hurdle in Japanese

Abstract

Learning a second language is, for the most part, a day-in day-out struggle against the mountain of new vocabulary a learner must acquire. Furthermore, since the number of new words to learn is so great, learners must acquire them autonomously. Evidence suggests that for languages with writing systems, native-like vocabulary sizes are only developed through reading widely, and that reading is only fruitful once learners have acquired the core vocabulary required for it to become smooth. Learners of Japanese have an especially high barrier in the form of the Japanese writing system, in particular its use of kanji characters. Recent work on dictionary accessibility has focused on compensating for learner errors in pronouncing unknown words, however much difficulty remains.

This thesis uses the rich visual nature of the Japanese orthography to support the study of vocabulary in several ways. Firstly, it proposes a range of kanji similarity measures and evaluates them over several new data sets, finding that the stroke edit distance and tree edit distance metrics best approximate human judgements. Secondly, it uses stroke edit distance construct a model of kanji misrecognition, which we use as the basis for a new form of kanji search by similarity. Analysing query logs, we find that this new form of search was rapidly adopted by users, indicating its utility. We finally combine kanji confusion and pronunciation models into a new adaptive testing platform, Kanji Tester, modelled after aspects of the Japanese Language Proficiency Test. As the user tests themselves, the system adapts to their error patterns and uses this information to make future tests more difficult. Investigating logs of use, we find a weak positive correlation between ability estimates and time the system has been used. Furthermore, our adaptive models generated questions which were significantly more difficult than their control counterparts.

Overall, these contributions make a concerted effort to improve tools for learner self-study, so that learners can successfully overcome the reading hurdle and propel themselves towards greater proficiency. The data collected from these tools also forms a useful basis for further study of learner error and vocabulary development.

This to certify that

i. the thesis comprises only my original work towards the PhD

ii. due acknowledgement has been made in the text to all other material used

iii. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendicies

Lars Yencken

Citations to Previously Published Work

Large portions of Chapter 4 have appeared in the following papers:

> YENCKEN, LARS and TIMOTHY BALDWIN. 2008. Measuring and predicting orthographic associations: modelling the similarity of Japanese kanji. In *Proceedings of the 22nd International Conference on Computational Linguistics*, 1041–1048, Manchester, UK.
>
> YENCKEN, LARS and TIMOTHY BALDWIN. 2006. Modelling the orthographic neighbourhood for Japanese Kanji. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*, 321-332, Sentosa, Singapore.

Similarly, large portions of Chapter 5 have appeared in:

> YENCKEN, LARS and TIMOTHY BALDWIN. 2008. Orthographic similarity search for dictionary lookup of Japanese words. In *Proceedings of the 18th European Conference on Artificial Intelligence*, 343–347, Patras, Greece.
>
> YENCKEN, LARS and TIMOTHY BALDWIN. 2005. Efficient grapheme-phoneme alignment for Japanese. In *Proceedings of the Australasian Language Technology Workshop 2005*, 143–151, Sydney, Australia.

Acknowledgments

There are many people who have supported me through this long adventure. I'd like to begin by thanking my parents for their persistent and dogged encouragement, my wife Lauren for her love and understanding, and my cat Nero for his quiet, enduring company along the way.

When I first began this work, Slaven Bilac visited our lab in Melbourne and handed over both source code and control of the FOKS system so that I could use it as a test bed for new dictionary research. I would like to thank him for his kindness and generosity, without which much of this work would not have been possible.

In late 2006 I spent 6 months with the Tanaka Lab at the University of Tokyo. I would like to thank Kumiko Tanaka-Ishii for kindly supporting my visit, and for expanding my perspective on many aspects of Japanese life, language and culture. My warmest thanks go also to my friend and colleague Zhihui Jin for our many long discussions during this period, and for his fresh insights as a Chinese learner of Japanese.

Finally, my most sincere and wholehearted thanks go to my long-suffering supervisor Tim Baldwin, for his sharp insight, stoic patience, constant warmth and all round good humour.

Thank you.

Dedicated to Lauren, my wife and partner in all things.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Words are the basic building blocks of language, and acquiring a large enough vocabulary to both understand others and be understood is the dominant task in second language learning. For learners of Japanese, learning words means learning the writing system used to encode them, in particular the *kanji* characters used to write them. Kanji are complex characters which embed meaning in a hierarchical, two-dimensional manner, in contrast to the linear, phonetic manner of most alphabetic scripts. This thesis centres around kanji, modelling how they may be confused by learners and in turn demonstrating how these confusion models can be used to improve dictionary lookup and automated test generation. Through this contribution, it aims to help learners in their self-study of Japanese vocabulary, so that they may better achieve their proficiency goals.

To make this more concrete, let us consider a common scenario of a hypothetical learner, Jane. Jane studies Japanese as a second language, but struggles constantly with the large number of words she has to learn. Since there are so many, most are only glossed over briefly in class and almost all her vocabulary study is done at home in her own time. Really, she would love to be reading in Japanese, but when she has tried even simple texts she finds she has to look up every second word in the dictionary.

One day she is trying to read a Japanese article, and comes across a new word, 養い. The second half い is phonetic, pronounced *i*, but she has no idea about the first half. Nothing about the character 養 indicates its pronunciation in this context, and without the pronunciation she cannot type the word into an online dictionary. Even if such a dictionary supported wildcards, *i* is a very common word ending, and she would never find her word. Jane realises that she must do things the hard way, and look it up in her paper dictionary. In order to look the word up, she sets about trying to find the first character in a traditional kanji dictionary. To do this, she has to identify the indexing radical, the component of the character traditionally used in dictionary indexes, usually the left-most or top-most component. Often components within characters are clearly delineated by white space, but in this character they are not. She guesses that the top half is probably the indexing radical, counts that it has 6 strokes, and tries to find it amongst the list of 6-stroke radicals in the dictionary. However, the nearest match she can find is 羊, which isn't quite what she was looking for because its main vertical stroke is too long at the bottom. She nonetheless continues just in case, going to the page number indicated for kanji containing 羊, counts that the rest of 養 would have 9 remaining strokes, and looks in the 9-stroke section for a match, without success.

At this point, with some frustration, she tries to work out what went wrong, rechecking stroke counts and wondering how else she could find this character in a dictionary. More frustrating still, Jane can recognise at least the bottom half of the kanji. It looks just like 食 from 食事 *shokuji* "meal". Then she recalls a recent online dictionary improvement which allows the user to search for words by visual similarity. She loads up its web site, and decides to put in the query 食い. The search results yield two words based on kanji which look similar to 食, and the second of these is the word she was looking for. She clicks "Translate", and finally reaches the information she needs. Her word is 養い *yashinai*, meaning "nutrition".

The broad problem Jane faces is really one of insufficient vocabulary, and it is common to all language leaners. It severely impedes early attempts at reading, since the time it takes to look up the many unknown words encountered makes comprehension difficult, and enjoyment unlikely. However, some of the additional difficulties faced by Jane are specific to learners of Japanese, especially from an English or Western language background, and relate to the problem of unknown kanji characters. Once they have acquired a small base of oral vocabulary, learners often study a new kanji character at the same time as they study a new word which uses it, so in Japanese study of kanji is very similar and ultimately bound to study of vocabulary. This thesis does not propose any revolutionary new method for studying kanji and words, but instead aims to help learners better utilise their existing study methods by modelling the implicit relationships between different kanji due to their visual similarity. This visual similarity is then available as a resource which can aid not only in dictionary lookup – as in Jane's case above – but also in evaluating a learner's proficiency.

This thesis targets Japanese for two main reasons. Firstly, Japanese is considered amongst the most difficult languages to learn for learners coming from an English or European language background, comparable only to Arabic, Chinese and Korean.[1] This difficulty makes it an attractive target for improvements to study methods, since even a small improvement may be significant over a long enough time span. Secondly, the aspects of Japanese kanji characters which present the largest obstacles for learners also present the largest opportunities for the use of rich user modelling.

In order to construct confusion models which approximate the human experience of kanji similarity and confusability, we draw inspiration from psycholinguistic research on kanji perception, and construct a number of plausible approximations to the human experience. These models consider the similarity of kanji as generic images, the extent to which components are shared, the extent to which broad layout is shared, the similarity in hierarchical tree structure, and the difference between the stroke sequences of two kanji. In order to choose between these models, we develop three separate datasets on which they can be evaluated. These data sets span a range of authentic sources, from the layman's raw judgements of kanji similarity to expert opinions on kanji confusability. By evaluating our models over these data sets, we find that the two models that best match human similarity judgements are tree edit distance and stroke edit distance.

As an initial application for these similarity models, we show that the FOKS (Forgiving Online Kanji Search) dictionary interface provides a useful test-bed for new search techniques. After re-engineering the dictionary into a more modular state, we convert our similarity models into confusion models, and use these kanji confusion models as the basis for search by similar kanji. It is this similarity search which allows queries like that of our earlier example, where 食い was used to find 養い. This significantly increases the accessibility of the dictionary by allowing learners to search for unknown kanji using their more

[1] From National Virtual Translation Center (2007) estimates of time required to learn a language fluently for English native speakers.

frequent and usually simpler visual neighbours. At the same time, it is able to automatically compensate for errors where users unintentionally mistake an unknown kanji for a familiar one they know, allowing learners to still find the word they were looking for. Through log analysis, we find that learners' use of this new form of search exceeds that of earlier lookup by pronunciation. Analysis of query and result pairs suggests that, rather than confusing rare items for known items as we had assumed, learners instead tend to confuse pairs of kanji where both lie on the periphery of their knowledge.

As part of our overhaul of the dictionary, we also rebuild its grapheme-phoneme alignment algorithm, the basis of its ability to model kanji reading errors. By moving to a semi-supervised algorithm, we increase its scalability with increasing dictionary sizes. This allows FOKS to upgrade its core dictionaries as new versions of them become available. We also extend the coverage of the dictionary to include greater coverage of place names, and improve its usability through better word translations and more transparent behaviour.

Most learners do not have the luxury of an immersive second-language environment in which to acquire words and test word knowledge through active use. This means that aside from dictionary-mediated attempts at reading, they limited to repetitive drills such as flashcards as their actual vocabulary study method. Whilst flashcards are a useful tool, they are not designed to measure proficiency over large numbers of words, such as an entire course or syllabus. For these larger proficiency estimates, learners typically use class exams, which are limited in scope to the class in question. For more rigorous testing and accreditation, the annual Japanese Language Proficiency Test (JLPT) is also available. However, the JLPT takes time, costs money, and can only be taken once a year. These obstacles are due to a fundamental limitation with manually constructed pen and paper tests: if the test is to measure what it was intended to, it cannot be taken by the same learner twice. This makes testing for proficiency expensive, since each new test requires a large investment of resources. Since it is expensive, tests with any strength are metered out sparingly to learners.

We show that using models of kanji confusion and kanji (mis)pronunciation, new tests can be generated automatically and uniquely for each learner. Since every test is different, there is no limit to the number of tests a user can take. We argue that by removing the scarcity from proficiency testing, learners are able to more effectively self-evaluate their knowledge, which in turn allows them to make better decisions about their study techniques.

Furthermore, by taking the learner's responses to earlier questions into account, these tests adapt to each learner individually, potentially creating more difficult tests which better highlight any flaws in a learner's knowledge. We do this through a new testing platform, Kanji Tester, which we develop to showcase the potential of automated test generation. Kanji Tester models itself on the JLPT, focusing on question types which directly test vocabulary or kanji knowledge. The JLPT is entirely multiple choice, so that it can be objectively marked; Kanji Tester uses its two main error models to generate plausibly incorrect multiple choice options, so as to increase the test's difficulty. Furthermore, each user's knowledge of kanji is explicitly modelled, and adapts to the user's responses after each test, so that any errors or areas of confusion the learner displays are more likely to be shown as options in later questions.

Analysis of several months of Kanji Tester's usage logs yields a number of interesting findings. The amount of per-user adaption was limited by data sparsity issues, suggesting that future systems should model errors across groups of users, for example across all users of the same first language background, or else ensure that learners revisit error cases more frequently. Despite this limitation, we find that adaptive test questions based on reading are significantly more difficult than those of our control baseline, due largely to the strength of the priors used in the adaptive models. Analysis of power user responses shows a weak positive correlation between ability and period of use, although most power users had high initial proficiency estimates, suggesting that scope for their improvement was limited.

Through this combination of enhanced dictionary support and improved, repeatable and accurate learner testing, this thesis supports learners of Japanese in their self-study of vocabulary, and ultimately in reaching their desired level of proficiency.

1.1 Aim and contributions

The primary aim of this research is to establish improved software-supported methods for self-study of vocabulary. In particular, we focus on Japanese as a language whose extreme orthographic depth means that learning to read well is difficult, and whose logographic characters offer a rich and under-utilised resource to draw from in aiding learners.

This thesis makes three main contributions. Firstly, it defines five separate models of graphemic similarity of Japanese kanji, establishes three separate data sets of human judgements, and evaluates the models over these judgements. It finds that stroke and tree edit distance metrics outperform other similarity models over these data sets. Secondly, it transforms these similarity models into models of plausible misrecognition of kanji, which in turn are used to provide a novel form of dictionary search by similarity. Through post-hoc log analysis, it demonstrates that this form of search is both viable and highly useful to learners. Thirdly, it combines kanji confusion models with existing models of plausible misreading in order to generate vocabulary tests for quick self-evaluation of proficiency. It finds that this modelling allows creation of questions which are significantly more difficult than those of a control baseline.

A general theme of this thesis is the use of careful error modelling to support learners in their study. These error models are used firstly to compensate for user errors in a dictionary search application, then to explicitly provoke user errors in a testing application. In both cases, we argue that learners are better served by the heightened awareness of their behaviour that these applications display.

1.2 Audience

This thesis is aimed at a general academic audience with an interest in second language learning, in particular for East-Asian languages. Prior knowledge of the Japanese and Chinese writing systems is helpful but not required. However, several chapters perform formal modelling or evaluation which require some basic knowledge of probability theory, metric spaces and uses concepts from information retrieval. We nonetheless expect a non-technical reader to be able to gloss over these parts and still gain a good understanding of this work.

1.3 Thesis structure

The remainder of this thesis is broken into six chapters as follows:

Chapter 2: The Japanese Writing System

We provide an overview of the Japanese Writing System, discussing its origins, the three main scripts used, and in particular the kanji script. Issues relating to computer input and dictionary lookup of Japanese are covered, before concluding with a brief overview of four separate word formation effects which complicate word pronunciation.

Chapter 3: Learning Japanese vocabulary

This chapter firstly makes the case that vocabulary study is the largest barrier for language learners, and argues that the majority of vocabulary is acquired through autonomous self-study. It examines in detail what it means to know a word, and thus demonstrates that the word is an appropriate lens through which to view second language acquisition.

We investigate and discuss strategies for deciding what words to study, and then how to study them. We then consider how words are accessed and stored in the mental lexicon, and relate these patterns to the general lexical relationships which form between words. We introduce the concept of graphemic neighbourhoods as the relation of near-homography, and situate it amongst other better known relationships such as near-synonymy and homophony. Finally, we explore the state-of-the-art in dictionary lookup for Japanese, in particular focusing on methods for input and lookup of kanji characters. We discuss the role of testing in second language learning, and consider the gradual transition from paper-based testing to computer-adaptive testing, before examining the current forms of tests actually used in the field.

Chapter 4: Orthographic similarity

We investigate the notion of graphemic similarity and propose several formal models: a cosine similarity over radical vectors, designed to capture the salience of radicals as significant subcomponents; the L_1 norm over kanji images, attempting to measure visual similarity in a raw and unbiased way; the tree edit distance between kanji structure representations, attempting to compare layout information; and, an edit distance over stroke vectors, which tries to both capture the salience of stroke order, and to perform a fuzzy matching of larger components.

Three separate data sets are used to evaluate these metrics: a simple experiment eliciting similarity judgements from learners, a series of expert judgements mined from a commercial flashcard set and an experiment eliciting native-speaker judgements of confusability. Evaluating against these data sets, we find that tree edit distance and stroke edit distance models achieve best performance, and do comparably well to one another. Since stroke edit distance is two orders of magnitude faster than tree edit distance to calculate, we establish it as the preferred metric given available data.

Chapter 5: Extending the dictionary

We examine the current contribution of the FOKS dictionary interface to the Japanese lookup space, and assess its suitability as a test bed for new forms of dictionary search, ultimately rebuilding it to serve this function. Its unsupervised grapheme-phoneme alignment algorithm is examined and replaced with a more scalable semi-supervised method using a dictionary of kanji readings. The resultant improvement in scalability allows our reimplementation of FOKS to utilise the newer and larger dictionaries versions becoming available over time.

The usability of the system is then enhanced in several ways: its coverage of proper nouns is extended through the construction of a simple place name gazetteer, dictionary translations are improved to display richer word information, and error correction is made transparent to users through a query explanation tool.

Finally, and most significantly, we use the stroke edit distance developed earlier to power a confusability model for Japanese kanji, based on the assumption that learners will confuse rare kanji for their more frequent neighbours. We use this model to implement search-by-similarity in FOKS. A theoretical argument is made concerning the potential improvement in accessibility that this new form of search can provide for learners. Post-hoc analysis then finds that over the log period analysed, search-by-similarity was used more frequently by learners than FOKS's existing intelligent reading search, demonstrating the utility of this method.

Chapter 6: Testing and drilling

This chapter investigates the use of error models to generate randomised vocabulary tests through a new system called Kanji Tester, which serves both as an improved means for learners to rapidly self-evaluate and as an alternative and more constrained platform for evaluating our error models. It proposes that proficiency tests can be sorted by their *scope* and their *availability*, and situates the ideal proficiency test, the JLPT tests, and our current work along these scales. It then discusses in detail the form of the JLPT test, giving examples of the question types that Kanji Tester attempts to emulate.

We then discuss Kanji Tester itself in detail, both in its interface from a user's perspective, and in its extensible architecture and underlying modelling from our perspective. Amongst the various forms of potential user models Kanji Tester could use, we discuss and justify our use of per-user per-kanji error models. We then describe our algorithm for generating questions, and for updating each user's error models.

In our evaluation, we discuss differences between human constructed tests and the automated tests generated by Kanji Tester. A weak positive correlation is found between test scores and time over which Kanji Tester is used. Through an analysis of rater responses we find that, despite issues with data sparsity, our adaptive test questions cause significantly more errors than those of a control baseline.

Chapter 7: Conclusion

Finally, we summarise the thesis and identify three main areas for future work. Firstly, we suggest potential improvements to the graphemic similarity models this thesis has proposed, and useful experiments which could be performed to better understand the topologies they generate. More generally, we identify in this discussion further areas of interest in the modelling of lexical relationships. Secondly, we suggest several areas in which dictionary accessibility could be improved, including better example sentence selection, crowdsourcing of dictionary enhancements, and the potential use of semagrams in future open dictionaries. Thirdly, we discuss the remaining gap between our automatically generated tests and human-constructed tests, and ways of narrowing that gap. These include improved user modelling for the same types of questions, and ways to generate new and different forms of questions which could help to test proficiency.

Chapter 2

The Japanese writing system

This chapter gives a brief and focused introduction to the Japanese writing system, in order to provide the background necessary to discuss issues specific to acquisition of Japanese vocabulary. Readers already familiar with Japanese may skip ahead to Chapter 3. Likewise, readers interested in a more comprehensive coverage of the Japanese writing system are referred to works by Backhouse (1996) and Tsujimura (1999).

2.1 Overview

The Japanese writing system uses three principal scripts, the morpho-syllabic *kanji* script and the syllabic *hiragana* and *katakana* scripts. The latter two scripts contain the same sound inventory, and so are potentially interchangeable, but in practice each finds complementary and largely non-overlapping use. They encode not full syllables, but a type of short syllable called a *mora*. These scripts are best understood in terms of the division of labour between them, as suggested by Backhouse (1996:43). Hiragana is principally used for grammatical elements such as inflectional suffixes for verbs and adjectives, case markers, grammatical nouns, amongst others. Whilst katakana encodes the same sounds, its use is likened to that of italics in English to mark a word for special attention. It finds principal use in writing loanwords or names from English or European languages. These two syllabic scripts are usually learned in full in a short period of time, and are not a significant challenge for learners. We now discuss the more complicated kanji script.

Kanji are used at the morpheme level in Japanese, and are used to write most lexicalised words, such as 家 *uchi* "house, family". Verbs and conjugating adjectives typically use kanji to represent the fixed stem of the word, and hiragana for the changing inflectional suffix (Backhouse 1996:51), as in 行く *iku* "to go". Kanji in context have readings of one or more mora, but this reading need not cover the whole stem, as shown by examples such as 厳しい *kibishī* "strict". Furthermore, the stem itself may not end on a mora boundary: our earlier example 行く *iku* has stem *ik-*, with inflected forms such as 行き *iki* and 行け *ike*.

As a generalisation, kanji which form standalone words have more concrete meanings, whereas kanji which form bound morphemes only take on concrete meanings with the addition of either other kanji to form a compound or hiragana inflection. The senses of an individual kanji can be considered its semantic contribution to the many compounds and words in which it takes part, where such a contribution is transparent. These senses in turn are often extensions of a single core meaning. For example, Halpern (1999) lists glosses for several senses of kanji 暴 *bō*:

1. violent, rough, wild, cruel, tyrannical
   as in 暴力 *bōryoku* "violence, force"

2. sudden, abrupt
   as in 暴落 *bōraku* "slump, crash (in prices)"

3. unrestrained, inordinate, wild, excessive, irrational
   as in 暴飲 *bōiN* "heavy drinking"

The core meaning given is "violent".

Kanji characters were historically borrowed from Chinese – whose characters we call *hanzi* – and most Japanese characters are also valid characters in Traditional Chinese. However, both Chinese and Japanese have undergone some simplification, and these simplified forms are not necessarily shared. The character etymology for 歳 *sai* "year, age" in Figure 2.2 is a common example of this process. We call two characters which share the same historical roots *cognate*. Despite more extensive character simplification in Chinese, this cognate relationship between character pairs means that they typically remain similar in form, and indeed are often identical. However, the same can not be said for their pronunciation.

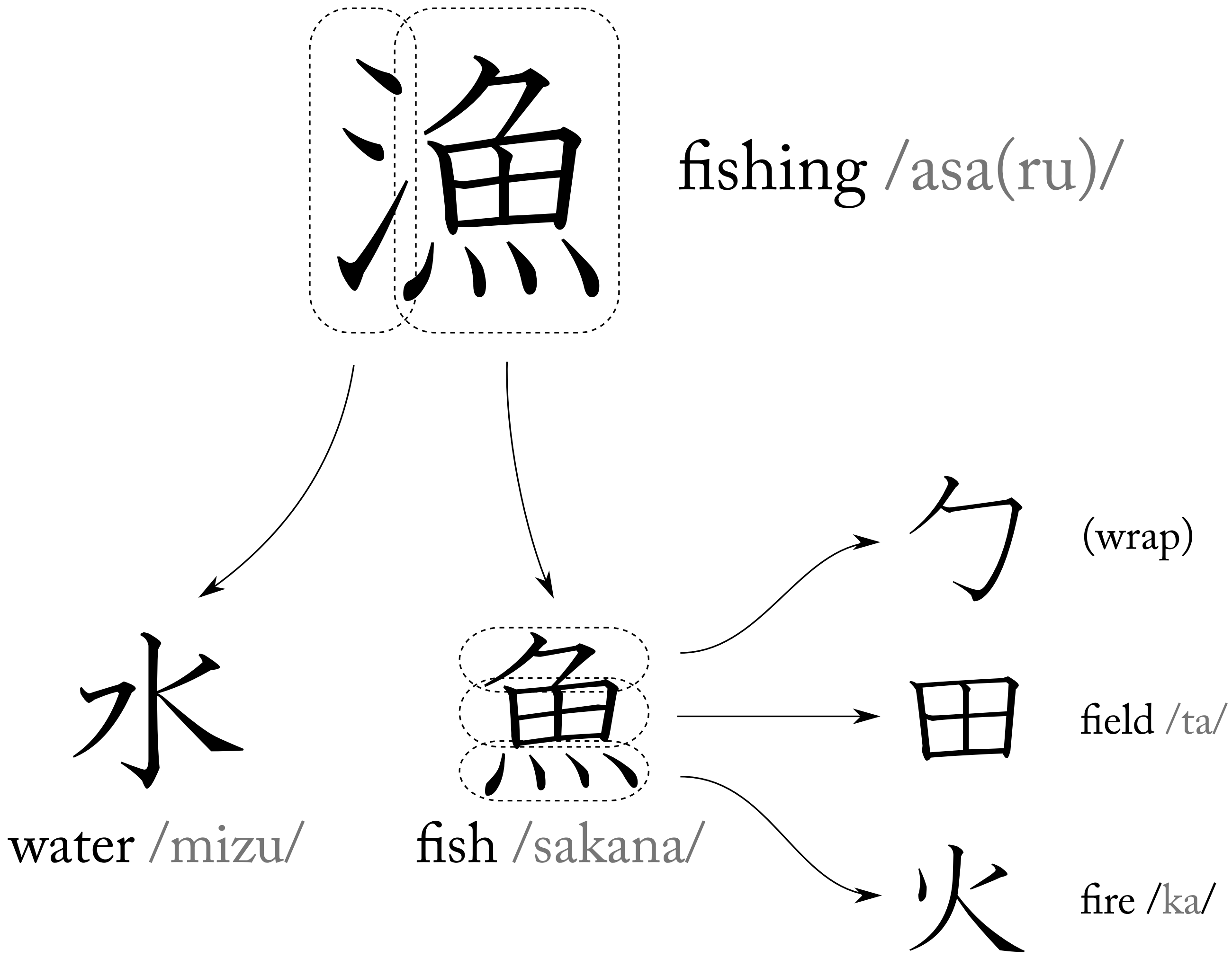


Figure 2.1: Exposing the compositional nature of kanji. 漁 *asa(ru)* “fishing” is made by combining 水 *mizu* “water” (via short form 氵) and 魚 *sakana* “fish”. Further decomposition of 魚 is possible, however its subcomponents do not have a semantic relationship with its whole; this is because 魚 is a pictograph (i.e. it is derived from a picture of a fish rather than from these components).

Each Japanese character, kanji or otherwise, occupies an identically sized square space when printed or hand-written. For example, a simple kanji such as 工 is given the same space as a more complex kanji like 蠹. This can be contrasted with other writing systems, for example the latin alphabet, where each letter may occupy a different amount of space. Kanji are also compositional, so that many basic kanji also occur as components of more complex kanji. An example of this composition is given in Figure 2.1, where we decompose 漁 *asa(ru)* "fishing" into its constituent parts 水 *mizu* "water" and 魚 *sakana* "fish". Note that to preserve space, 水 is given in its short form 氵; a number of kanji have a short form radicals for composition in this manner. In this case, both "water" and "fish" are semantically related to "fishing", so 漁 is a good example of reasonably transparent semantic composition. Note that 魚 itself is composed of radicals 勹, 田, and 火 (via short form 灬), with meanings "wrap", "field" and "fire" respectively, but these meanings seem to bear no relation to the overall meaning "fish". This semantically opaque example occurs since 魚 is a pictograph, historically derived from a picture of a fish rather than from these component radicals.

Each kanji may have multiple pronunciations which it takes in different contexts, and to explain this it helps to briefly discuss how these readings occurred. Kanji were originally borrowed from Chinese during several different periods in history when different Chinese dialects were prominent, so each individual kanji may have different pronunciations borrowed from any or all of these periods. These borrowed readings are known as *on* readings (from 音 *oN* "sound"), and are the source of the limited similarity in pronunciation between cognate characters and words in Japanese and modern day Chinese. An *on* reading may be indicated visually by a phonetic component within a kanji, such as 同 *dō*, as in 銅 *dō* "copper", but may also occur without visual cues. In contrast, *kun* readings (from 訓 *kuN* "explanation") are native Japanese pronunciations which are never visually indicated, and must simply be learned.[1]

As a broad generalisation, *on* readings are used for kanji when combined with other kanji into compounds, whereas *kun* readings are used in combination with hiragana inflectional suffixes. Native speakers learn both for common characters, and can distinguish between

[1] Note that for transliteration of Japanese linguistic examples we use the Hepburn system with Backhouse style "N" *mora*. This allows us to distinguish between the transliterations of 勲位 *kuNi* "order of merit" (segmented as *ku-N-i*) and the 国 *kuni* "country" (segmented as *ku-ni*). However, when borrowing linguistic terminology like *kun* from Japanese, we use the standard Hepburn system for simplicity.

reading types. The frequency distributions of each reading type also differ, naturally reflecting the languages and dialects they originated from. In extreme examples, *koN* only occurs in Japanese as an *on* reading, but 154 kanji may take this pronunciation, including 金, 今 and 近; *aki* is only used as a *kun* reading, but 108 kanji may take it, including 日, 明 and 成.

If we compare with Chinese, we find that Chinese only uses the hanzi script, and its major dialects have different sound inventories to Japanese. Furthermore, each hanzi typically has only one valid pronunciation in each dialect, which may in turn be cued by phonetic radicals where they occur. This substantially simpler form-to-sound mapping means that work based on Japanese pronunciation is not likely to be directly useful for Chinese. However, where this thesis concerns itself with the visual form of single kanji and of multi-kanji compounds, we expect such work to generalise gracefully.

Our examples of *koN* and *aki* also indicate the extreme level of homophony which is found in Japanese at the character level, though this is reduced at the word level. In spoken language, context and reading frequency provide sufficient tools to disambiguate a kanji or word. In written language, the use of kanji makes this effect negligible: a kanji's pronunciation is reflected in its visual form in a limited manner, so homophony and homography need not overlap. A clever tongue-twister takes advantage of this. にわにはにわにわとりがいる is pronounced *niwa niwa niwa niwatori ga iru*, and is extremely ambiguous if either spoken aloud or written in hiragana due to its use of several homophones. However, when kanji are used it is written as 庭には二羽鶏がいる and contains no semantic ambiguity, meaning "there are two chickens in the garden". The comparable *Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo* sentence in English, comprised exclusively of three different senses of the word *buffalo*, retains its ambiguity in written form (Pinker 1994:210). This concisely illustrates the utility of kanji in disambiguating homophones during reading.

2.2 Input and lookup of Japanese

Computer input in Chinese and Japanese is markedly different from that of languages such as English with alphabetic scripts, since there are far too many different characters to admit any simple mapping from the available keys on a keyboard to characters a user may wish to write. For this reason, a sequence of keystrokes is required to build an input

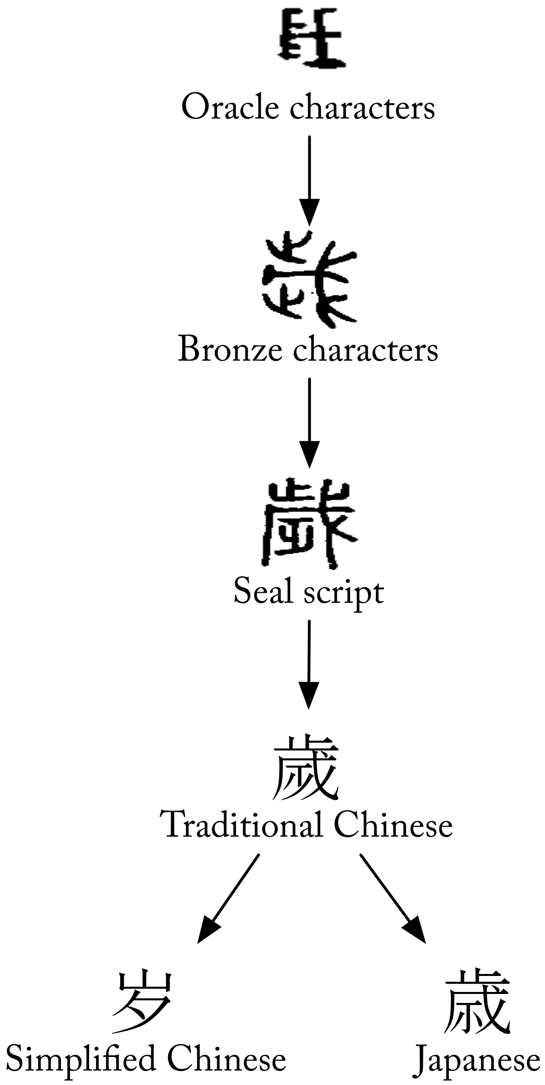
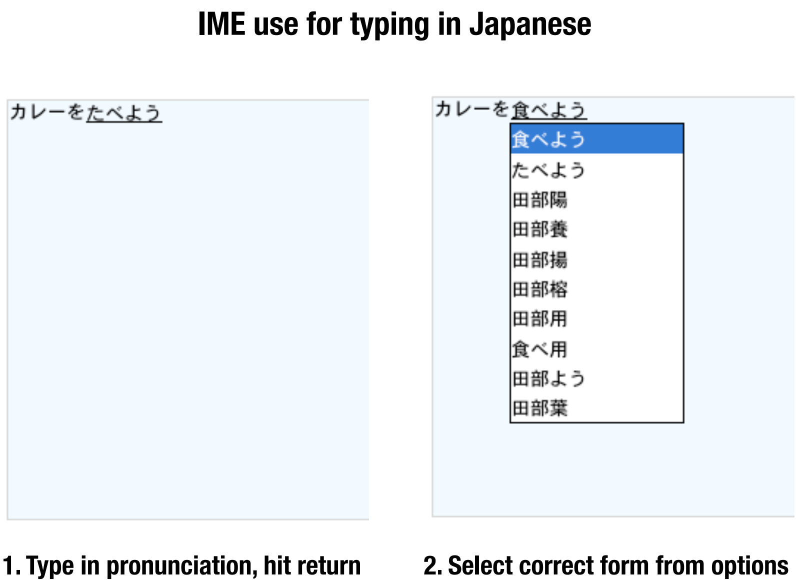


Figure 2.2: The etymology for the character 歳 *sai* "year, age" from oracle characters to modern day usage in Chinese and Japanese, and the cognate relationship between the two usages. Cognates are from Goh *et al.* (2005), and early character forms are from Sears (2008).

space large enough to map to any character. This led to the use of input method editors (IMEs), programs which translate a sequence of keystrokes into a shorter sequence of more complicated characters, usually interactively. Languages with alphabetic scripts also use an IME when the keyboard size is restricted, for example the "predictive text" feature on most modern mobile phones.



IME use for typing in Japanese

カレーをたべよう

1. Type in pronunciation, hit return

カレーを食べよう

食べよう
たべよう
田部陽
田部養
田部揚
田部榕
田部用
食べ用
田部よう
田部葉

2. Select correct form from options

Figure 2.3: An example of IME use to type in Japanese. The pronunciation is first entered, then matching word forms are displayed to choose from. The example reads カレーを食べよう *karē o tabeyō* "let's eat curry"

In Chinese, where only one script is used,[2] there is a considerable variety of methods used by IMEs to constrain the space of characters down to the character the user was looking for. Ultimately, this is achieved based on cues about the character's *form*, as in the Wubizixing (五笔字型 [*wŭbĭ zìxíng*]) or Cangjie (仓颉 [*cāngjié*]) methods, or based on the character's *pronunciation*, as in the Pinyin (拼音 [*pīnyīn*]) or Zhuyin (注音 [*zhùyīn*]) methods. For both Chinese and Japanese languages, the large amount of homophony means that input by pronunciation is best disambiguated in word-level chunks, since the additional context helps to constrain possible matches.

[2] Actually, modern Chinese characters occur in both *simplified* and *traditional* forms, as shown in Figure 2.2. Since each dialect predominantly uses one script or the other, we can call Chinese languages single-script languages.

Both forms of input methods suffer the same basic problem: when based on form, the user is unable to type a character whose form they have forgotten or never learned; when based on pronunciation, the user is unable to type a character whose pronunciation they have forgotten or never learned. Since Japanese contains the two syllabic kana scripts, its input methods are based on pronunciation, and Japanese input thus faces the latter problem. However, the problem is worse for Japanese, since the user cannot easily change input method when faced with a problematic character, as graphemic input methods are not readily available for Japanese.[3]

Compounded with this, kanji have much weaker cues to pronunciation than hanzi. Recall that some kanji have reliably phonetic radicals which provide a known *on* reading, as in our earlier example 同 *dō*, as in 銅 *dō* "copper" and 胴 *dō* "body". However, even in these cases, only one of many potential readings is known. For example, 同 can also be read *ona* (as in 同じ *onaji* "same") and 銅 can also be read as *okagane*. In most cases there is no reliable phonetic to suggest the pronunciation of an unknown kanji, and thus no easy means of computer input.

An appropriate response to this fundamental problem for Japanese IMEs is a change in input modality. This is attempted by the many handwriting recognition interfaces for Japanese: dictionaries for the Nintendo DS,[4] Todd Ruddick's web-based recognition applet,[5] Tomoe[6] and many commercial electronic dictionaries. However, of the computer-based systems, many 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. On the other hand, such technology has improved markedly in the past few years. For example, Ben Bullock's new hand-written interface[7] is significantly more robust than earlier interfaces in terms of these issues. Overall, whilst such systems are not always optimal, they provide a very useful

[3] Phonemic input methods are reasonably natural, since input is a simple representation of speech. However, graphemic input methods require some more complicated mapping scheme, which takes time and training to use. Furthermore, the two additional syllabic scripts in Japanese are far better suited to phonemic input methods.

[4] For example, 漢字そのまま楽引辞典 *kaNji sono mama rakubiki jiteN*.

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

[6] `http://tomoe.sourceforge.jp/`

[7] `http://kanji.sljfaq.org/draw.html`

alternative in situations where other input methods are too slow.

The Kanjiru dictionary takes a different approach, adapting Wills and MacKay's (2006) Dasher accessibility interface, attempting to interactively assemble a character by shape and stroke in an adaptive manner (Winstead 2006; Winstead and Neely 2007). 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. Though cumbersome for typical input, it provides an alternative version of input-by-form for difficult characters.

Despite this arsenal of lookup and input techniques, inputting or looking up an unknown kanji or word remains a barrier for learners, as indicated by the continuing emergence of new lookup methods. We revisit the topic of dictionary lookup in more detail in Section 3.3.

2.3 Word formation in Japanese

Japanese is an agglutinative language, which can present additional difficulties in appropriately determining word-hood. New words in Japanese are principally created through one of several mechanisms: borrowing from foreign languages, compounding of existing words, modification of existing words, or clipping of existing words to create more colloquial forms (Backhouse 1996:81). Since most loanwords are written in katakana, which is phonetically transparent to learners, we largely ignore word formation from borrowing in this thesis. In later chapters we will show that many of the errors which learners make in dictionary lookup and testing relate to a lack of transparency in the compounding process by which many words are created. We thus also ignore word formation by modification or clipping in order to focus better on aspects of the compounding process.

In later parts of this thesis it will become important to be able to reverse the compounding process, decomposing a compound back into its smallest constituent morphemes. In Japanese, this task is essentially one of Grapheme-Phoneme alignment, which we consider in earnest in Section 5.2. Here, we discuss briefly four pertinent features of word formation by compounding and its reversal: *okurigana*, *sequential voicing*, *sound euphony* and *grapheme gapping*.

Okurigana

Single kanji do not always correspond to whole morphemes in written language, often requiring a hiragana suffix to form a unit. This is especially true for kanji forming verbs or *i*-adjectives, where the hiragana suffix forms an inflectional ending (Backhouse 1996:43). For example, 行 is not a word by itself, but with suffix く *ku* forms the verb 行く *iku* "to go". In general, these suffixes are called *okurigana*, and useful alignments should include them along with their kanji stem in order to preserve the basic morpho-phonemic structure of the compound.

Although most cases of okurigana represent verb and adjective conjugation, there are many general cases such as that of the kanji 取, which occurs in compounds almost exclusively with suffix り *ri* as 取り *tori*, but is sometimes written with the suffix *ri* conflated with the kanji stem. For some words, the written form can occur either way. For example, 受け取り *uketori* "receipt" can equally be written 受取り, 受け取 or 受取, depending on whether the two hiragana け *ke* or り *ri* are conflated into their respective kanji stems. Ideally, alignment systems should capture such alternations in order to achieve consistent segmentation behaviour. This is also useful for attaining an accurate estimate of the frequency with which a given kanji occurs with a particular reading which is independent of the exact lexical form of the word.

Sequential voicing

Sequential voicing occurs when the second component in a compound has its initial consonant changed from an unvoiced sound to a voiced sound (Backhouse 1996:82). For example: 本 *hoN* "book" + 棚 *tana* "shelf" → 本棚 *hoNdana* "bookshelf". Its occurrence is limited by a number of constraints. For example, it is mainly restricted to native words, and occurrence is further constrained by Lyman's law, which states that sequential voicing will not occur where there are existing voiced obstruents in the tailing segment (Vance 1987). It occurs in about 75% of cases where Lyman's law is not violated, with some systematic irregularities for noun-noun compounds (Rosen 2003).

Alignment methods based on precedence or frequency counts may be hindered by sequential voicing, since aligned grapheme/phoneme pairs may not be recognised as phonological variants of previously seen kanji–reading pairs. Fortunately, devoicing is a relatively simple 1-to-1 process, so a common approach is to simply consider voiced and devoiced grapheme/phoneme pairs to be equivalent for counting or comparison.

Sound euphony

Sound euphony occurs when the first component's last syllable alters to match the sound of the tailing segment. This creates a syllable-final geminate consonant which is pronounced for audibly longer than a normal short consonant (Backhouse 1996:26). In Japanese, gemination is indicated by the small っ character, and is typically romanised into double consonants, such as in まって *matte* "wait" and いっぱい *ippai* "full". An example of sound euphonic change is in the combination of 国 *koku* "country" and 境 *kyō* "boundary" to create 国境 *kokkyō* "national border".

Unlike sequential voicing, which imposes an easily reversible transformation, it is not clear from 国境 *kokkyō* "national border" what the original pronunciation for 国 was (possibilities include *koki*, *koku*, *kosu* and *kotsu*). This can introduce some difficulty in reversing the compounding process and determining correctly the pronunciation of the original parts.

Grapheme gapping

We saw earlier that in cases of okurigana, part of the reading for some words can be optionally conflated into the kanji. Grapheme gapping refers to a much rarer occurrence where a particle which is a standalone morpheme is subsumed into a kanji compound. Typically in such cases it is also acceptable to write the particle explicitly. For example: 山 *yama* "mountain" + の *no* "GENITIVE" + 手 *te* "hand" can be written as either 山手 or 山の手, both with identical pronunciation *yamanote*. In the first case the particle *no* is subsumed as part of the compound; in the second it is explicitly written. Grapheme gapping is very rare, normally only occurs with the particles *ga* or *no*, and tends not to be productive, suggesting that even high-precision alignment systems need only store individual known cases.

—

With this focused overview of the Japanese writing system complete, and in particular having discussed the basic problems for learners associated with the kanji script, we can turn to the main theme of this thesis, vocabulary acquisition.

Chapter 3

Learning Japanese vocabulary

In this thesis, we focus on improving study tools for learners of Japanese so that they may better acquire vocabulary and ultimately, improve their Japanese proficiency. This chapter provides the motivation for our approach, and then extended background into each of the focuses for later chapters.

In Section 3.1 we firstly situate this thesis within the debate on how to best facilitate second language acquisition. We argue that supporting learners in their self-study of vocabulary is likely to yield the greatest benefit. We then examine current self-study strategies and tools, focusing on Japanese as a target language, and argue that better modelling of user errors could greatly improve their effectiveness, especially for learners at the crucial early reading phase.

In order to guide this error modelling, Section 3.2 examines current research as to the structure of the mental lexicon to try to determine how words are accessed during reading. We show that the access structure for Japanese kanji is suggestive for how we might model misrecognition errors and error models based on visual similarity.

Since dictionary lookup and language testing are arguably the two most fundamental mechanisms for second language learning, we go on to examine each in turn. Section 3.3 firstly discusses the current state of dictionary lookup for Japanese from the perspective of both readers and writers, and then Section 3.4 considers recent advances in language testing. These sections provide the necessary background for Chapters 5 and 6 respectively.

In discussing vocabulary, this chapter uses the basic unit of vocabulary, the word, as a lens through which to consider language acquisition. However, different fields of research use different definitions of "word". In this thesis, we use "word" to mean *lemma*, a headword in canonical form from which inflected forms can transparently be derived. For example, we consider *change*, *changes*, *changed* and *changing* to be forms of the same word *change*, rather than separate words in their own right. Where we refer to research which uses an alternative definition, such as the narrower meaning of *token* or the broader meaning of *word family*, we use the more specific terminology instead.

Whilst we recognise the strong role of memory in word knowledge, a proper treatment of this broad issue is beyond our scope. This chapter examines in detail how words are accessed and many aspects of how they are best stored, but largely ignores longer-term issues of ensuring that words learned stay learned. Readers looking for a more comprehensive discussion of the role of memory in language are referred instead to works by Baddeley (1997, 2003).

A final caveat to our discussion is that, despite referring to relevant Chinese- or Japanese-language work that we are aware of, our coverage of academic publications in these languages is regrettably more limited than that of English-language publications. This has little effect on our main focus of second language learning, since such work is multilingual by nature, however it does restrict our view of native speakers of these languages. Detailed aspects of native speaker development particular to Chinese and Japanese are thus beyond the scope of this thesis.

3.1 Autonomous vocabulary acquisition

The beginner's paradox

Vocabulary is significant, above all other areas of language proficiency, because it is open-ended. New product names, person names, and word forms are created every day, and a native speaker will continue to acquire words throughout their entire life. This open-ended nature makes learning sufficient vocabulary a heavy burden on second language learners. Frequency and redundancy allow communication with a limited lexicon, but for fluent communication learners must still develop a lexicon of sufficient size.

So how do learners acquire such a lexicon? For first-language (L1) learners, our answer has two parts. By the age of five, L1 English children know some 3000-5000 "word families", covering most of spoken language, without direct instruction (Nation 2001:96). The source of this initial oral vocabulary is debated, but they are presumed to have learned it through exposure to adult speech. Yet between the ages of 8 and 18, they similarly continue to learn between 7 and 15 words per day, depending on how we count (Landauer and Dumais 1997; Vermeer 2001). Furthermore, since about three quarters of an English speaking L1 adult vocabulary occurs almost exclusively in written text, it must have been acquired through reading, as suggested by Miller (1941). The vocabulary acquired this way seems to be the "long tail" of language, made up of large numbers of medium-to-low frequency words. Current evidence thus suggests that in an L1, vocabulary is acquired *autonomously*, and *through reading widely*.

In a second language (L2), Krashen (1989) similarly argued that reading widely is the source of most vocabulary acquired. However, in an L2 this is premised on having already acquired the early core vocabulary. This is reflected in high-level reading skills which only begin transferring from a learner's L1 to their L2 at around the 3000 word family mark (Laufer 1997:24), and also in Hsueh-Chao and Nation's (2000) study of unknown word density which suggested 98% of word tokens in a text should be known for reading to be comfortable. Although graded readers[1] can be used to achieve this at lower vocabulary levels, learners must still acquire a significant amount of vocabulary before they can expect success in a reading program. Indeed, many factors may play a part in the effectiveness of reading widely for L2 vocabulary learning, and several studies such as Coady (1997:226) and Fraser (1999) describe mixed results. Landauer and Dumais (1997) provide a potential explanation for some of these results, suggesting that word knowledge is built gradually from large numbers of weak links with other words. If this was the case, then much of the vocabulary improvement from reading a series of text would appear in words not explicitly encountered in the texts, but rather whose meaning has solidified through relationships with encountered words. This suggests a potential source of measurement error for studies

[1] A graded reader is a text where words assumed to be too difficult for the reader are either pre-glossed or replaced by simpler near-synonyms. In this way a text is constructed with an appropriate density of unknown words for the reader.

of reading that look for vocabulary growth only in explicitly encountered words.

Nonetheless, some controversy remains amongst experts as to the full role of reading in L2 vocabulary growth. In this thesis, we make the reasonable assumption that L2 learners achieve the same vocabulary growth from reading as L1 learners given similar vocabulary size, reading materials, and opportunities for input. Where differences exist, we assume they lie in background, environmental or strategic factors rather than the L1/L2 distinction.

Vocabulary knowledge promotes reading comprehension: it is the greatest predictor of reading success, better than general reading ability or syntactic ability (Laufer 1997; Urquhart and Weir 1998). The converse is also clear, that reading comprehension promotes vocabulary growth, since strong context allows learners to better infer the meaning of unknown words. These two mutually enforcing factors seem to form a positive feedback loop, which could form a strong platform for reaching fluency.

Nuttall (1996:127) however found that there are not one but two feedback loops, depending on a learner's attitudes, motivation and existing reading proficiency: a "virtuous circle" reinforcing success and a "vicious circle" reinforcing failure (Figure 3.1). These circles are so-named because starting at any of the points leads to each of the other points in the cycle. Pulido and Hambrick (2008) in turn validate these ideas through empirical study of L2 reading. Coady (1997) calls this problem the "Beginner's Paradox", since learners must acquire vocabulary through reading, but need sufficient vocabulary to bootstrap the process.

If we are serious about helping learners to reach higher levels of fluency, we must focus our efforts where the long-term impact will be the greatest. This is not a new idea: where the appropriate strategies are not developed automatically, many researchers now advocate teaching autonomous learning strategies explicitly, at the cost of time spent explicitly teaching language, because of the beneficial effects of these strategies on long-term autonomous behaviour (Huckin and Coady 1999; Barker 2007). Use of appropriate tools is also increasing; Tozcu and Coady (2004) found that teaching high-frequency words using a CALL system improved reading more than simple reading practice. This thesis describes attempts to improve tool support for learning vocabulary at the early reading stage, in order to ensure leaners achieve a positive cycle of vocabulary growth in their reading.

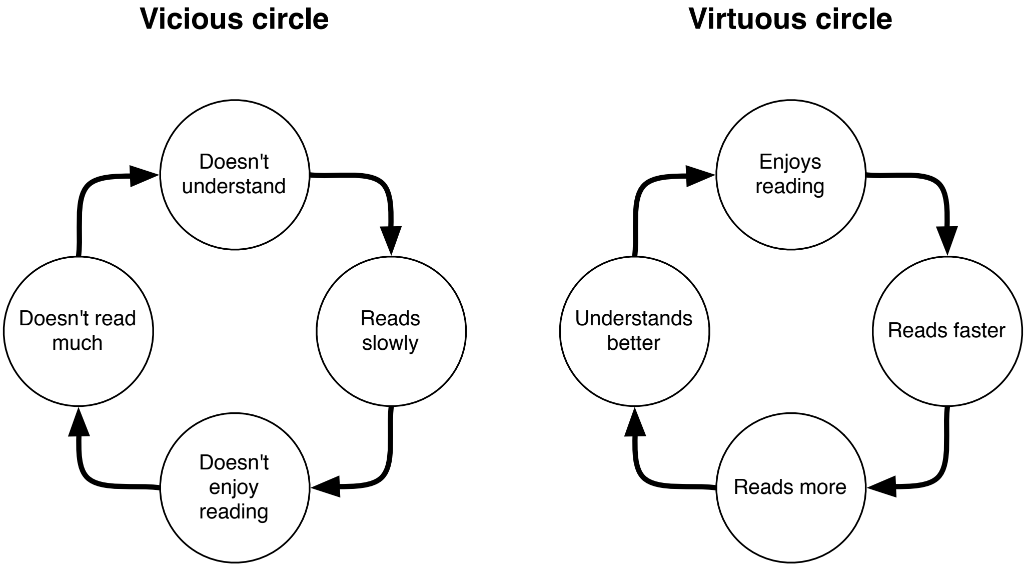


Figure 3.1: Nuttall's (1996) cycles of frustration and growth in L2 reading.

Knowing a word

In order for a learner's vocabulary size to be accurately measured, we need a sense of what it means to acquire a word. We settled earlier on using "word" to mean lemma. To determine what we might mean by acquire, we ask, what does it mean to know a word?

Nation (2001:27) attempts to provide an exhaustive listing of dimensions of word knowledge. Each aspect fits under the broad categories of *form*, *meaning* and *use*, and furthermore distinguishes between the knowledge required to recognise a word (receptive knowledge [R]) and that required to produce the word correctly (productive knowledge [P]). Using this distinction, Table 3.1 attempts to exhaustively describe the dimensions of word knowledge. This enumeration is important since it provides an implicit definition of *depth* of vocabulary knowledge: the depth of knowledge of a word is the learner's coverage over these aspects of knowledge. It also serves as a reminder that word knowledge does not exist independently in a vacuum, but in fact cross-cuts almost all aspects of linguistic proficiency.

Most studies into vocabulary acquisition have used languages with alphabetic orthographies (in particular English, French and German) as their L2. So what differences could we expect with languages like Chinese and Japanese which use more complex scripts? At the

Form	spoken	R	What does the word sound like?
		P	How is the word pronounced?
	written	R	What does the word look like?
		P	How is the word written and spelled?
	word parts	R	What parts are recognisable in this word?
		P	What word parts are needed to express the meaning?
Meaning	form and meaning	R	What meaning does this word form signal?
		P	What word form can be used to express this meaning?
	concept and referents	R	What is included in the concept?
		P	What items can the concept refer to?
	associations	R	What other words does this make us think of?
		P	What other words could we use instead of this one?
Use	grammatical functions	R	In what patterns does the word occur?
		P	In what patterns must we use this word?
	collocations	R	What words or types of words occur with this one?
		P	What words or types of words must we use with this one?
	constraints on use (register, frequency)	R	Where, when and how often would we expect to meet this word?
		P	Where, when and how often can we use this word?

Table 3.1: Aspects of word knowledge, marked “R” for receptive knowledge and “P” for productive knowledge. From Nation (2001:27).

Form	spoken	P	How is the kanji pronounced in this context?
	written	R	What semantic components are used in this kanji? How do they contribute to the whole-kanji meaning?
		R	What phonetic components are used in this kanji? Do they provide any *on* readings?
		R	What other variants exist for this kanji?
		P	What is the correct stroke order for this kanji?
Use	constraints on use	P	In what context is it appropriate to use this kanji? When should its kana equivalent be written instead?

Table 3.2: Our proposed aspects of kanji knowledge, in complement to aspects of word knowledge shown in Table 3.1.

kanji level, many word-level aspects also apply, but some elements such as pronunciations in context, variants and stroke order are also important. We propose the additional aspects of knowledge in Table 3.2 as applying to kanji.

Returning to our question of what it means to acquire a word, we are left with two extremes. A word we have never encountered is clearly not acquired, yet must a learner know every aspect of a word (or kanji) to have acquired it? Acquisition is clearly a graded concept. In particular, the extent to which a learner knows about a word is considered the *depth* of their knowledge, whereas the number of words of which they have some knowledge is the *breadth*.

Most vocabulary tests clearly measure *breadth* of knowledge, or how many words are known to some basic degree. Is one more important than the other in language learning? Fortunately, the evidence so far suggests breadth and depth are strongly related. Vermeer's (2001) study of 4 and 7 year old Dutch children showed high correlations between breadth and depth measures for both L1 and L2 children. We already considered words as nodes in a highly interconnected lexical network. Vermeer (2001:218) argues that "the denser the network around a word, the richer the set of connections around that word, the greater the number of words known, and the deeper the knowledge of that word." The more words a learner knows, the better each individual word's meaning is delineated from other words it is connected to. Similarly a word's exact aspects of meaning can only be known in the context of other words which are related to it. Thus, increased breadth in vocabulary knowledge coincides with increased depth, and vice versa, allowing us to side-step the depth issue in applications and use common breadth measures of word knowledge instead.

We now consider the order in which words should be studied to maximise the impact of the words acquired.

Ordering strategies

Given authentic input, most learners quickly run into far more unknown words than they have time to study. Since time and attention are finite resources, they should put effort into ensuring the words that they do study are those which will gain them the most benefit. The cost of learning a word in terms of these and potentially other resources is called its *learning burden* (Nation 2006; Barker 2007). Once acquired, knowing a word also accrues some benefit to the learner: the ability to understand its use in context. This section presents a brief discussion of two main strategies for choosing what words to study next. The first is a focus on compositional aspects of words, the second a focus on frequency of occurrence. We call both *ordering strategies*.

A focus on composition advocates studying minimal words or morphemes before studying the larger words or multiword expressions in which they occur. This is typical of how native Japanese children study kanji, learning basic kanji first, and then their use as components in more complicated characters. It is also supported by the depth-of-processing hypothesis for memory, which suggests that the *depth* (extent of semantic engagement) with which a word is processed determines the success of subsequent recall (Craik and Tulving 1975). Since new words will be mentally linked to their component parts, we expect greater semantic engagement during their study; they should thus be easier to remember than words without such links. This effect was also confirmed by Yamashita and Maru's (2000) study of kanji compositionality and ease-of-learning. We also note that basic kanji often relate to simple concepts, and that studies in English likewise suggest that such concepts are easier to learn (Rosch *et al.* 1976). In general, this strategy advocates a so-called "greedy" approach to studying words, in which learners study the easiest of the unknown words first and thus amass a large vocabulary quickly.

The most common criticism of ordering by composition as a strict strategy is that it advocates delaying the study of common words in favour of much rarer words with known components. Learners may end up with a larger vocabulary, but it may not serve them as well as a smaller vocabulary of more appropriately chosen words. In contrast, our second ordering strategy is to study words or characters in frequency order. In a simple form, fixed vocabulary lists are constructed for learners based the frequency of occurrence of words (or kanji) in some representative corpora of texts. This approach tries to get maximum coverage over new and unseen input by asking learners to study the words most likely to reoccur, regardless of their difficulty. It also assumes both that learners are uniform in their exposure to input, and that words are uniform in the learning burden they incur.

In practice, there may be less competition between these strategies than supposed. Given that each new word presents both a potential burden and benefit, Barker (2007) suggests a more sophisticated combination of these strategies, where learners pay attention to how often they encounter a word which is novel or on the cusp of their knowledge. They can then decide which words to study, trading off a word's learning burden against its benefit, and thus using their time most efficiently.

Regardless of the ordering strategy chosen, the words themselves must still be learned. We now ask how individual words might best be learned.

Acquisition strategies

In Section 3.1, we argued that vocabulary acquisition was largely an autonomous matter. To briefly recap, although the most obvious strategy is to explicitly teach vocabulary in the classroom, and doing so has strong advantages for the words which are taught, student-teacher time is ultimately limited. The sheer amount of vocabulary learners must acquire precludes this as a primary method of study. A consensus has formed that for high frequency highly polysemous words, explicit teaching is useful, but that the vast majority of words will be nonetheless be acquired through autonomous study. We now consider two main strategies for such study: *mnemonics* and *inference from context*.

As we discuss these strategies, we recognise that a large number of variables may affect their success. It is however beyond the scope of this thesis to consider them in full. We instead refer the interested reader to Mohseni-Far's (2008) excellent overview of acquisition variables and strategies, which we draw on in our discussion.

Mnemonics

From an abstract perspective, vocabulary learning is purely a matter of memorising information, so general memorisation methods such as mnemonics are applicable. Mnemonics in language learning are mediation strategies for mentally linking the form of an L2 word with its definition, or with an approximate L1 match (Mohseni-Far 2008). For example, suppose an English speaker wishes to learn the Japanese word 五 *go* "five". Since its pronunciation bears similarity to the English word *go*, they may use this acoustic link to associate movement with the number five, perhaps creating a mental image of five men *going* somewhere. When they wish to recall the Japanese word for five, they remember the mental image they have constructed, and through this link recall its pronunciation as *go*. This technique is called the *keyword method*. Note that the association with movement and the additional details of the mental image are not part of the meaning of the Japanese word, but simply additional details used to aid its recall.

A popular method for kanji study making use of mnemonics is presented in the "Remembering the kanji" series of books by Heisig (1985), and is known informally as the "Heisig method". Three main points define this method. Firstly, Heisig advocates studying a large number of kanji for their meaning only, and ignoring pronunciation. This is a form of ordering strategy, deferring study of speech, with the goal of greatly simplifying kanji acquisition. Secondly, the kanji studied should be studied in compositional order, a strategy we discussed earlier. Lastly, each individual kanji is studied through a mnemonic method, where the meanings of kanji-components are tied with the whole-kanji meaning into a story the learner creates. For example: the kanji 砕 *sai* "smash" could be remembered by the story: "To smash something you should hit it with a *stone* (石) *nine* (九) or *ten* (十) times."[2] This story provides a form of semantic mediation which may aid recall and increase processing depth. There have been no formal studies of the effectiveness of Heisig's approach, but criticisms focus on the trade-off suggested by his ordering strategy; in contrast his mnemonic method is considered to be a useful tool for learners (So 2008).

Mnemonic methods are supported by the *depth of processing* hypothesis, as referred to in our discussion of ordering strategies. By creating additional phonetic and semantic links,

[2] A user example submitted to the site "Reviewing the Kanji", at `http://www.kanji.koohii.com/`.

the word is being processed at an increased depth, and retention is subsequently increased.

Though generally successful, mnemonic methods have several limitations. Firstly, many require sufficient phonetic or orthographic similarity between L1 and L2 words in order that mediating links can easily be constructed. Secondly, they can be difficult to use for abstract words. Finally, they only establish links for one aspect of word knowledge, namely form-to-meaning links in Heisig's case, and sound-to-meaning links in our earlier example 五 *go*. Beyond the chosen dimension, they have little to offer in the acquisition or memorisation of the many other aspects of word knowledge discussed earlier.

Inferring from context (IFC)

Mnemonic methods are a way of adding depth to the process of memorisation, and linking new words with existing knowledge so as to provide alternative mental access methods for the new information. An alternative to constructing these links "artificially" is to attempt to study words in their authentic context. This way, the text itself provides the related concepts to link to. The "Kanji in Context" series of kanji study books takes just this approach, attempting to provide authentic and rich contexts for each kanji (Nishiguchi 1994). This method is most applicable in open reading, when unknown words are encountered naturally in context. In such a case, learners can increase engagement by actively trying to guess the item's meaning instead of or as well as looking it up. This technique is called inferring from context (IFC), or the "guessing strategy". The underlying principle of IFC, that the meaning of a word is embedded in the many contexts in which it occurs, is well known as the *distributional hypothesis* in Computational Linguistics.[3]

An example given by Walters (2006) is illustrative of the idea: *Typhoon Vera killed or injured 218 people and crippled the seaport city of Keelung*. Suppose that the learner knows every word in the sentence but *crippled*. Then the context constrains the meaning of the unknown word, allowing the learner to ask, what would a typhoon do to a city? Notice also that the meaning is not uniquely constrained by this encounter; many closely related words could work in place of *crippled*, such as *destroyed*, *devastated* or *ravaged*, each with slightly different meanings or usage in other contexts. In other cases, the context may barely

[3] The *distributional hypothesis*, originally proposed by Harris (1954), is that words which occur in similar contexts tend to have similar meanings.

constrain the meaning at all, and guessing may be unhelpful. Nonetheless, over the course of many encounters in sufficiently varied contexts, the learner should be able to gradually build a more complete understanding of word meaning. Part of this understanding may come from learning related words and thereby better delineating each word's meaning.

In its broadest sense, IFC can refer to any situation in which the learner attempts to determine the meaning of an unknown word based on available cues. These could include diagrams, images, topic information, the other words in a sentence, or any other contextual knowledge available.[4] Although IFC as a strategy could refer to many uses and situations, it is usually described as a strategy to aid reading.

The vocabulary size that learners bring to bear in their reading attempts strongly defines their reading experience. As a rough approximation, the relationship between the number of unique words and text size can be modelled as log-normal for alphabetic orthographies.[5] This distribution suggests how often learners are likely to encounter unknown words as their vocabulary increases in size. There will thus be stages in aptitude as a learner progresses from most words in general text being unknown, to one unknown word sense per phrase, to one unknown sense per sentence, then per paragraph, and so on. Even at a level of several thousand lexemes, a learner still falls far short of the vocabulary of a native speaker, and thus encounters unknown words far more often.

IFC is somewhat controversial for second language learners for several reasons. Firstly, inferring accurately requires a large amount of local and global context, which may not be available for every word. Only in rare cases will the word's meaning be given fully and redundantly in the course of the sentence and surrounds; this is usually called *pregnant context*. In normal cases, there may not be enough information to fully constrain the word's meaning, especially if the context itself contains other unknown words. Thus IFC is most accurate for native speakers, and becomes less and less accurate with decreasing vocabulary sizes. In reality, this common argument is not restricted to IFC and L2 learners, but based rather on the amount of existing vocabulary knowledge. As we discussed earlier in Section 3.1,

[4] Normally this strategy refers to inference from context within receptive materials, for example written or audio-visual materials. In dialogues, where a learner is better placed to negotiate meaning and attempt clarification, it is less necessary.

[5] See Baayen (1993) for a comparison of log-normal, inverse Gauss-Poisson and generalised Zipf's "laws" on modelling this distribution.

attempting to read widely with insufficient knowledge may be a difficult and unproductive exercise, causing the learner to enter a cycle of poor reading ability and poor motivation which is difficult to break from.

Secondly, Dycus (1997) argues that unknown words are typically rarer in usage than known words, hence carry more information (by information-theoretic measures) than the known words, and are thus harder to guess. Unknown words are certainly likely to be rarer than known words: if we assume that learners build knowledge about new words through encounters with those words, a corollary is that unknown words are in general those encountered the least. For this reason, their exact meaning is indeed likely to be rarer. However, Dycus assumes that learners must guess the exact meaning. Returning to our earlier example, the words *crippled*, *destroyed*, *devastated* and *ravaged* would all have similar meaning in the provided context, only differing in slight and subtle ways. On first encounter, a learner would not be able to determine these nuances, but could still narrow down the word's potential meaning to some broad sense shared between these words, an abstract concept of damage and destruction. Subsequent encounters or some additional source of information are needed for learners to acquire the new word's full meaning. IFC remains useful as a technique for gradually refining a new word's meaning until the learner's understanding approximates that of the L1 population.

In addition to IFC or as an alternative to IFC, unknown words can be ignored or they can be looked up in a dictionary. If ignored, the learner suffers the resultant penalty to comprehension. If a dictionary is used, the learner suffers from the costly time-delay before meaning is provided. The longer the delay, the less likely the learner is to continue to retain the earlier context in their working memory, making swift dictionary lookup important for comprehension. In our earlier example, dictionary lookup of *crippled* after guessing would also have allowed the learner to pick up differences between it and its near-synonyms.

In advocacy of IFC, Fraser's (1999) study showed that inferring meaning gave similar vocabulary retention to dictionary lookup, although accuracy was lower. The highest retention strategy was to guess the word's meaning, then confirm the meaning by dictionary lookup; the deeper processing of the combined approach explains the higher retention rate. Mondria (2003) performed later experiments where learners were given pregnant contexts so that inferring an incorrect meaning would be unlikely, and found instead that inferring and lookup had similar retention rates, despite the use of a verification step in the IFC strategy. It may be that semantic engagement is increased by the process of reconciling an incorrect guess with an authoritative word definition in the verification step.

The partial successes of the guessing strategy could be explained by its effectiveness where there is a large amount of beneficial L1 transfer, for example, between languages that share Latin as a common ancestor. In Japanese many loanwords are borrowed from English or other European languages and written in katakana. Guessing the meaning of a loanword may be effective when the source word is in the learner's L1, and phonetic and semantically similarity to its source word overlap. However, in many cases the strategy will still have difficulty: the source word could be from another language; the nearest phonetic neighbour could be a different word; the loanword or the source word could have drifted in meaning from the time when the loan originated; or, the word could be an foreign loanword construction not used outside Japan.[6]

Suppose that no beneficial L1 transfer is available to aid IFC. Then the effectiveness of IFC is supported by a compositional ordering strategy for learning words, since such a strategy will allow learners to utilise the internal cues given by a word's components as well as its context in order to infer its meaning. More broadly though, we take the position that many of the difficulties in applying IFC relate to the lack of an adequate base vocabulary, relative to the text being read.

To summarise, learners are typically unable to use guessing effectively until they reach very large vocabulary sizes, except in the presence of strong, beneficial L1 transfer. They thus rely on dictionary lookup heavily to aid them through their reading until they reach a level where they can utilise this technique usefully. Any improvements we can thus make to dictionary lookup will save much time for these learners, and may also help to maintain motivation to read.

[6] For example マイナスドライバー *mainasudoraibā* "flathead screwdriver", from "minus (screw)driver". Such constructions are known as 和製英語 *wasei eigo* "Japanese-made English".

Self-study tools

A vast number of vocabulary and kanji study tools are available on the web for Japanese, perhaps more so than for other languages. Much of this richness is attributable to the availability of the EDICT Japanese-English dictionary[7] constructed by Jim Breen in 1991, and the matching KANJIDIC dictionary of kanji meanings.[8] These resources allowed developers of study tools and dictionary interfaces to focus on user interaction rather than on lexicography. For a broader selection of study tools than we can discuss here, we refer the interested reader to Jim Breen's Japanese Page[9] and a recent portal built for this purpose by the Japan Foundation called Nihongo-e-na (日本語eな).[10] German speakers may also be find the Wadoku Wiki a useful resource for Japanese learning.[11]

Ultimately, each of tools in this section has at its core one of the two themes from the previous section, either helping the user to memorise words or aiding them in acquiring them through naturalistic input. We discuss both forms of self-study tool in this section.

Memorisation tools

The first tool we examine is Quizlet,[12] a language independent flashcard tool for memorisation of vocabulary or even general facts. Whilst there are many flashcard interfaces online, most are highly specialised and very limited in scope. Quizlet is interesting for several reasons. It allows users to develop flashcards themselves from their own study materials, to share their flashcard sets with others, and to study a variety of different modes. Its main mode is a learning mode, where the learner must type explicitly the answer to the flashcard, or else postpone answering the question until later in the session. Each session thus constitutes one successful pass through the flashcard set. It also provides a test mode, which can generate tests from the flashcard set using multiple-choice or productive question formats, and two game modes for some variety.

Despite its usefulness as a general learning tool, Quizlet has two main limitations.

[7] http://www.csse.monash.edu.au/~jwb/j_edict.html

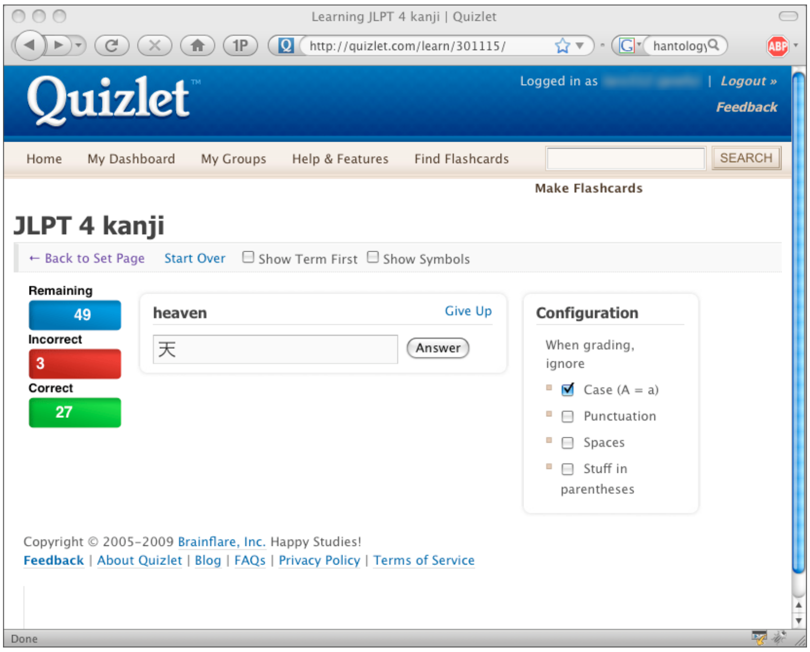
[8] http://www.csse.monash.edu.au/~jwb/kanjidic.html

[9] http://www.csse.monash.edu.au/~jwb/japanese.html

[10] http://nihongo-e-na.com/

[11] http://www.wadoku.de/wiki/display/WAD/WadokuWiki

[12] http://quizlet.com/



JLPT 4 kanji

□ Show Term First □ Show Symbols

Remaining 49
Incorrect 3
Correct 27

heaven
天

Configuration

When grading, ignore
☑ Case (A = a)
□ Punctuation
□ Spaces
□ Stuff in parentheses

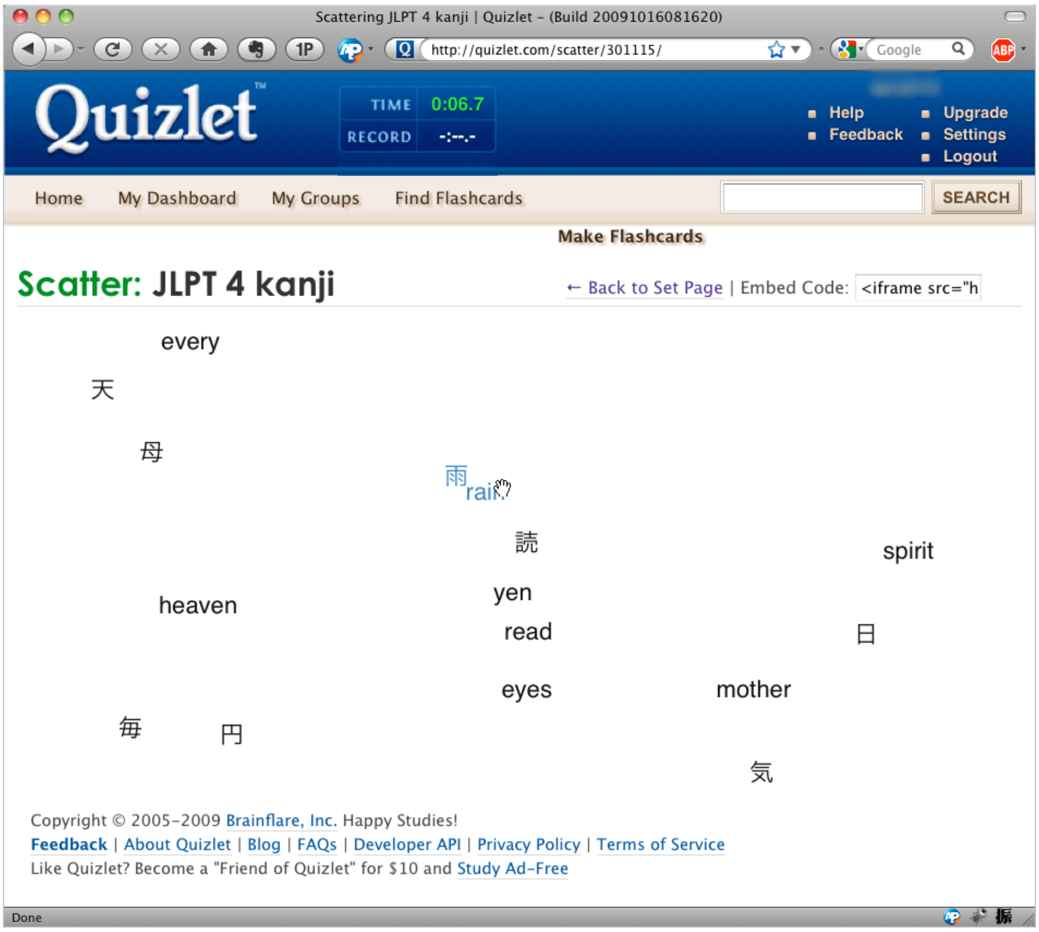
Copyright © 2005–2009 Brainflare, Inc. Happy Studies!
Feedback | About Quizlet | Blog | FAQs | Privacy Policy | Terms of Service

Figure 3.2: The basic study mode of Quizlet, shown part way through study of the JLPT 4 kanji set. The JLPT 4 kanji list was entered by another user.

Firstly, we know that word knowledge is multi-dimensional, yet the general-purpose nature of its flashcards limit users to studying a single dimension of knowledge at a time. This criticism applies to the flashcard paradigm as a whole, although other more specialised flashcard sets – for example the paper flashcards developed by White Rabbit Press for kanji study[13] – offer far richer linguistic aids which might help learners to acquire broader aspects of word knowledge. Secondly, for all its basis in memorisation Quizlet doesn't provide support for spaced repetition in any form, thus undermining its usefulness in developing long-term retention.

We discussed earlier the Heisig method as advocating a particular form of mnemonic, where a story is constructed for a kanji involving either aspects of its form or its component radicals. An innovative site supporting this study method is the Reviewing The Kanji site.[14] Heisig (1985) advocates learners developing their own mnemonic stories, however this can sometimes be difficult if the elements a learner must include in these stories seem unrelated.

[13] `http://www.whiterabbitpress.com/catalog/Flashcards-orderby0-p-1-c-248.html`
[14] `http://kanji.koohii.com/`



Scattering JLPT 4 kanji | Quizlet - (Build 20091016081620)

http://quizlet.com/scatter/301115/

Scatter: JLPT 4 kanji

← Back to Set Page | Embed Code: <iframe src="h

Copyright © 2005-2009 Brainflare, Inc. Happy Studies!

Feedback | About Quizlet | Blog | FAQs | Developer API | Privacy Policy | Terms of Service

Like Quizlet? Become a "Friend of Quizlet" for $10 and Study Ad-Free

Done

Figure 3.3: The "Scatter" game in Quizlet, where users must drag and drop matching elements of word-gloss pairs onto each other to eliminate them. The user is scored by their time taken, which is compared to their previous best.

Reviewing the Kanji allows learners to share mnemonic stories and vote on them, thus providing a very useful aid to this form of memorisation.

Reviewing the Kanji uses a form of spaced repetition developed by Leitner (1972) and discussed later by Mondria and Vries (1994), where a number of bins are used to store the different concepts being memorised (Figure 3.4). Each bin is used in a self-test, where a successful recall causes the concept to advance to the next bin, and a failed recall moves the concept back to the starting bin. Each successive bin represents an increasing timeframe before the next self-test: for example, the succession 1 day, 3 days, 1 week, 1 month, 6 months could be used. This system reduces the amount of time spent on successfully learned words, and yet also revisits them in sufficient spacing to promote their long-term accessibility in memory.

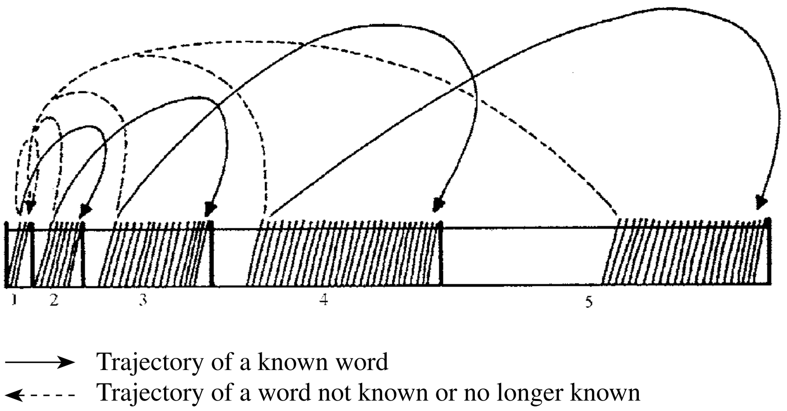

1
2
3
4
5
Trajectory of a known word
Trajectory of a word not known or no longer known

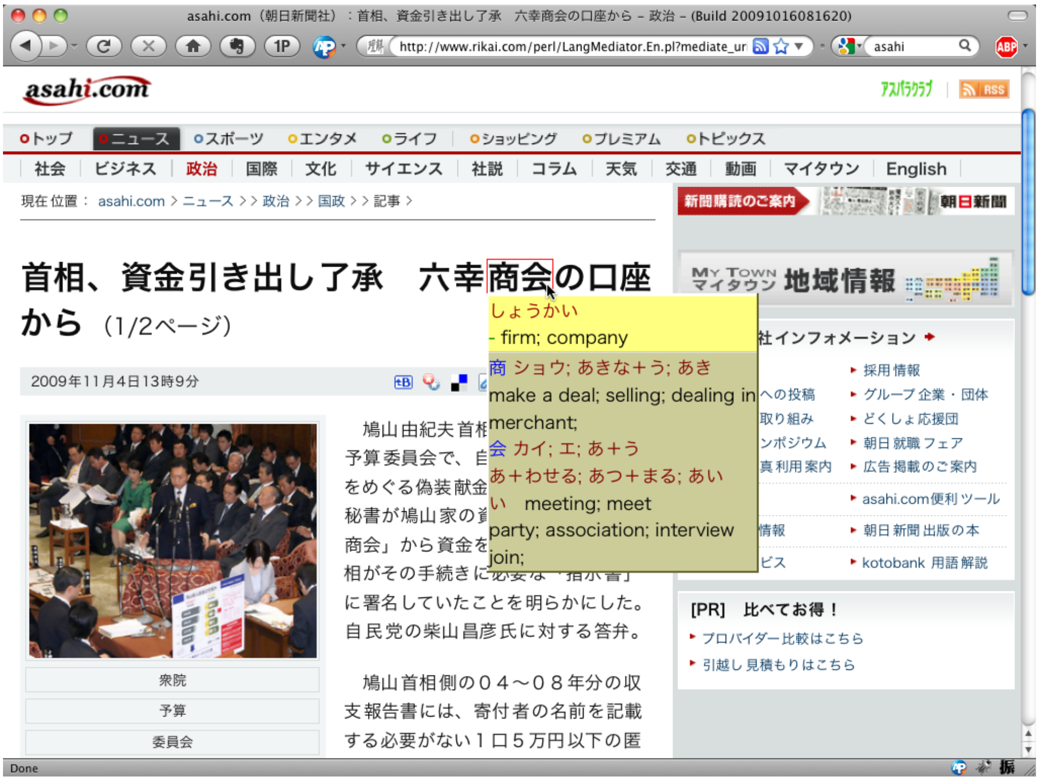
Figure 3.4: The card system proposed by Leitner for spaced repetition. From Mondria (2007).

Reading aids

In contrast to flashcard tools, the last two systems we consider aim to promote success in reading by augmenting computer-based texts with dictionary glosses. This approach is comparable to preparing a graded reader where rare or out-of-syllabus words are pre-glossed, but preparing text in this way using a manual annotation system such as JGloss[15]

[15] `http://jgloss.sourceforge.net/`

is time consuming. Instead, the PopJisyo[16] and Rikai[17] sites both provide a manner of pre-processing an arbitrary Japanese text, and loading it with pop-up translations for all recognised words. Since the translations only pop-up if the user places their mouse over the word, they are not obtrusive and distracting for readers who already know a word's meaning. However, when a word is unknown, the effect is that of instantaneous dictionary lookup. Figure 3.5 shows such an example where Rikai is used to aid reading of a Japanese news article. A similar system is provided by the the Reading Tutor toolbox,[18] which prepares on-demand translations in a right-hand column next to the text.


asahi.com（朝日新聞社）：首相、資金引き出し了承　六幸商会の口座から – 政治 – (Build 20091016081620)

http://www.rikai.com/perl/LangMediator.En.pl?mediate_ur

現在 位置： asahi.com > ニュース > > 政治 > > 国政 > > 記事 >

首相、資金引き出し了承　六幸商会の口座から（1/2ページ）

2009年11月4日13時9分

しょうかい
- firm; company
商 ショウ; あきな+う; あき
make a deal; selling; dealing in
merchant;
会 カイ; エ; あ+う
あ+わせる; あつ+まる; あい
い　meeting; meet
party; association; interview
join;

鳩山由紀夫首相
予算委員会で、自
をめぐる偽装献金
秘書が鳩山家の資
商会」から資金を
相がその手続きに必要な「指示書」
に署名していたことを明らかにした。
自民党の柴山昌彦氏に対する答弁。

鳩山首相側の０４～０８年分の収
支報告書には、寄付者の名前を記載
する必要がない１口５万円以下の匿

Figure 3.5: Rikai used to aid reading of a news article from Asahi Shimbun. When the mouse is hovered over the word 商会, its gloss "firm; company" is displayed, as well as an individual gloss for each kanji in the word.

For all of these systems, the lack of a significant time-cost penalty for unknown words could have several undesirable consequences, in particular over-reliance on and abuse of

[16] http://www.popjisyo.com/WebHint/Portal_e.aspx
[17] http://www.rikai.com/perl/Home.pl
[18] http://language.tiu.ac.jp/index_e.html

the dictionary as a learning tool. Critics of paper-based dictionaries charge that excessive dictionary use leads to reliance on word-by-word translation, potentially ignoring multiword expressions, idioms and issues of lexical gridding (Walz 1990; Bell 1998; Prichard 2008). Furthermore, with instantaneous lookup the incentive to learn a word may be too small for a learner to invest the time and attention required to learn it. Nonetheless, they allow learners of a much greater variety of proficiencies to read authentic texts of their choosing, and are thus potentially a strong aid in facilitating the virtuous circle of successful reading.

A second criticism is that they provide little support for reading-for-vocabulary, in the sense that most learners will find too many new words to keep track of, with little to differentiate which new words should be further studied. Learning a new word takes time, but the payoff in doing so varies according to the word's frequency of occurrence. Of the three systems only Reading Tutor provides cues as to frequency, by colouring words in the source text by their difficulty according to Japanese Language Proficiency Test (JLPT) levels, and none of the systems provides a more integrated user-oriented lexicon based on actual exposure to words.

Finally, these systems are limited by the media the learner is trying to read, and offline or scanned texts without optical character recognition are still beyond the reach of these reading aids. For such media, the learner must consult a dictionary system.

3.2 Structure of the mental lexicon

Having made our case for supporting autonomous study of vocabulary, and having discussed how words are chosen for study and then learned, we now consider how we might improve the learning process. We contend that if we more closely model how learners make errors, particularly during reading, we can vastly improve the tools which support their vocabulary acquisition. For this reason, we now consider contemporary models of the mental lexicon.

Such models are important for two main reasons. Firstly, they are suggestive of how to best populate the lexicon with new L2 words, which is indeed the focus of this thesis. Secondly, the relationships between words, especially those which emerge due to the lexicon's structure, might be leveraged to support vocabulary acquisition in a variety of applications.

As writing systems vary significantly as to what information they encode (Koda 2007), our investigations of graphemic similarity will inevitably be writing-system specific. In our discussion, we try to maintain a multilingual perspective where possible, focusing on Japanese only where necessary.

Visual word recognition

We learn about the structure of the mental lexicon primarily through studying visual word recognition, with the caveat that we are considering the ultimate structure for skilled L1 readers. Models of visual word recognition have to account for many known psycholinguistic effects if they are to be successful (Taft 1991:11; Handke 1995:166). Of these, Lupker (2005) suggests that the most important are the *word superiority*, *word frequency*, *semantic priming* and *masked repetition priming* effects.

In general, proposed models fall into one of two categories: *search models* where some autonomous search process identifies candidates from sub-lexical features and proceeds to verify them until a match is found, or *activation models* where continuous activation occurs between various levels of processing, and the word perceived is that with the strongest activation level (Lupker 2005). In this section we focus on two variants of the basic interactive-activation model, which we show in Figure 3.6, and which address each of these key effects appropriately.

Note that we focus on the word level in isolation, rather than considering larger units such as phrases or sentences. As a receptive model, we model the process initiated by visual input and concluded by successful access to a word. This can be compared with productive models, for example that provided by Levelt *et al.* (1999) for speech production. Though not shown on Figure 3.6, variants of the interactive-activation model posit an intermediate level between the letter and word levels which activates related phonological units. If we are to consider Chinese or Japanese characters, then radicals seem to be appropriate sub-word units to occupy this level.

Saito *et al.* (1998) adapted this model to Japanese, developing the Companion Activation Model (CAM) shown in Figure 3.7. The CAM does indeed use radicals as intermediate sub-word units, however above radicals and below words lies the additional whole-

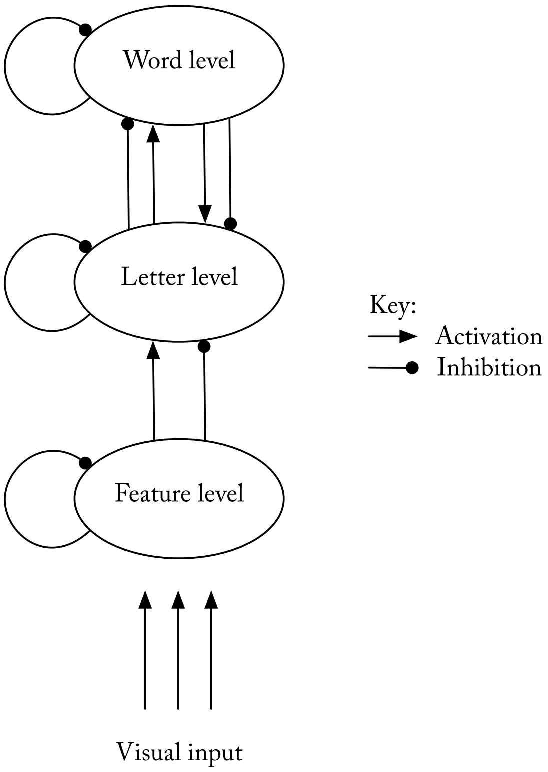


Figure 3.6: The multilevel interactive-activation model for alphabetic orthographies. From McClelland and Rumelhart (1981).

kanji level. They also assume two types of activation (foreground and background) which interact with one another. When a kanji radical is activated, in combination with other foregrounded radicals it activates whole-character matches in the foreground, and visual neighbours which also contain the radical ("companions") in the background. Their results in several experiments are interpreted to mean that phonetic properties of radicals are automatically activated. This matches the results of earlier experiments by Flores d'Arcais *et al.* (1995), where pre-exposure to a phonetic radical reduced latency in a naming task.

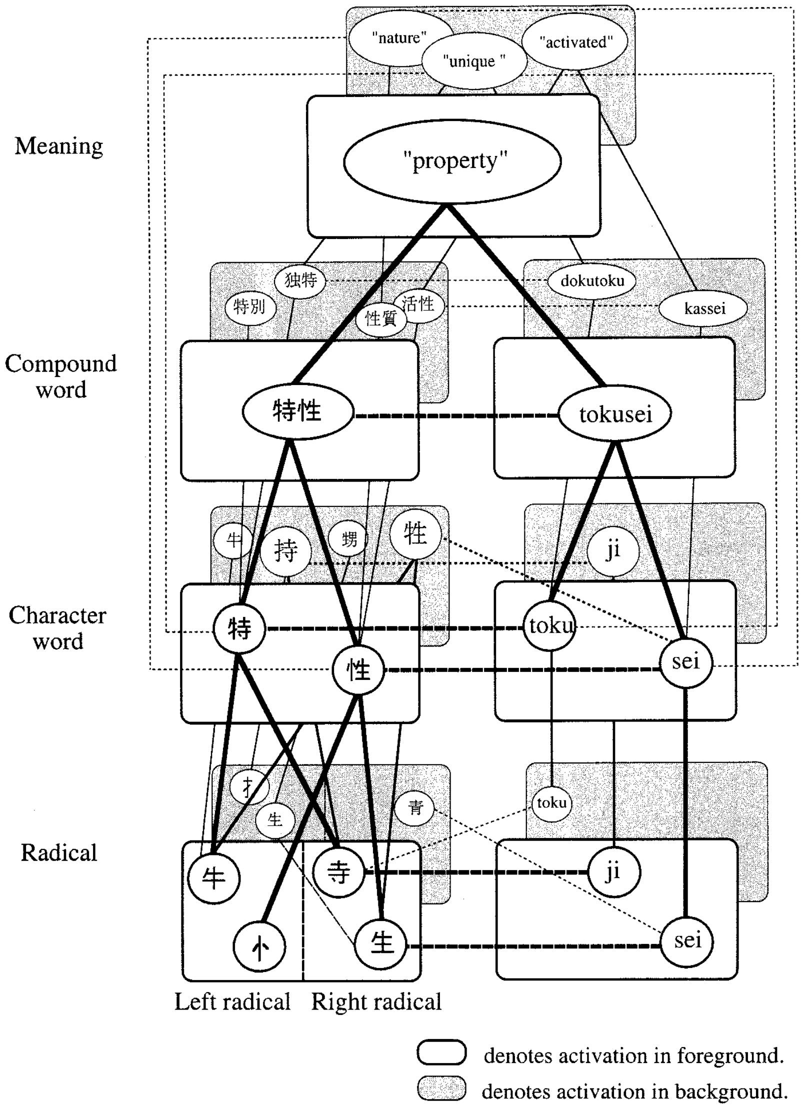


Figure 3.7: The Companion Activation Model (CAM). From Saito *et al.* (1998).

Although Saito *et al.*'s (1998) model only considers kanji split into left and right components – roughly 53% of the JIS standard set – it could easily be extended to other kanji shapes: only left-right kanji have phonetic radicals with any reliability, so for other shapes we would assume that radical-level phonology is not activated. It also provide explicit suggestions of other candidates which are considered in the recognition process, thus implicitly suggesting potential error outcomes should misrecognition occur. Unfortunately, the CAM provides little explanation of the role of multiple scripts in word recognition. It also suffers from problems of "representational redundancy, homographs, and varying degrees of semantic transparency" (Joyce 2002:80). Motivated by these problems, an alternative interactive-activation model was constructed by Taft *et al.* (1999a), and adapted for Japanese by Joyce (1999).

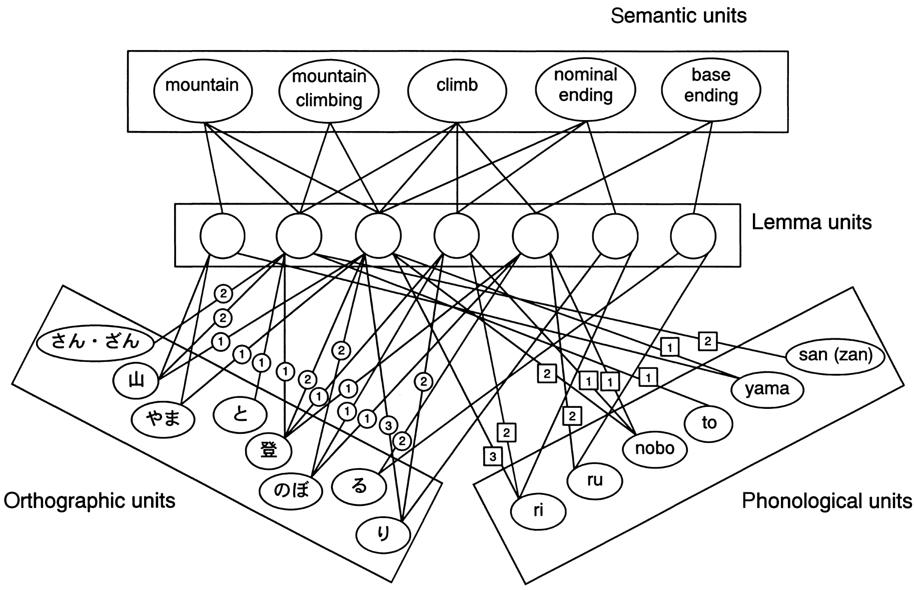


Figure 3.8: Lemma unit connections in Joyce's (2002) multilevel interactive-activation model for Japanese.

This model is very similar to CAM in its broad structure, in that activation flows from the lower levels upwards through to meaning, however lemma units are added which mediate between orthography, meaning and phonology (Figure 3.8). These abstract units roughly represent the morpheme level, and they contain additional information about their links to orthographic and phonetic units which allows them to order their connections. Amongst

Link type	**Intralingual**	**Interlingual**
Semantic	(near-)synonymy/antonymy hyponymy meronymy holonymy entailment	(near-)synonymy cognate
Phonetic	(near-)homophony	(near-)homophony
Graphemic	(near-)homography	(near-)homography

Table 3.3: Types of lexical relationships available to the ideal bilingual speaker.

its other advantages, this model provides clearer integration of the multiple scripts used in Japanese.

Each of these models makes heavy use of the hierarchical nature of kanji, which is unsurprising, given that such structure exists. Activated candidates during the reading process are suggestive of potential erroneous outcomes if a word is misread. They also give us insight into how words may become related through the recognition process. For example, kanji sharing radicals may be commonly activated as candidates when processed through reading, and this common activation may develop a bond between these characters. We now consider these and other relationships which form between words.

General lexical relationships

Our discussion of Japanese-specific aspects of the mental lexicon was dominated by the processing of kanji, and kanji compounds. In this section we consider more broadly the relationships between words which occur within any language, and indeed interlingually between words of different languages. Ultimately, quite a large inventory of such relationships can be uncovered, as shown in Table 3.3.

As is clear from Table 3.3, the richest variety of relationships is semantic and available monolingually, i.e. between words of the same language. For semantic relationships within English the primary resource is WordNet, a semantic network supporting several relations (Fellbaum 1998), following earlier work on lexical relations by Cruse (1986). Firstly, words are grouped into synonym sets ("synsets"). For nouns, *hyponymy* (the is-a/kind-of relationship) and *meronymy* (the part-of relationship) are provided; for verbs, *hyponymy*, *troponymy* (the manner-of-doing relationship) and *entailment* are available. For Japanese, the Goi-Taikei project (Ikehara *et al.* 1997) also provides a hierarchical semantic lexicon, using a subset of the relations available for the WordNet project. More recently, a Japanese version WordNet was constructed using the synsets from the English WordNet as its base (Isahara *et al.* 2008).

We refer to semantic links in general as *associations*, and our listing of association types should not be considered exhaustive. These rich semantic relations are still insufficient to capture the semantic proximity of pairs such as *doctor* and *patient*. For this reason, work on circumventing the tip-of-the-tongue problem (Zock 2002; Zock and Bilac 2004) has more recently focused on extending WordNet with syntagmatic relationships (Ferret and Zock 2006). Other relationships may also exist: for example, recent experiments by Gaume *et al.* (2008) comparing child and adult learners of French found a salient semantic relationship between verbs based on co-hyponymy. Overall, sufficient proficiency in any language yields associations between words which are shared across speakers, and these can be captured by studying sufficient numbers of speakers, as in the case of Joyce's (2005) large-scale database of such associations for Japanese.[19]

If we now consider interlingual word relationships the most obvious is synonymy, or more accurately near-synonymy. Every bilingual dictionary attempts to match words in one language with their closest synonyms in another. If we consider some abstract space of all possible meanings, then each language partitions and covers this space with words in a different manner. This has been called *lexical gridding*, with the idea that the "grids" of two languages often do not "line up" (Laufer-Dvorkin 1991:16). For example, in English we describe the middle front of a creature's head as either its nose (for people), muzzle (for dogs), snout (for pigs), or trunk (for elephants); in Japanese, the single word 鼻 *hana* is used in all these contexts. In more complex examples, words only partially correspond to one-another by overlapping in a few of their available senses, but not in others. Ploux and Ji (2003) visualise this situation by constructing two-dimensional semantic maps of near-synonymous English and French words. Lexical gridding effects in general are not limited

[19] `http://www.valdes.titech.ac.jp/~terry/jwad.html`

to denotational meaning, but cover all aspects of word knowledge, as we shall later discuss (Section 3.1).

When L2 words sound like words from our L1, or are written like words in our L1, we associate them with these words, regardless of semantic differences. Indeed, contemporary product names are chosen carefully to avoid negative associations with consumers from different linguistic backgrounds. When two languages share some common ancestry, word pairs will exist between these languages which share the same etymology. These word pairs are *cognates*, and due to the slow speed of linguistic change, these pairs are often near-synonyms, near-homographs, and near-homophones.

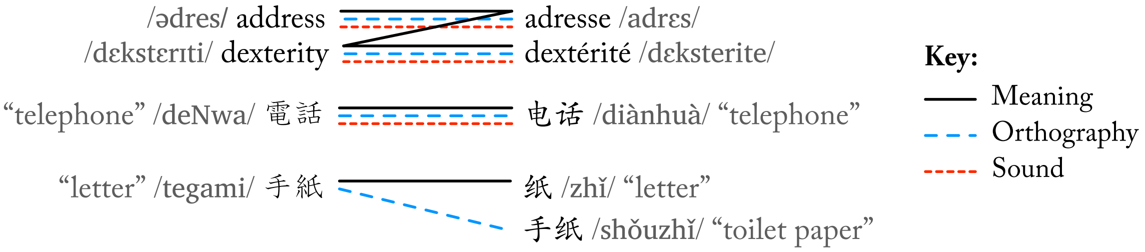


Figure 3.9: Interlingual relationships based on near- homophony, homography and synonymy. The first example is English-French; the second and third are Japanese-Chinese.

We can bring together many of these ideas with the examples shown in Figure 3.9. In the first example, the French word *adresse* is cognate with the English *address*, and shares the meaning “postal address”. However, it also means “dexterity”, which itself has a matching cognate *dextérité* in French. Both cognate pairs show correspondence in meaning in at least one sense of each word. Importantly, the cognate relationship is predictable from the orthographic and phonetic similarity. In our second example, a similar situation exists between compounds in Chinese and Japanese; again the cognate relationship is predictable and reliable. In our final example, the cognate relationship between 手紙 and 手纸 exists between characters at the morpheme level, but not at the whole-word level. The subsequent mismatch in whole-word meaning is significant, and in such cases we call the pair *false friends*, indicating the problems they cause for learners.

Altogether, these relationships transform our view of the mental lexicon from a series of isolated words to a densely interconnected graph. Lexical resources attempt to represent a subset of the vertices and edges in this graph. In Japanese and Chinese, their rich orthographies provide substantial graphemic relationships based on near-homography which have not yet been fully explored. These relationships could provide some important structure for learners, and deserve more attention.

Graphemic neighbourhoods: errors and associations

If we measure the physical similarity of two written symbols, for example using metric measurements or on-screen pixels, we expect that the closer the two symbols the more likely they are to be confused. This is trivially true at extreme levels of proximity, but how the confusability changes as the distance between symbols increases depends strongly on the psychological reality of how the symbols are perceived. It is precisely this reason we looked at the word recognition process for Japanese earlier in this section. The general visual neighbourhood around a Japanese character – presumed to be shared between native speakers – remains unknown, though several experiments give useful indicators as to the writing system's topology.

The concept of a visual neighbourhood is important because a word's neighbours are assumed to be competing candidates during the word recognition process, precisely because they share salient visual features with the target word. Applications wishing to predict visual recognition errors should thus focus on visual neighbours as error candidates. Our previous discussion considered what lexical relationships might be consciously available to readers, but these conscious graphemic associations may be affected by visual recognition processes. We firstly discuss these processes, before considering evidence for the topology of such neighbourhoods.

Accessibility of graphemic neighbours

L2 learners and bilingual speakers clearly associate visually similar words interlingually, usually as cognates or mnemonic aids. However, the extent to which these high-similarity graphemic relationships are readily accessible within a single language is unclear because of conflicting theoretical predictions from lexical competition and global activation effects.

Lexical competition suggests that in processing the stimulus word, visual neighbours of that word will be inhibited (Figure 3.6). In contrast, global activation suggests that the increased activation in that region will increase the activation in neighbours of the stimulus as well. In fact, evidence from Carreiras *et al.* (1997) suggests that both can occur and are task-dependent: orthographic density about a word facilitated naming and lexical decision tasks, but had an inhibitory effect in a progressive demasking task.[20]

If the stimulus word is previously unseen by the reader, then several possibilities are raised. Lexical retrieval will fail, so we might expect inhibitory effects at the word-level to reduce or disappear, thus making graphemic neighbours more consciously accessible. On the other hand, Gaskell and Dumay's (2003) study of spoken word recognition suggested that the first few exposures to non-words can activate the nearest known word. This result seems intuitive: for example, if a new word *epple* was coined in English and presented as stimulus, we might expect readers to mentally access its neighbour *apple*. This could occur regardless of whether they thought the new word was a spelling mistake or not. If this effect also occurs in visual word recognition, inhibition effects could occur from the closest neighbour. However, we cannot examine these effects more carefully without defining our notion of neighbour more carefully.

Graphemic neighbourhood topology

Studies of visual word processing in alphabetic orthographies have long been interested in how the behaviour of such processing changes in response to many variables, including visual neighbourhood density. For this reason, Landauer and Streeter (1973) introduced the N metric of visual density around a word. N is calculated as the number of words which differ from the original in one character only, implicitly defining this as the neighbour criterion. However, the many applications for orthographic similarity – including optical character recognition (Taghva *et al.* 1994), record matching (Cohen *et al.* 2003), spelling correction (Wagner and Fischer 1974) and others – have developed far richer string distance

[20] Progressive demasking is a variant of the naming task where a word (e.g. "ladder") is alternated with a visual mask (e.g. "######") on-screen. First the mask is displayed, then it is replace very briefly by the word. The mask and word alternate, and in each cycle the mask reduces in duration and the word increases in duration. The participant must read the word out aloud as soon as they can identify it, and the time taken for them to do this is recorded.

metrics than this. These richer metrics potentially provide the means to model much more closely the psychological reality of word perception, however the expense of human trials and lack of data on human similarity perception makes their evaluation difficult. Furthermore, choosing between them on a theoretical basis is difficult when many may use features of unknown cognitive salience.

The case for Japanese and Chinese is similar, in that the psychological literature defines only a basic orthographic neighbourhood relationship, where neighbours of a kanji may have one radical from the original kanji swapped, or identical radicals but different layouts (see Saito *et al.* (1998) or Taft *et al.* (1999b)). However, the space of broader kanji distance metrics remains impoverished in Japanese for two reasons. Firstly, many applications of such metrics in English have no equivalents in Japanese. For example, computer input is mediated by an IME and the types of mistakes made do not equate directly to spelling errors that could use such distance metrics for correction.[21] Secondly, unlike words in alphabetic orthographies whose layout is constructed linearly of symbols, kanji have a complex, nested two-dimensional layout, limiting the ability of abstract string distance metrics to transfer to the kanji distance problem.

So how might we best go about measuring and predicting the psychological reality of kanji similarity? Visual search tasks, such as the one described by Yeh and Li (2002) for Chinese, provide a good start. In the task, native Chinese speakers had to determine if a target character was present amongst distractors. Yeh and Li found that shared structure between target and distractors provided the strongest interference, slowing decision times. They investigated two types of characters: those split into left and right components, and those split into top and bottom components. Shared radicals also slowed decision times, but only where the broad layout was also shared. This suggests that the layout of a characters into broad components is dominant initial feature used in kanji perception, and that components are considered as secondary features. This makes intuitive sense, since most kanji are covered by a handful of structural variants. For example, the popular SKIP indexing scheme uses a choice between four basic structures as its primary index feature (Halpern 1999). We discuss such indexing schemes in detail in Section 3.3 which follows.

[21] Common mistakes when using an IME include incorrectly choosing the first kanji compound displayed in the IME, when the correct compound appears later in the list.

Despite this high-level information on neighbourhood topology, how these neighbourhoods are structured remains open. Furthermore, applications cannot make use of this information until at least some of this structure is resolved.

3.3 Dictionary lookup of Japanese

Lookup for producers

Producers of language – speakers or writers – begin with a meaning in mind, and are looking for the right words to use to express that meaning. Even in a monolingual context, finding the right word can be difficult, whether the problem is one of limited knowledge or simply one of access, as in the case of the tip-of-the-tongue problem.

Access based on meaning (onomasiological search) requires firstly expressing that meaning somehow. This is done almost exclusively through other semantically proximate or related words. We discussed types of semantic relationship earlier in Section 3.2; each of these relationships defines a method of semantic access for producers, and indeed resources providing these relationships form lookup methods themselves. In monolingual context, the user specifies the meaning by using a near-synonym; this leads to the traditional supporting resource, a monolingual thesaurus such as Roget's International Thesaurus (Roget and Chapman 1977). In a multilingual context, the learner can specify the meaning in their L1; the traditional resource is thus an L1-to-L2 dictionary, making available the same synonymy relationship but between bilingual word-pairs. Occasionally these two resources are merged, as in the "Hebrew-English-English" dictionary described by Laufer and Levitzky-Aviad (2006).

The main exception in the dictionary space is the use of *semagrams* by Moerdijk *et al.* (2008) in the Algemeen Nederlands Woordenboek, an online scholarly dictionary of contemporary standard Dutch currently under construction. A semagram is similar to a semantic frame, but serves as a formalised structure for the meaning of a word rather than for an event or state of being. Each word sense is a member of one or more semantic classes, and each class has a type template defining which slots are available and values these slots can take. The term semagram refers to a type class with values. For example, the semagram for *cow* has slots for size, colour, build, function, and many other aspects of meaning, based on its upper category as an animal and the type template for animals. This rich semantic encoding for dictionary entries provides a correspondingly rich semantic search which can be performed across many aspects of meaning.

Lookup for receivers

Receivers trying to decode a word into meaning rely on looking up either its orthographic form or its pronunciation in a dictionary. For languages which are orthographically shallow, these two are roughly interchangeable; a transcribed pronunciation in such a language will match the correct word spelling or form. For deep orthographies such as Japanese with complex form-to-sound relationships, form and pronunciation are markedly different sources of information to draw on. In this section we focus on lookup for Japanese and Chinese.

For these languages, lookup is also the mirror-image of the input problem. If the character or word can be input quickly and accurately, lookup is trivial. If a method allows quick and accurate lookup, it likewise has potential uses as an input method. The problem of input and that of lookup by form are thus equivalent. Where input methods may be circumvented, for example by copying and pasting a problematic word from an electronic text into a dictionary, lookup is also trivial – hence the availability of the reading aids discussed earlier. We also saw earlier that input by form is non-trivial, and discussed some solutions from the input perspective. We now consider the lookup perspective of the same problem by examining three different kanji indexing schemes, as shown in Figure 3.10.

Traditional paper dictionaries contain three parts: an index of primary radicals ("section headers") ordered by stroke count, a per-primary radical index of characters which use it indexed by remaining stroke count, and the dictionary entries themselves. Learners can easily select the wrong component as the primary radical, or incorrectly count the number of strokes, and thus have much trouble finding the character they seek. The multiple indexing steps are also time consuming, even without mistakes. The conversion to electronic dictionaries eliminates the time it would take to turn pages in the paper dictionary, but still requires the visual scan of the original dictionary.

明

Traditional

1. IDENTIFY: 日 as section header
2. COUNT: strokes in 日, finding 4
3. LOOKUP: 日 in 4-stroke radical index
4. COUNT: remaining strokes, finding 4
5. LOOKUP: page no. for 明 in 8-stroke characters containing 日
6. LOOKUP: 明 at given page no.

SKIP

1. IDENTIFY: shape as type 1 (left-right ⿰)
2. COUNT: strokes in 日, finding 4
3. COUNT: strokes in 月, finding 4
4. LOOKUP: 明 at index 1-4-4

Kansuke

1. COUNT: horizontal strokes, finding 6
2. COUNT: vertical strokes, finding 3
3. COUNT: other strokes, finding 1
4. LOOKUP: 明 at 6-3-1

Figure 3.10: Kanji lookup methods to find 明 *aka* “bright”. Steps beginning with “IDENTIFY” or “COUNT” are potential error sites for learners. Steps beginning with “LOOKUP” involve use of an index, and subsequent visual scan for the desired item.

The SKIP (System of Kanji Indexing by Patterns) system of lookup provides an alternative indexing scheme based on a kanji's overall shape (Halpern 1999). Each kanji is indexed by a three digit code. The first digit represents its broad shape: ⿰ left-right, ⿱ top-bottom, ⿴ exterior-interior or ⿻ other. The remaining two digits are given by the stroke count of the two separated components, except in the "other" case, where they specify one of four sub-patterns and the overall stroke count. For example, 明 *aka* "bright" has skip code 1-4-4. The SKIP system has two main advantages. Firstly, it removes the need to identify a primary radical for the character. Secondly, it uses only one index, ordered on a code which can be determined without consulting the dictionary. This makes correct access about as fast as alphabetic dictionaries for other languages, though mistakes in stroke counting can still impede access.

The Kansuke electronic dictionary by Tanaka-Ishii and Godon (2006) aims to simplify the stroke counting method in order to avoid such problems. Users instead form a three-number code representing the number of *horizontal*, *vertical* and *other* strokes that make up a character. To avoid stroke count ambiguity, complex strokes are split into their component parts. For example, the single hand written stroke ㇕ is counted as two strokes when determining the Kansuke code, one horizontal and one vertical. 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 Japanese-German dictionary constructed by Hans-Jörg Bibiko[22] provides, amongst other lookup methods, a method based on selecting from a large table of components. If a radical such as 日 is selected from the table, a shortlist of matches is presented showing kanji containing this radical, but the table of components is also updated so that incompatible components (i.e. those which never co-occur with the current selection) are removed. By propagating constraints this way, the user can more easily find the kanji they are looking for. Multi-paradigm dictionaries such as JEdict[23] also combine several forms of lookup in a single interface.

Each of these dictionaries provides a complementary approach which can be added to the learner's arsenal. Looking again at Figure 3.10, we can see that in newer systems the

[22] `http://lingweb.eva.mpg.de/kanji/`

[23] `http://jedict.com/`

learner generates a single index key for the kanji, then performs just one lookup using this key, thus saving time.

The trend across these systems is to reduce the amount of assumed or required knowledge to perform lookup. However, this means that additional knowledge is not used, even when it is available. A competing trend is to allow the user to specify partial knowledge whenever possible. This approach is taken by the FOKS dictionary system.

FOKS dictionary

We suggested earlier that if the pronunciation for a word is known, dictionary lookup becomes simple and fast. What of the case where the user only has partial knowledge of the pronunciation? Such a case is common in Japanese, where the correct kanji reading is context-dependent. In such a case, the FOKS (Forgiving Online Kanji Search) system by Bilac (2005) provides a useful lookup method.

Suppose, for example, a user wishes to find 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*. In this case, 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.

FOKS uses an error model to correct for the most significant recoverable errors made by users in their dictionary queries. Incorrect choice of kanji reading accounts for roughly 80% of user errors, according to post-hoc log analysis (Bilac *et al.* 2004), often in combination with other error types. Three other error sources are also significant: incorrect voicing, incorrect gemination, and incorrect vowel length. Voicing and gemination are changes to character-level pronunciation that occur during word formation, as discussed in Section 2.3, and which can be incorrectly applied by learners. The third source of error occurs when a learner mistakes a long vowel for a short vowel, or vice versa. Vowel length errors are expected to be especially prevalent with learners from languages such as English which do not maintain an equivalent vowel length distinction. Like many dictionaries, FOKS also supports the use of wildcards in queries, which offers some help for lookup of partially known words.

3.4 Second language testing

This section provides an overview of the current state-of-the-art in second language vocabulary testing. It is divided into three main parts, considering the role of testing in language study, general testing theory, and finally current testing methods and their limitations. These limitations motivate our work on adaptive randomised testing.

The role of testing

The broad purpose of testing is to make decisions (Murphy and Davidshofer 1998:2). In the context of language learning, the information from tests allows: the appropriate reward for students through grades; the optimal focus of future learning opportunities, for example targeting them to a particular student's needs; and evidence-based evaluation of course structure and teaching methods based on their measured performance.

Any aspect of language that can be learned can be tested, but since our focus is on vocabulary study, a simple answer to the question, "What should we test?" could be *all aspects of word knowledge*. However, to do this even for a handful of words is prohibitive, since the testable aspects of knowledge for any individual word is large. This dilemma is a form of the bandwidth-fidelity dilemma discussed by Murphy and Davidshofer (1998:143) in the context of testing, and attributed to Shannon and Weaver (1949), where the amount of information conveyed is traded off against the accuracy with which it is conveyed. In general, tests bias towards breadth because testing depth is quite hard, given that the connections between words form a significant part of word knowledge (Meara 2009:74). In other words, knowing a word at depth means knowing its relationship to words around it, so testing a word at depth means testing these relationships too.

As this indicates, breadth and depth are now believed to be strongly intertwined, and a comparison of depth and breadth tests by Vermeer (2001) found strong correlation between them. This supports the validity of focusing on breadth in testing, for example in the vocabulary levels tests (Nation 2001:412),[24] provided that such tests are appropriately constructed. In the levels test, learners are tested on words sampled from a variety of frequency bands. By observing tail-off from more frequent bands to less frequent bands, the learner's lexicon size can be estimated.

A related and important consideration is the distinction between receptive and productive aspects of word knowledge discussed earlier in Section 3.1. If nearly any aspect of word knowledge can be composed of both receptive and productive parts, which should we test? Many researchers believe the receptive/productive distinction to be related to depth of word knowledge, and that receptive skills are a subset of productive skills. This view is supported by the fact that subjects score significantly higher on receptive tests than productive tests (Nation 2001), indicating they are easier. As a secondary effect, subjects' scores increase if they are tested in the same manner that they are taught (i.e. productively or receptively). However, productive vocabulary tests are notoriously difficult to construct, since a question might typically require the subject to produce a particular word, but will not reward them for unexpected but correct alternatives (Meara 2009:34). Many early tests constructed this way thus yielded poor reliability, with widely varying scores on retests.

A rethink of productive testing, in the form of the Lex30 test (Meara 2009:37),[25] elicited free word associations in order to obtain a representative sample of learner vocabulary knowledge. The resulting scores, based on the occurrence frequency of the words a learner provided, yield an index for productive vocabulary knowledge, although they do not as yet estimate it directly. By avoiding the issue of requiring a particular answer to be produced by the learner, the reliability of the method is increased. However, it remains unable to test a particular word of interest, instead testing a learner's productive vocabulary as a whole.

Finally, an experiment performed by Mondria (2007) found that studying receptively and productively yielded equivalent *receptive* recall, but that productive study takes longer. She concluded that productive study (and testing) should only take place if productive ability was important.

[24] `http://www.lextutor.ca/tests/`

[25] `http://www.lognostics.co.uk/tools/Lex30/index.htm`

Testing theory

Modern testing theory aims to guide test design so as to improve the various quality attributes of tests, and provides statistical models for constructing and analysing tests. This section provides a brief overview of recent changes in testing theory, and aims to highlight three long running and related trends in testing: firstly, the trend from subjective to objective testing; secondly the trend from estimating "true" test scores to estimating participant ability; and thirdly, the trend from pen-and-paper testing to computer-based testing. For a more comprehensive overview of modern testing theory than this section allows, we refer the reader to Hambleton and Jones (1993).

Two core concepts are central to any discussion of testing: *validity* and *reliability*. A test has *validity* to the extent that it actually measures the underlying attribute intended. For example, a written test might be a poor method of measuring oral language ability and might thus have low validity. Validity can also refer to the logical validity of arguments (decisions) made on a basis of test results. *Reliability* on the other hand is the extent to which each individual's test score is consistent and repeatable – free from noise or error – and is measured over a sample of test participants.

The trend towards objective testing is motivated by the desire to improve the validity and reliability of tests. Subjective tests are tests where the outcome is affected by participant, examiner and marker bias, and these biases in turn reduce test validity. Objective tests are tests which are unaffected by these biases, and they strongly favour question formats with only one correct answer, for example multiple-choice questions or simple scales, thus avoiding the subjectivity of marking more open-ended question types. A side-effect of this trend in testing is that the human element is removed from marking tests, allowing tests to be assessed automatically by computer.

A parallel trend has occurred in the statistical modelling of tests and test responses, a shift from modelling test scores to modelling participant ability directly. This is most evident in the gradual transition from Classical Testing Theory (CTT) to Item Response Theory (IRT). Both of these theories provide different and competing methods of formally assessing reliability. Classical Test Theory models a test score as composed of an individual's "true score" over the test and an error term, which should be minimised. Item Response

Theory instead models each individual question as contributing information about a participant's ability. This information is usually described in the form of an item-characteristic curve, the plot of the likelihood of answering correctly as a function of participant ability (Bull and McKenna 2004:81). Crucially, this per-question modelling allows construction of large item banks which are drawn upon in designing tests to provide a desired level of reliability in ability estimates. Together, the shift to modelling ability and calibrating individual questions raises the prospect of *adaptive tests* where each individual is shown test items which are maximally relevant to determining their ability level. However, adaptive tests were infeasible before the advent of computer-based testing, our final trend under discussion.

Computer-based testing offers many benefits and disadvantages over traditional pen-and-paper testing. For example, provided that tests are objective, marking can be done instantaneously by computer, thus providing direct feedback to participants. The cost of administering tests to large populations is also significantly reduced. The disadvantages include the large set-up costs even when testing small populations, the logistics of computer resource availability and use during test administration, and medium-of-delivery effects (Chalhoub–Deville and Deville 1999). However, computer-based testing in combination with the large item-banks calibrated with IRT makes computer-adaptive testing feasible, since personalised tests can be constructed on-the-fly and in response to an individual's estimated ability so far.

Adaptive tests are simply tests where the questions asked vary for each examinee. If examinees are shown different questions, then the use of the total number of questions correctly answered as a measure of test performance becomes meaningless. Indeed, use of IRT to model item difficulty is usually a prerequisite for this form of testing; in this case each response from the individual yields better information about their ability estimate, which in turn is used to determine which question would maximally refine that estimate. In this way a test can be made more reliable, shorter, or potentially both.

The main problem which computer-adaptive tests still suffer from is one of limited item bank size. This means that even adaptive tests reduce in validity when individuals re-test themselves, since participants will encounter the same questions as they did in earlier tests with high probability. Within a limited domain, that of vocabulary testing, this thesis aims

	⊕ Recall	⊖ Recognition
⊕ Active (retrieval of form)	Translate the L1 word into the correct L2 word	Choose the correct L2 translation for this L1 word amongst distractors.
⊖ Passive (retrieval of meaning)	Translate the L2 word into the correct L1 word	Choose the correct L1 translation for this L2 word amongst distractors.

Table 3.4: Four types of question used by Laufer and Goldstein (2004) in CATSS, each indicating the task the learner must perform. For each condition ⊕ indicates increased difficulty, ⊖ indicates decreased difficulty.

to show that this limitation can be overcome through the use of randomised questions generated on-the-fly.

Vocabulary testing and drilling

Drills are a short, fast, throwaway form of testing in which the act of testing itself forms part of the learning process. If testing is used to make decisions, then drills are used to make immediate and short-term decisions about what to study next. Indeed, this description matches very closely the flashcard software we described earlier, in which this simple form of testing is used. However, flashcards limit drills to simple forms of questions, with ultimately limited linguistic motivation. This section asks, what types of questions are used in testing, and which are promising for automated testing?

A popular form of testing are vocabulary levels tests, as discussed earlier, which test receptive knowledge of words (Nation 2001). In extension to this form of testing, Laufer and Goldstein (2004) describes the CATSS (Computer Adaptive Test of vocabulary Size and Strength) test. Instead of our earlier view of vocabulary knowledge as multidimensional, Laufer and Goldstein considers knowledge of a word to be unidimensional but continuous, ranging from superficial familiarity to the ability to use the word correctly in open speech. Furthermore, she aligns the receptive/productive knowledge distinction along this axis of word proficiency, and considers four scenarios for the test questions, as given in Table 3.4.

Let us use 山 *yama* "mountain" as our example, with English the L1 and Japanese the L2. In the active condition, the L1 meaning "mountain" is given, and the learner has to either recall the L2 form 山 or recognise it amongst distractors. In the passive condition the L2 form 山 is given and the learner has to either supply its L1 meaning "mountain" or recognise amongst distractors (e.g. "city", "hill", "vehicle", "river"). These correspond to the hypotheses that accessing the meaning from the form is easier than accessing the form from the meaning, and likewise that recognising a word is easier than producing it.

CATSS is adaptive in that, having hypothesised these levels of word knowledge, the lower levels of knowledge are only tested if a higher level has failed. Furthermore, the test is structured into levels based on 1000-word frequency bands, which the learner is tested on serially. Although selection of an appropriate frequency band would seem a prime candidate for further adaptive testing, instead CATSS allows individual learners to both skip ahead to later levels, and to determine for themselves when they have reached their cut-off point and stop the test.

In contrast to adaptive tests, we now consider drills, and their relationships to these tests. Zock and Quint (2004) proposed developing drill tutors from dictionaries based on pattern drills. Since Japanese has freely available dictionary resources, it seems a prime candidate for such drills. This work was later continued (Zock and Afantenos 2007) and developed into a system for drilling basic conversational patterns. These patterns aim to not just aid in vocabulary growth, but to improve automaticity of knowledge, an additional commonly postulated dimension of vocabulary knowledge.

If we compare such drills to formalised tests, several factors emerge. Formalised tests often have much at stake for learners, so the emphasis on the reliability and validity of test outcomes is large. This means that strong tests are developed and evaluated for their discriminating ability amongst candidates. In contrast, drills are typically characterised by flashcard-like simplicity, and are prized for their variety and importantly their coverage of the words under study. Since drill systems are used as vocabulary study methods, their users required this coverage to ensure that their desired vocabulary is acquired.

The amount of manual work required to generate the more linguistically interesting test questions – for example those used the Simple Kanji Aptitude Test (SKAT) described by Toyoda and Hashimoto (2002) – means that full coverage will never be achieved by these traditional methods. Furthermore, the manual labour involved and subsequent limited number of test questions means that these tests are impractical for use as a means of self-study; the number of tests available is small, and after the first exposure, subsequent exposures have greatly weakened validity. However, the flashcard drills which allow timely self-evaluation lack the sophistication of current adaptive tests which use statistical modelling to quickly hone in on learner aptitude.

One of the main contributions of this thesis is in bridging these two methods of testing, and generating linguistically motivated drills which have the potential for use in adaptive testing. Questions for these drills are aimed to be equivalent to questions manually developed for paper tests or computer adaptive tests, but by generating these questions automatically, they also have the coverage and availability required to be used successfully in learner drills.

3.5 Conclusion

In this chapter we discussed a range of issues relating to vocabulary study for Japanese. In particular, we argued that supporting the early reading process is the best way to help learners attain the large vocabulary they need for fluency. Of the many lexical relationships available, we identified near-homography as a promising resource yet to be utilised for aiding learner study. In Chapter 4 to follow, we examine near-homography in depth and attempt to determine the accessibility of a word's orthographic neighbourhood. We also argued that dictionary resources and testing could both be greatly improved with better error modelling. Graphemic relationship models from Chapter 4 then provide a basis for extending the FOKS dictionary to allow partial knowledge search by kanji in Chapter 5. Ultimately, we combine both phonetic and graphemic confusion models into a system for adaptive testing in Chapter 6, before concluding with our overall findings.

Chapter 4

Orthographic similarity

4.1 Introduction

Overview

Chapter 3 has made clear the need for more accessible dictionaries for languages such as Japanese and Chinese. We now concern ourselves with 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. We develop this form of lookup later in Chapter 5.

In order to do so, we require some formal notion of the distance between two characters so as to distinguish near-neighbours, which are plausibly confusable or mentally associated, from higher distance pairs. This chapter then concerns itself with accurate modelling of graphemic similarity.

Distance within alphabetic orthographies

Within alphabetic orthographies such as English, distance metrics are used at the word level or higher, rather than at the character level. This is natural, since alphabetic characters individually do not have associated semantics. At the word-level, the most common distance metric is Levenshtein distance (or *edit distance*). The most natural application is spelling correction, since words with spelling errors are usually visually similar to their correct form.

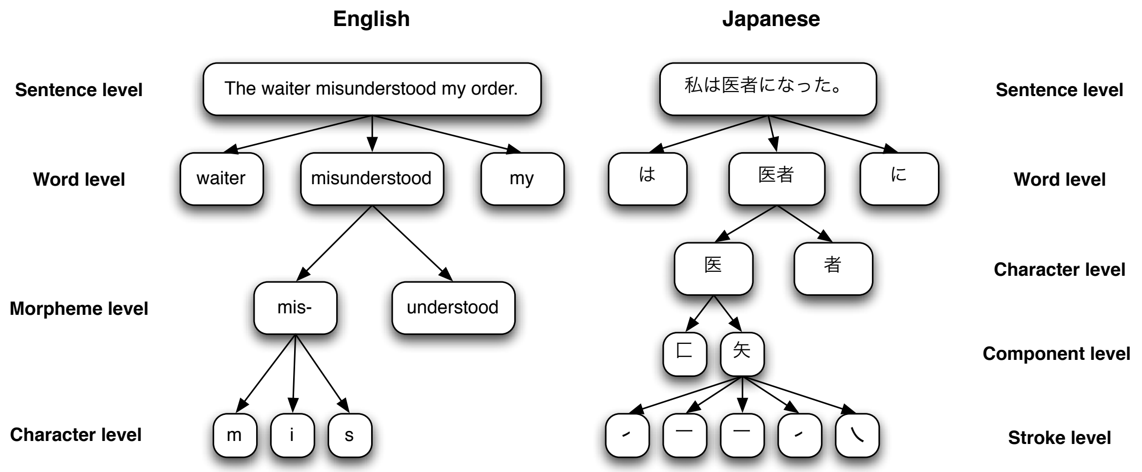


Figure 4.1: Drawing rough equivalence between the linguistic units used in English and Japanese.

Figure 4.1 provides a comparison of the linguistic units in English and Japanese. In Japanese, the individual kanji level is roughly equivalent to the morpheme level for English, with free morphemes as 日 *hi* "day" which are words in their own right, and bound morphemes such as 向 which can only occur as part of a larger word. For example, 向 can occur with conjugational suffixes (向かい *mukai* "facing" or 向かう *mukau* "to face") or as part of larger compounds (向心力 *kōshiNryoku* "centripetal force"). Although the stroke level in Japanese roughly aligns to the character level in English, it lacks both the phonemic contribution to whole-word pronunciation and the linear spatial continuity of characters within an English word.

Between strokes and characters lie common stroke groupings known as *radicals*. Some radicals are themselves characters, such as 日 *hi* "day/sun" and 月 *gatsu* "month/moon" within 明 *aka(rui)* "bright". Others are known short-form variants of full kanji, for example the radical 扌 as a short form for 手 *te* "hand" and 氵 as a short form for 水 *mizu* "water". Finally, some radicals have no whole-character equivalents, though their semantics are well known, such as 艹 "grass". So far all example radicals presented have been atomic. However, combinations of radicals themselves may be considered radicals with stable semantics or phonetics.

The complexity of kanji compared to English morphemes or words thus demands new methods of measuring similarity.

4.2 Distance models: a first approach

Form of models

We can formalise our distance models as follows. Let K be the set of all kanji. A distance model is a function $d : K \times K \rightarrow \mathbb{R}^{+}$, mapping kanji pairs to the value of the distance between them. In practice, each d is composed of two functions $\phi : K \rightarrow F$ and $d' : F \times F \rightarrow \mathbb{R}^{+}$, where ϕ maps each kanji to its representation in some intermediate feature space F, and d' is a distance function on F.

There are three additional desirable constraints on our choice of d. We require, for all $k_a, k_b, k_c \in K$:

$$d(k_a, k_b) = 0 \iff k_a = k_b \qquad \textit{identity of indiscernibles} \tag{4.1}$$

$$d(k_a, k_b) = d(k_b, k_a) \qquad \textit{symmetry} \tag{4.2}$$

$$d(k_a, k_c) \leq d(k_a, k_b) + d(k_b, k_c) \qquad \textit{triangle inequality} \tag{4.3}$$

Equation 4.1 ensures that d can distinguish between different characters, Equation 4.2 meets our intuition that similarity is a symmetric relationship between characters, and Equation 4.3 simply imposes additional regularity on the geometry that d generates on K. If these three requirements are met, then (K, d) is a *metric space*, and d is a *metric* on K.

These requirements in turn constrain our intermediate functions ϕ and d'. To satisfy them, d' should also be a metric on F. Furthermore, if two kanji have an identical feature representation, then d' will not be able to distinguish between them, and thus their composition will not meet Equation 4.1. For this reason, ϕ must be injective, mapping each kanji to a unique representation in F.

Having scoped possible forms of d, we now consider two simple distance models.

Bag of radicals model

We have established that radicals are highly salient features of kanji, as components with their own semantic and phonemic information. Indeed, they are the primary means of looking up an unknown character using the traditional lookup method.[1]

At the radical level, the primary data set available is *radkfile*,[2] which provides radical-membership data for each kanji, and serves as the basis of the WWWJIC multi-radical kanji lookup system.[3] The natural feature space is thus that of sets of radicals, where kanji are decomposed into their constituents, as shown in Figure 4.2.

新 → {八, 立, 十, 辛, 木, 斤}

薪 → {艹, 八, 立, 十, 辛, 木, 斤}

Figure 4.2: Kanji decomposed using *radkfile* into their naive radical-member feature set representations.

If we take the natural choice of ϕ, mapping kanji directly to their radical sets, we immediately find that two kanji can have the same feature representation. Examples of kanji which are mutually indistinguishable in this manner are shown in Table 4.1. This occurs for several reasons.

Firstly, radical membership naturally discards both position and number of radicals, and many small sets of kanji can be found which differ only in these aspects. For example, 木, 林 and 森 only differ in the number of the same primary radical (木) used, and are thus considered identical. Similarly, kanji pairs in which only the position of the radicals differ, for example with 略 and 畧 (radicals: 田, 口, 夂).

[1] The first known Chinese dictionary to index via radicals was the 说文解字 [*shūowénjǐezì*], a Chinese dictionary written in the 1st century A.D. by 许慎 [*xǔshèn*]. It chose one radical from each character, usually the left-most or top-most semantic radical, as the header under which the character could be found.

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

[3] `http://www.csse.monash.edu.au/~jwb/cgi-bin/wwwjdic.cgi?1R`

Identity group	*Radicals*	*Reason*
木，林，森	木	number of 木 radicals
略，畧	田, 口, 夂	position of 田 radical
拐，招	扌, 口, 刀	position of 口 and 刀 radicals
万，丑，乃，乍，垂	ノ, 一, ｜	ambiguity of subcomponents

Table 4.1: Groups of characters which share an identical feature-set representation according to *radkfile*.

Secondly, there is some small amount of noise generated by the ambiguity in the transition from the stroke level to the radical level. For example, 乃 *nai* "whereupon" is itself a radical, but is listed as containing the single-stroke radicals ノ, 一, and ｜. Whilst these may be useful from a dictionary lookup perspective, from a similarity perspective they add noise. The vast majority of characters are either themselves simple non-stroke radicals or are built completely from such radicals, but there are still many characters for which such a description is inaccurate, since they either contain known radicals with extra strokes, or they are composed of stroke groups too rare to be considered radicals.

We get around both problems by adding each kanji to its radical-set, using

$$\phi_{\text{radical}}(k) = \text{radicals}(k) \cup \{k\} \tag{4.4}$$

This ensures a unique representation for each kanji. We can now add a distance metric onto our feature space. We choose a function based on cosine similarity, a commonly used measure in information retrieval, as given in Equation 4.5 below.

$$d_{\text{radical}}(x, y) = 1 - \frac{|\phi_{\text{radical}}(x) \cap \phi_{\text{radical}}(y)|}{|\phi_{\text{radical}}(x)||\phi_{\text{radical}}(y)|} \tag{4.5}$$

Having focused on modelling at the radical level, we now consider the whole-kanji level as an alternative means of modelling similarity.

L_1 norm across rendered images

In Section 3.2, we briefly discussed evidence for stroke level processing in visual character recognition. Indeed, the limited real-life confusability data which is available suggests that stroke-level contributions to whole-character similarity are a source of confusion. For example, a native speaker analysis of FOKS error logs determined that 基 *ki, moto* "basis" and 墓 *bo, haka* "grave/tomb" were confused by a learner during search (Bilac *et al.* 2003). This example shows that learners can mistake very similar looking kanji with few if any shared radicals, provided there are sufficient similar looking strokes in similar positions.

The ultimate representation of how kanji are displayed to users are the actual pixels which will are displayed on-screen. These will vary by font, colour, size, placement and context. We naturally abstract away context, and ignore size and colour by rendering each kanji to a fixed-size black and white image. By choosing rendered images (or rather the intensity for each pixel in that rendered image) as our feature space, we are discarding any additional knowledge we may have about each kanji's internal structure, and instead considering them as arbitrary symbols.

Image distance metrics may be used directly for image similarity search (Zhang and Lu 2003), or embedded within image recognition algorithms such as Radial Basis Function Support Vector Machines, Principal Component Analysis and Bayesian Similarity (Wang *et al.* 2005). For many applications, distance models must typically be scale-, translation- and rotation-invariant (Yang and Wang 2001), transformations which humans easily ignore but which require sophisticated algorithms to counteract. Fortunately, these issues do not occur in our limited feature space: Chinese characters occupy regular sized blocks when typeset, whereas roman characters are variable sized, and must undergo kerning during typesetting to ensure the perceived spacing is even between characters. This difference also exists at a pedagogical level: practice books for children learning to write English provide horizontal lines so that they learn to write with uniform height, whereas Japanese practice books contain a square grid so that children learn to write with uniform size.

Since kanji are rendered aligned with each other, it suffices to use a simple image distance metric. We thus choose the L_1 norm from the family of L_p norms known as *Minkowski distances*. The L_1 norm in particular is also known as the *Manhattan* or *taxi-cab* distance, and is a simple baseline in image similarity literature:

$$L_1(x, y) = \sum_{i,j} |p_x(i,j) - p_y(i,j)| \tag{4.6}$$

The L_1 norm is very simple to calculate, but is known to be sensitive to relatively small image perturbations. For our purposes, its output may be affected by changes to the rendering method, in particular the size of the rendered images and the font used for rendering. We considered an image size of 80×80 pixels to be sufficiently detailed, and used this in all experiments described here. To attempt to attain reasonable font independence, the same calculation was done over 5 commonly available fonts, then averaged. The fonts used were: Kochi Gothic (medium gothic), Kochi Mincho (thin mincho), Mikachan (handwriting), MS Gothic (thick gothic), and MS Mincho (thin mincho). The graphics program Inkscape[4] was used to render them non-interactively.

We expect that the L_1 norm is likely to underestimate the perceptual salience that repeated stroke units (i.e. radicals) have, and thus underestimate radical-level similarity, except where identical radicals are well aligned. Nevertheless, we expect it to correlate well with human responses where stroke-level similarity is present. Pairs scored as highly similar by this method should thus also be rated as highly similar by human judgements.

We now discuss an experiment aimed at collecting data to evaluate these two metrics.

4.3 Similarity experiment

Experiment outline

In order to effectively evaluate our preliminary similarity models, we conducted an exploratory web experiment with the aim of collecting a set of gold-standard orthographic similarity judgements. Participants were first asked to state their first-language background, and level of kanji knowledge, pegged to one of the levels of either the Japanese Kanji Aptitude Test[5] or the Japanese Language Proficiency Test.[6] Participants were then exposed to a number of pairs of kanji, in a manner shown in Figure 4.3, and asked to rate each pair on a five point graded similarity scale. The number of similarity grades chosen represents a trade-off between rater agreement, which is highest with only two grades, and discrimina-

[4] http://www.inkscape.org

[5] 日本漢字能力検定試験: The Japanese government test of kanji proficiency intended for native speakers, which is initially tied to Japanese grade school levels, but culminates at a level well above that expected in high-school graduates.

[6] 日本語能力試験: The Japanese government general-purpose test of Japanese aptitude for second-language learners of Japanese. The test is administered by the Japan Foundation.

tion, which is highest with a large number of grades. Each kanji displayed was rendered as a 50×50 pixel image in MS Gothic font, for consistency across browsers. Whilst this did not guarantee a fixed visual size for each kanji rendered, since participants may have used computer displays of differing spatial pixel density, it was nonetheless expected to reduce the variation in a kanji's visual size across participants.

Although participants included both first and second language readers of Chinese, only Japanese kanji were included in the stimulus. Chinese hanzi and Japanese hiragana and katakana were not used for stimulus, in order to avoid potential confounding effects of character variants and of differing scripts. The pairs were also shuffled for each participant, with the ordering of kanji within a pair also random, in order to reduce any effects caused by participants shifting their judgements part-way through the experiment.

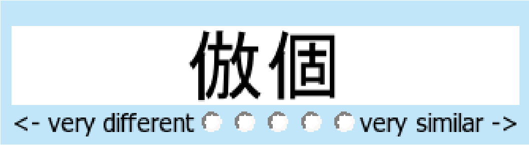


Figure 4.3: Example stimulus pair for the similarity experiment. This pair contains a shared radical on the left.

Each participant was exposed to a common set of 65 control pairs, to be discussed in Section 4.3 below. Further, a remaining 100 random kanji pairs were shown where both kanji were within the user's specified level of kanji knowledge (where possible), and 100 were shown where one or both kanji were outside the user's level of knowledge. This was in order to determine any effects caused by knowing a kanji's meaning, its frequency, its readings, or any other potentially confounding properties.

Web-based experiments are known to provide access to large numbers of participants and a high degree of voluntariness, at the cost of self-selection (Reips 2002). Although participants of all language backgrounds and all levels of kanji knowledge were solicited, the nature of the experiment and the lists advertised to biased participants to be mainly of an English, Chinese or Japanese first-language background.

Control pairs

There are many possible influences on orthographic similarity judgements which we hoped to detect in order to determine whether the data could be taken at face value. A sample pair and a description of each control effect is given in Figure 4.4. Since the number of potential effects considered was quite large, the aim was not statistical significance for the presence or absence of any effect, but rather guidance in similarity modelling should any individual effect seem strong. All frequency and co-occurrence counts were taken from 1990–1999 Nikkei Shimbun corpus data.

Results

The experiment had 236 participants, with a dropout rate of 24%. The participants who did not complete the experiment, and those who gave no positive responses, were filtered from the data set. The remaining 179 participants are spread across 20 different first languages. Mapping the responses from "Very different" as 0 to "Very similar" as 4, the mean response over the whole data set was 1.06, with an average standard deviation for each stimulus across raters of 0.98. The full data set is available in unfiltered form online.[7]

To measure the inter-rater agreement, we consider the mean rank-correlation across all pairs of raters. Although the kappa statistic is often used (Eugenio and Glass 2004), it underestimates agreement over data with graded responses. The mean rank correlation for all participants over the control set was strong at 0.60. However, it is still lower than that for many tasks, suggesting that many raters lack strong intuitions about what makes one kanji similar to another.

Since many of the first language backgrounds had too few raters to do significant analysis on, they were reduced to larger groupings of backgrounds, with the assumption that all alphabetic backgrounds were equivalent. Firstly, we group first-language speakers of Chinese (CFL) and Japanese (JFL). Secondly, we divide the remaining participants from alphabetic backgrounds into second language learners of Japanese (JSL), second language learners of Chinese (CSL), and the remainder (non-CJK). Participants who studied both languages were put into their dominant language based on their comments, or into the JSL

[7] `http://ww2.cs.mu.oz.au/~lljy/datasets/#kanjiexp`

Effect type	**Example**	**Description**
Frequency (independent)	会 店	Frequency of occurrence of each kanji individually. Both kanji in the example pair are high-frequency.
Co-occurrence	法 考	Both kanji co-occur with high frequency with some third kanji. For example, 法 *hō* “Act (law)” occurs in 法案 *hōaN* “bill (law)”, and 考 *kaNga(e)* “thought” occurs in 考案 *kōaN* “plan, idea”.
Homophones	弘 博	Both kanji share a reading. In the example, both 弘 *hiro(i)* “spacious” and 博 *haku* “doctor” share a reading *hiro*. For 博 this is a name reading.
Stroke overlap	策 英	Both kanji share many similar strokes, although no radicals are shared.
Shared graphemes	働 動	Both kanji share one or more graphical elements. These elements might occur in any position.
Shared structure	幣 哲	Both kanji share the same structural break-down into sub-components, although the sub-components differ.
Stroke count	奮 撃	Pairs comparing and contrasting stroke counts. Both examples here have a very high stroke count.
Part of speech/function	方 事	Both kanji have a common syntactic function.
Semantic similarity	千 万	Both kanji are semantically similar. In the example, they are both numbering units.

Figure 4.4: Groups of control pairs used, with an example for each. Parts of readings in brackets indicate *okurigana*, necessary suffixes before the given kanji forms a word.

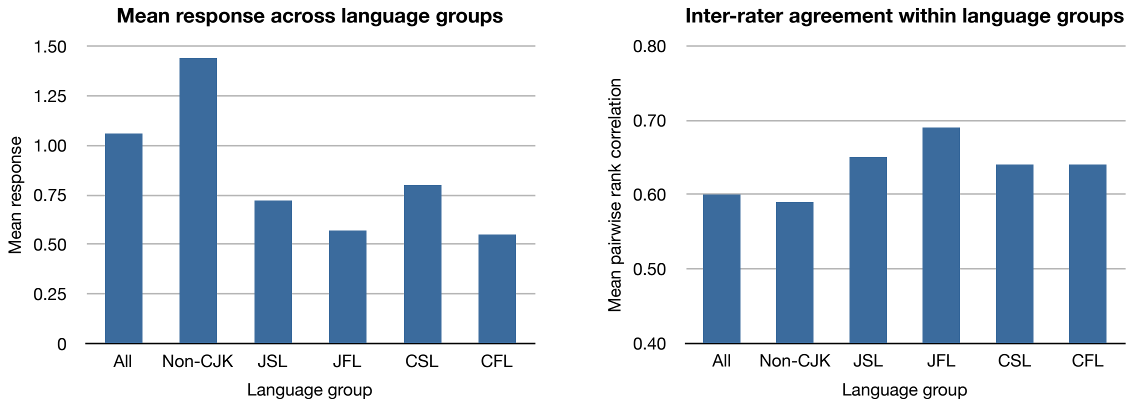


Figure 4.5: Participant responses grouped by language background, measured over the control set stimulus. On the left we give the mean response for each group; on the right, the mean pairwise rank correlation between raters of the same group.

group in borderline cases.[8]

Figure 4.5 shows mean responses and agreement data within these participant groups. This grouping of raters is validated by the marked difference in mean responses across these groups. The *non-CJK* group shows high mean responses, which are then halved for second language learners, and lower still for first language speakers. Agreement is higher for the first-language groups (JFL and CFL) than the second-language groups (JSL and CSL), which in turn have higher agreement than the non-speakers. Both of these results together suggest that with increasing experience, participants were more discerning about what they found to be similar, and more consistent in their judgements.

Evaluating similarity models

Normally, with high levels of agreement, we would distil a gold standard data-set of similarity judgements, and evaluate any model of kanji similarity against our gold-standard judgements. Since agreement for the experiment was not sufficiently high, we instead evaluate a given model against all rater responses in a given rater group, measuring the mean rank-correlation between the model and all individual raters in that group.

[8] Many alternative groupings were considered. Here we restrict ourselves to the most interesting one.

We also have reference points to determine good levels of agreement, by measuring the performance of the *mean rating* and the *median rater response* this way. The mean rating for a stimulus pair is simply the average response across all raters to that pair. The median rater response is the response of the best performing rater within each stimulus set (i.e. the most "agreeable" rater for each ability level), calculated using the above measure.

Model evaluation

The pixel and radical models were evaluated against human judgements in various participant groups, as shown in Figure 4.6, and can be compared to the mean rating and median raters. The pixel based similarity method exhibits weak rank correlation across the board, but higher correlation with increasing kanji knowledge. The radical model however shows strong rank correlation for all groups but the non-CJK, and better improvements in the other groups.

These results match our predictions for the pixel-based approach, in that it performs reasonably, but remains only an approximation. The radical method's results, however, are of a comparable level of agreement within the CFL and JFL groups to the median rater, a very strong result. It suggests that native speakers, when asked to assess the similarity of two characters, make their judgements primarily based either on the radicals which are shared between the two characters. Intuitively, this makes sense. Native speakers have the greatest knowledge of radicals, their meaning, and their semantic and phonetic reliability. They also have the most experience in decomposing kanji into radicals for learning, writing and dictionary lookup.

Group	**Mean**	**Median**	L_1	d_{radical}
Non-CJK	0.69	0.55	0.34	0.47
CSL	0.60	0.65	0.38	0.56
CFL	0.51	0.62	0.44	0.66
JSL	0.64	0.70	0.43	0.59
JFL	0.56	0.69	0.46	0.68
All	0.65	0.62	0.39	0.54

Figure 4.6: Rank correlation of pixel and radical models against raters in given participant groups. Mean and median raters provided as reference scores.

Band	Mean	Median	L_1	d_{radical}
[0, 1)	0.69	0.55	0.34	0.47
[1, 200)	0.62	0.60	0.38	0.53
[200, 600)	0.64	0.69	0.41	0.61
[600, 1000)	0.69	0.72	0.46	0.52
[1000, 2000)	0.56	0.70	0.46	0.65
[2000, ...)	0.58	0.73	0.48	0.70

Figure 4.7: Rank correlation of pixel and radical models against raters in across bands of kanji knowledge. Each band contains raters whose number of known kanji falls within that band's range.

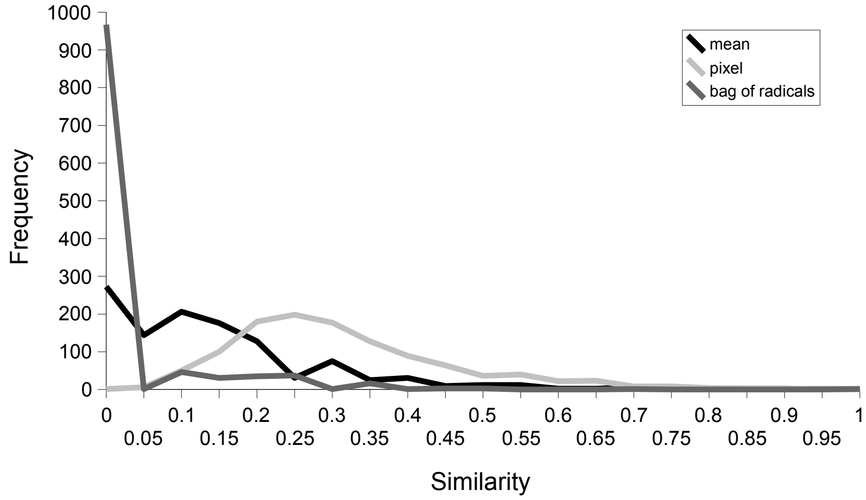


Figure 4.8: Histograms of scaled responses across all experimental stimulus pairs, taken from mean rating, pixel and bag of radical models. Responses were scaled into the range $[0, 1]$.

The radical model still has poor correlation with the non-CJK group, but this is not an issue for applications, since similarity applications primarily target either native speakers or learners, who either already have or will pick up the skill of decomposing characters into radicals. To attempt to determine when such a skill gets picked up, Figure 4.7 shows agreement when raters are instead grouped by the number of kanji they claimed to know, based on their proficiency level. Aside from the $[600, 1000)$ band, there are consistent increases in agreement with the radical method as more kanji are learned, suggesting that the change is gradual, rather than sudden. Indeed, learners may start by focusing on strokes, only to shift towards using radicals more as their knowledge of radicals improves.

The response histograms in Figure 4.8 show stark differences between human responses and the two models. The radical model considers the majority of stimuli to be completely dissimilar. Once it reaches stimulus pairs with at least one shared radical, its responses are highly quantised. The pixel model in comparison always finds some similarities and some differences, and is distributed normally. Human responses for our experiment lie somewhere in between the pixel and radical models, featuring a much smaller number of stimuli which are completely dissimilar, and a shorter tail of high similarity than found with the pixel model. A potentially significant parameter which we do not investigate here is the relationship between the human response histogram and the visual size of kanji stimulus presented. One participant commented that they commonly confuse kanji for one another, but only at small font sizes, not at the size presented in our experiment. It remains open whether the same experiment with kanji rendered to smaller sizes would have elicited increased judgements of high similarity, and equally the extent to which noise (as measured by rater agreement) is affected. Full consideration of this parameter is beyond the scope of our investigation.

4.4 Distance models: a second approach

Our first attempts at modelling similarity gave several insights, relating to both modelling and to evaluation. Our radical metric captured coarse-grained salient features of kanji, but largely ignored the organisation of these features within a kanji. Our pixel model in contrast responded well to structure, but more poorly to salient substructure. In this section we describe an improvement to the radical model and two new metrics which aim to reach a middle ground between these features. Our previous experiment sampled the space of kanji pairings randomly for the majority of human judgements sought. This was done in order to remove potential bias from human selection of pairs. However, what this unintentionally highlighted was the rarity of high similarity pairs: the vast majority of characters are simply reasonably distinctive from one another.

For any given character, there appear to be only a handful or fewer high-similarity neighbours for which it might be reasonably confused. We are thus virtually assured that a randomly chosen pair of kanji will be distinctive from one another, rather than similar. Our random sampling thus had the unintended effect of limiting our measurement to low similarity pairs. For applications, detection and accuracy over high similarity pairs is far more important. For this reason, we introduce two new data-sets for evaluation, and compare metrics over these data sets.

Models

Bag of radicals with shape

We discussed earlier the salience of radicals, and developed our bag-of-radicals model d_{radical}. However, this model ignores the position of radicals, which is known to be important in similarity judgements, and also the number of each radical within a kanji. To address multiplicity of radicals and the findings of Yeh and Li's (2002) study, we set the above metric to unit distance whenever the two characters differ in their basic shape. 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 by *kanjidic* (introduced in Section 3.1), and radical membership by *radkfile* (introduced in Section 4.2).

This change allows the adjusted metric, $d_{\text{radical+shape}}$, to distinguish between examples with repeated components. The final metric aims to capture the visual and semantic salience of radicals in kanji perception, and to also take into account some basic shape similarity.

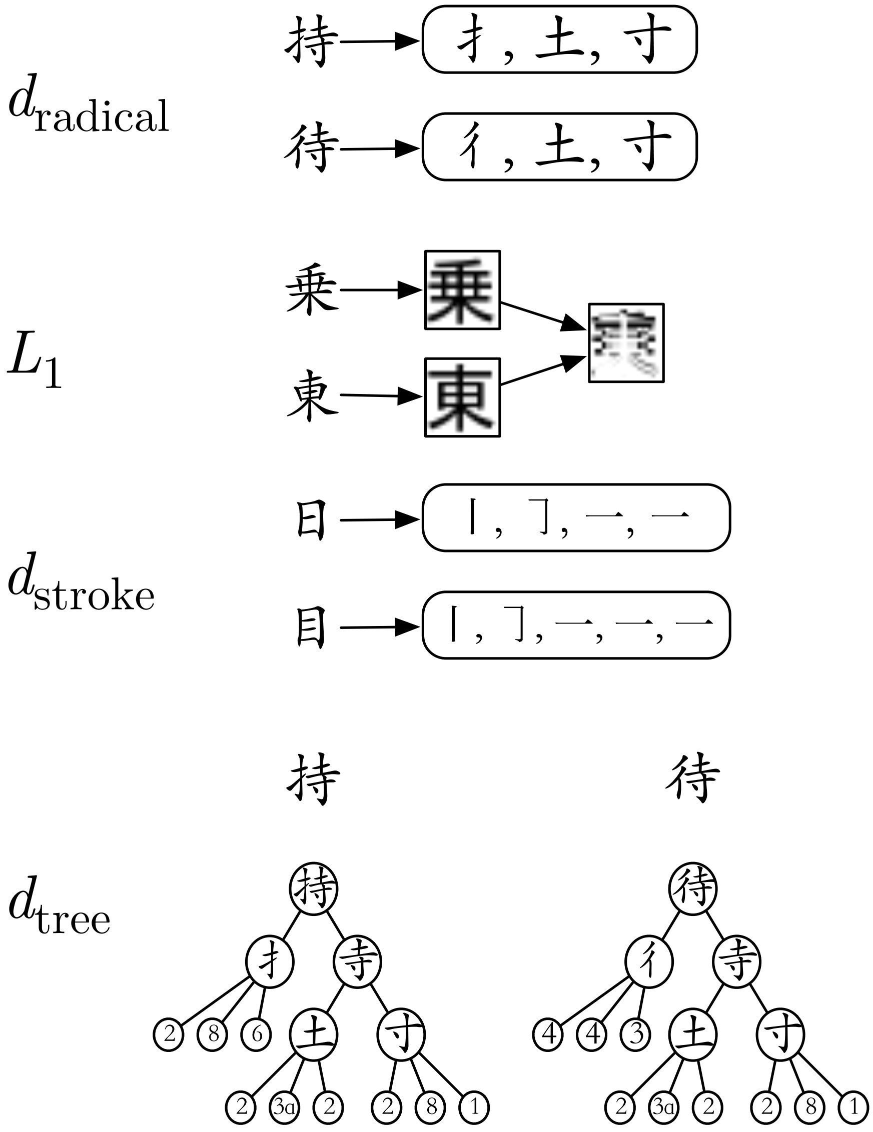


Figure 4.9: A summary of our kanji distance metrics

Stroke edit distance

We saw earlier in Figure 4.1 that just as individual letters occupy the lowest orthographic level for English, individual strokes do the same in Japanese. Spelling correction in English has largely focused on mistakes at the letter level, since computer input for English is mediated by the keyboard, and the keyboard is a per-letter input device. In Japanese, with computer input mediated by IME software, stroke errors in input can not occur unless the mistaken character is also a homophone of the desired character. Without such a source of errors to correct, stroke-based distance metrics have not to our knowledge been examined for Japanese. However, since strokes are the lowest unit from which radicals and kanji are constructed, metrics focusing on the stroke level should be able to capture very fine distinctions between characters, and thus distinguish between highly similar pairs.

A major obstacle to such metrics historically has been the lack of data sets for kanji which describe them at the stroke-level. Fortunately, a hierarchical data set for Japanese kanji was created by Apel and Quint (2004) which provides such a description. 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). Each kanji has a fixed stroke order, a single correct order in which the strokes are hand-written to construct the kanji correctly. The strokes are provided by the dataset in this order.

Ideally, we might wish to transfer successful distance metrics for alphabetic orthographies and examine their performance in Japanese. The most common metric used is edit distance, which is the minimum cost over all edit paths – sequences of *insertions*, *deletions* and *substitutions* which transform one string into another. How might we adequately represent kanji as linear strings, given their two-dimensional layout and the hierarchical data available?

Although kanji are two-dimensional, the *order* in which strokes are written is fixed not only for each character, but also for each component reused across characters. The ordered list of strokes used to write a character thus provides the representation we need to use edit distance, and effectively serves as a signature for each character. In terms of Apel and Quint's data set, this signature is formed from the leaves of each kanji's tree representation, which represent strokes. Taking the edit distance over such signatures gives our d_{stroke} metric.

Figure 4.9 shows example signatures for 日 *hi* "sun" and 目 *me* "eye", where each stroke type is represented by an alphanumeric code. For example, the signature for 日 is the ordered sequence of strokes {丨, ㇆, ㇐, ㇐}, internally represented by the alphanumeric codes $\{3, 11\text{a}, 2\text{a}, 2\text{a}\}$. In its generalised form the edit distance is the minimum cost over all edit paths from one string to another, and the cost of edit operations may be arbitrary and positive. However, we make the simplifying assumption that all strokes types are equally distinctive from one another. This allows us to use a unit cost for all edit operations. We also normalise the edit distance so as to avoid penalising larger kanji due to their large stroke count alone.

Although we discard structural features of each character to create its signature, much useful information remains preserved. Since radicals within each character are written in sequence, they form contiguous blocks within stroke signatures. The edit distance thus recognises shared radicals whenever their position is similar enough. The order of radicals blocks in a signature also reflects their position as part of the larger compound, since 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 recognise the similarity between pairs like 日 *hi* "sun" and 目 *me* "eye".

Tree edit distance

We discarded much of the hierarchical data to generate stroke signatures, in order to use string edit distance as a metric. This carries the implicit assumption that hierarchical information is not as useful as stroke information when measuring similarity. To test this assumption, we also created an additional metric which used the entire tree representation: the ordered tree edit distance between kanji tree representations, d_{tree}.

Tree edit distance is defined as the minimum cost edit path between one tree and another, where an edit path is any sequence of node *insertions*, *deletions* and *relabellings* which transforms one tree into another (Bille 2005). These operations mirrors closely the edit operations used in string edit distance. As with string edit distance, a cost function can be specified giving arbitrary positive symbol-dependent costs to each type of edit operation. We again make a simplifying assumption, that all structural aspects are equally important, allowing us to use unit cost for each edit operation. Figure 4.9 provides an overview of the structure of each kanji's tree representation, though actual trees we generate also contain phonetic elements, radicals, and stroke groups whose strokes are in non-contiguous blocks.

For our kanji trees, generalised tree edit distance completely subsumes stroke edit distance. By this we mean that tree edit distance can perfectly emulate stroke edit distance if an appropriate cost function is chosen, say one which gives zero-cost edit operations for non-stroke nodes. This is possible regardless of the cost function used for stroke edit distance. From this perspective, a comparison between these two methods is only a comparison between cost functions for tree edit distance.

However, both the algorithmic complexity and the implementation of string edit distance are far simpler than that of tree edit distance. For two kanji i and j, let s_i be the number of strokes in kanji i, and n_i is the number of nodes in its tree representation. Then stroke edit distance takes $O(s_i s_j)$ time. In comparison, using Demaine *et al.*'s (2007) optimal decomposition algorithm – which the authors generously provided sample code for – our tree edit distance implementation had worst-case time complexity $O(n_i n_j^2(1+\log(\frac{n_i}{n_j})))$. This is a significant improvement over earlier algorithms, such as those described by Bille (2005), but in practice our implementation remained two orders of magnitude slower than that for string edit distance.

Evaluation

We evaluated all four kanji distance metrics over three data sets. Firstly, we re-evaluated over the earlier similarity experiment data (from Section 4.3) in order to compare the newly added metrics against those already discussed. However, this data-set had the acknowledged weakness of not including enough high-similarity pairs. We thus sought to rectify this by finding a source of expert similarity judgements, finding these judgements in the form of the White Rabbit JLPT Level 3 kanji flashcard set. 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. Finally, we elicited further native speaker judgements in a similarity pool experiment, and again compared each of the metrics on this new data set. We discuss each of these evaluations below.

Similarity experiment data

Calculating the rank correlation ρ averaged over raters in each group, as we did earlier, gives the results shown in Figure 4.10.

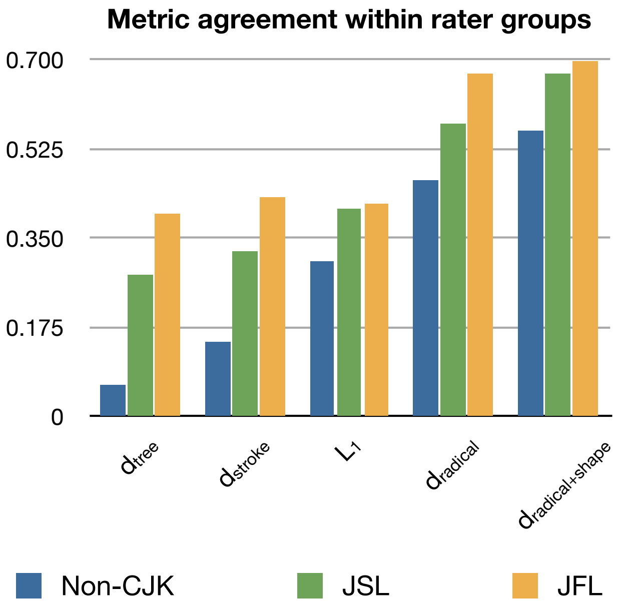


Figure 4.10: Mean value of Spearman's rank correlation ρ, calculated over each rater group for each metric.

The results show the same broad patterns as before in terms of language ability: the mean rank correlation increased as the participants' knowledge of Japanese increased. However, the $d_{\text{radical+shape}}$ metric dominates over the other metrics, including the original d_{radical}, 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, despite large differences between them for non-speakers, all perform equivalently for native-speakers.

Despite overall poor performance from our new metrics, we have been able to improve on d_{radical} by adding shape information. We now describe the flashcard data set, and evaluate over it for comparison.

Flashcard data set

Having identified problems with our earlier experiment due to the lack of high-similarity stimulus pairs, we looked for a source of such pairs. A series of kanji flashcards developed by White Rabbit Press provided this source. Each card is designed to allow the study of a single kanji, for example 晩 in Figure 4.11. For each kanji one or two visual neighbours are provided whose visual similarity to the kanji being studied makes them potentially confusable. We used their JLPT 3 flashcards, providing such visual neighbours for 245 kanji in total. We have provided these similarity pairs online for further investigation and use.[9]

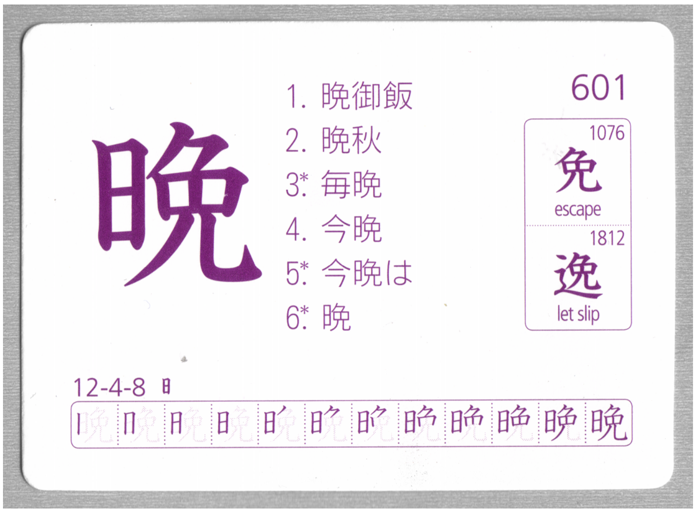


Figure 4.11: An example White Rabbit Press flashcard for the kanji 晩 *baN* “evening”. Note the two large characters on the right provided due to visual similarity with the main kanji.

We took two different approaches to evaluation. Firstly, for each high-similarity pair (a *pivot* kanji and its distractor), we randomly selected a third kanji from the jōyō character set[10] which we combined with the pivot to form a second pair. By virtue of the large number of potential kanji pairings, most of which bear no real similarity, this second pair is

[9] `http://ww2.cs.mu.oz.au/~lljy/datasets/#whiterabbit`

[10] The “common use” government kanji set, containing 1945 characters.

Metric	Accuracy
L_1	0.954
d_{tree}	0.952
d_{stroke}	0.926
d_{radical}	0.736
$d_{\text{radical+shape}}$	0.603
random baseline	0.500

Table 4.2: Accuracy at detecting which of two pairs (flashcard vs. random) has high similarity

highly likely to be of lower similarity than the first pair. We then compared 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 4.2. We include a theoretical random baseline of 0.500, since any decision has a 50% chance of being successful.

One immediate and surprising result is that the $d_{\text{radical+shape}}$ and d_{radical} metrics which dominated the previous task perform far poorer than the other metrics. This suggests that they are poor at distinguishing between high and medium similarity pairs, though our earlier evaluation suggests they broadly order examples correctly across the whole spectrum, as shown by its performance on the similarity experiment data. Their 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 three other metrics have accuracy above 0.9 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. For applications, it is most important that the few true high-similarity neighbours are accurately reported by the chosen metric, and are not swamped by erroneous candidates.

To better evaluate the retrieval performance of each metric, we considered the main flashcard kanji to be a query and its neighbours ranked in order of proximity as the retrieved documents for this query. We used the high-similarity flashcard neighbours as the full set of relevant documents, implicitly considering all other kanji as irrelevant documents.

Metric	MAP	p@1	p@5	p@10
d_{tree}	0.320	0.313	0.149	0.094
d_{stroke}	0.318	0.310	0.151	0.099
L_1	0.271	0.257	0.139	0.089
$d_{\text{radical+shape}}$	0.211	0.197	0.087	0.063
d_{radical}	0.177	0.144	0.085	0.065

Table 4.3: The mean average precision (MAP), and mean precision at $N \in \{1, 5, 10\}$ over the flashcard data

	d_{tree}	d_{stroke}	L_1	d_{radical}	$d_{\text{radical+shape}}$
d_{tree}	-	0.934	0.026	<0.001	<0.001
d_{stroke}	0.934	-	0.027	<0.001	<0.001
L_1	0.026	0.027	-	<0.001	<0.001
d_{radical}	<0.001	<0.001	<0.001	-	0.254
$d_{\text{radical+shape}}$	<0.001	<0.001	<0.001	0.254	-

Table 4.4: Pairwise significance figures for MAP scores over the flashcard data, shown as p values from a two-tailed Student's t-test

For each query we calculated the average precision (AP) for the given metric, according to Equation 4.7, where $P(r)$ is the precision at a rank cutoff r and rel(r) is a binary function set to 1 when the retrieved result at rank r is a relevant document.

$$\text{AP} = \frac{1}{n_{\text{relevant}}} \sum_{r=1}^{n_{\text{retrieved}}} P(r) \times \text{rel}(r) \tag{4.7}$$

Using all queries gives a vector of AP values for each metric. Taking the mean of each vector yields the mean average precision (MAP) statistic for each metric. MAP is widely used in Information Retrieval to evaluate document ranking methods. This makes it suitable for use with similarity metrics, since for many applications the most important feature of a metric is the ranking of neighbours which it provides. A strength of MAP is thus that it evaluates metrics in a manner that corresponds closely to their intended use. A disadvantage is that it assumes that all relevant documents are equally relevant, whereas in practice we may find important differences in proximity even between high-similarity neighbours. In practice this is not a limitation, since the high-similarity neighbours provided by the flashcard set are provided to us unranked.

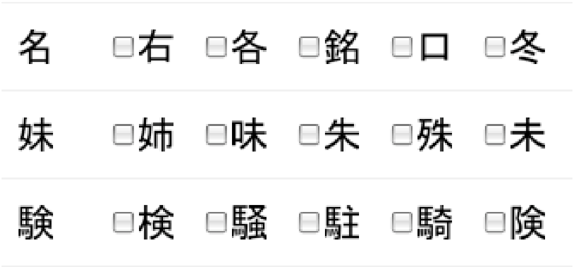
Table 4.3 shows the MAP score for each metric, along with the precision at N neighbours, for varied N. The precision statistics confirm the rough ranking of metrics found in the earlier classification task, with the radical-based metrics performing worst. In this more difficult task, the L_1 norm is outperformed by d_{stroke} and d_{tree}. Pairwise significance figures as given in Table 4.4 were calculated by performing a two-tailed Student's t-test on the average precision vectors for each pairing of metrics. Most differences in MAP scores are significant at the 95% confidence level, with a few key exceptions that are telling of the relationship between each of the metrics. d_{stroke} and d_{tree} give a p value of 0.934, indicating their difference is nearly identical for this task. The two radical-based methods give a p value of 0.254, which gives low confidence that their performance genuinely differs. Both of these results are unsurprising, given the similarity between the two forms of edit distance and the two radical-based methods.

Distractor pool experiment

The flashcard data, though providing good examples of high-similarity pairs, suffers from several problems. Firstly, the constraints of the flashcard format limit the number of high-similarity neighbours which are presented on each flashcard to at most two; in some cases we might expect more. Secondly, the methodology for selecting high-similarity neighbours appears subjective.

For these reasons, we conducted an experiment to attempt to replicate the flashcard data. 100 kanji were randomly chosen from JLPT 3 set (hereafter *pivots*). For each pivot kanji, we generated a pool of possible high-similarity neighbours in the following way. Firstly, the pool was seeded 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 performing a dry run, each participant was presented with a series of pivots and their pooled neighbours, as shown in Figure 4.12. Their task was to select the neighbours (if any) which might be confused for the pivot kanji, based on their graphical similarity. The order of stimuli was randomised for each rater, as was the order of neighbours for each pivot. Kanji were provided as 30×30 pixel images using MS Gothic font for consistency across browsers, and with our earlier similarity experiment (Section 4.3).



名	□右	□各	□銘	□口	□冬
妹	□姉	□味	□朱	□殊	□未
験	□検	□騒	□駐	□騎	□険

Figure 4.12: Example stimulus from the distractor pool experiment. For each kanji on the left-hand side, participants could mark one or more of the kanji on the right as potentially confusable visual neighbours.

3 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. The full dataset is available for scrutiny and further use online.[11]

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 is small, we simply pooled 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 4.5.

Compared with the flashcard data set, the ordering and relative performance of metrics is similar, with d_{stroke} marginally improving on d_{tree}, but both 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.

[11] `http://ww2.cs.mu.oz.au/~lljy/datasets/#poolexp`

Metric	MAP	p@1	p@5	p@10
d_{stroke}	0.406	0.530	0.240	0.146
d_{tree}	0.383	0.560	0.234	0.142
L_1	0.349	0.530	0.210	0.123
d_{radical}	0.288	0.350	0.168	0.122
$d_{\text{radical+shape}}$	0.211	0.270	0.122	0.096

Table 4.5: The mean average precision (MAP), and mean precision at $N \in \{1, 5, 10\}$ over the pooled distractor data

	d_{stroke}	d_{tree}	L_1	d_{radical}	$d_{\text{radical+shape}}$
d_{stroke}	-	0.467	0.103	0.002	<0.001
d_{tree}	0.467	-	0.368	0.023	<0.001
L_1	0.103	0.368	-	0.115	<0.001
d_{radical}	0.002	0.023	0.115	-	0.004
$d_{\text{radical+shape}}$	<0.001	<0.001	<0.001	0.004	-

Table 4.6: Pairwise significance figures for MAP scores over the pooled distractor data, shown as p values from a two-tailed Student's t-test

Upon checking pairwise significance tests for MAP calculations over this new data set (Table 4.6), some differences become clear. The performance improvement between d_{radical} and $d_{\text{radical+shape}}$ is significant at the 95% confidence level, rather than the weaker confidence shown over the same measurement for the flashcard data. The differences between d_{stroke}, d_{tree} and L_1 are also reduced in significance. Together, these changes could indicate a difference between the flashcard and pooled distractor data, perhaps reflecting the different biases provided by the experimental participants in comparison to the creators of the flashcard set.

4.5 Discussion

Modelling of graphemic similarity was initially hampered by a lack of data. In this chapter we have identified the White Rabbit Press flashcards as a source of expert judgements, and developed data sets for evaluating similarity metrics through two different experiments. This area is no longer data poor, and we anticipate the development of better metrics still based on this data.

Stroke edit distance and tree edit distance performed best on nearly all our evaluation methods on high-similarity pairs, and indeed were roughly comparable for each task. This suggests that the stroke signatures alone capture much of the structural information important to whole-character similarity, or alternatively that both metrics were able to provide an appropriate level of fuzzy matching between similar components. Fortunately, both metrics also have substantial scope for improvement, merely by providing more principled weights for edit operations; in particular, the tree edit distance has the ability to increase or decrease the estimated salience of different structural features through such weights, and perhaps better fit the reality of human perception.

Although we have focused on Japanese, these models are all equally applicable to Chinese, subject to finding or constructing the requisite data sources, and may suggest approaches for other more general symbol systems which aren't adequately indexed by the existing body of research on image similarity search. Alphabetic scripts lack a tree-based structure, though they may still use stroke sequences, and can easily be rendered as images, indicating that at some of these metrics are transferrable more generally to other languages with simpler scripts.

4.6 Conclusion

Visual similarity is assumed to be the basis for many kanji misrecognition errors, and so accurate kanji similarity models have useful application in error correction and dictionary search. In this chapter we firstly attempted to model similarity by using bag-of-radicals and image difference approaches. In order to evaluate these metrics, we performed a large experiment which directly asked participants to rate the graphical similarity of random pairs of kanji. Our initial evaluation on this data showed best performance for the bag-of-radicals metric, particularly when additional layout information was added, whereas the image distance metric performed poorly. We initially interpreted this to indicate the salience of radicals as sub-components. It remained clear that our metrics were still too noisy for downstream applications, and that the randomly chosen stimulus pairs followed too closely the natural distribution of high-similarity pairs – that is, there were very few in the entire data set. Since most applications rely heavily on high-similarity pairs, the data set was found to be inappropriate for evaluating metrics for their intended purpose.

We addressed this issue by using human expert-selected similarity pairs, in the form of potentially confusable pairs borrowed from the White Rabbit Press JLPT 3 & 4 flashcards. We simultaneously considered two new metrics, both based on Ulrich Apel's tree data for kanji: the stroke edit distance, and the tree edit distance. Both performed comparably poorly to the L_1 norm on the original data set, which required only that metrics correctly order medium-to-low similarity pairs. However, on the new flashcard data set, we found a reversal of these results. The radical level which had proved so appropriate for ordering low-to-medium similarity pairs was simply too coarse to differentiate high-similarity pairs, and was outperformed by the other metrics. In particular, stroke and tree edit distances performed comparably best, followed closely by the L_1 norm.

To determine how reproducible the flashcard data set was, we performed a small experiment which took a number of stimulus kanji and for each kanji pooled the top-ranked neighbours from each of our similarity metrics into a set of potentially confusable neighbours. We asked native or native-like speakers of Japanese to assess each neighbour pool and mark those neighbours which they considered truly confusable by learners. Since we had few participants, we in turn pooled their affirmative responses for each kanji presented. The pooled responses agreed moderately with the flashcard data, yielding a κ of 0.44, a result better than any individual rater paired with the flashcard data. This suggested that, although individuals may differ in their immediate conscious judgements, aggregated responses across groups of raters hold the promise of matching more closely the human experience of similarity and visual confusion.

We now wish to apply these metrics to error modelling so that we may aid learners in common tasks they perform. Since one of the most used tools in language learning is the dictionary, the following chapter describes our attempts to enhance a dictionary with improved usability and accessibility features. A key feature of this improved dictionary is a search by similarity, which makes use of metrics from this chapter and incorporates them into the error models required for lookup. Chapter 6 then takes these confusability models one step further by incorporating them into automatic learner drills. The similarity metrics from this chapter provide the basis for these new models of learner error and the new applications that result.

Chapter 5

Extending the dictionary

In Chapter 3 we identified vocabulary acquisition as a crucial hurdle for language learners, and established the important role of dictionaries in supporting vocabulary learning, particularly in Japanese. We further argued that graphemic relationships modelling was underdeveloped in Japanese; these models were then developed in Chapter 4. This chapter examines an individual approach to dictionary lookup, FOKS, and investigates how improvements to the method – including use of these new graphemic neighbour models – might better support both lookup and retention.

We begin in Section 5.1 with an extended background to the FOKS system, including its original architecture and error models. Our work required re-implementation of FOKS; significant architectural changes from the original system are thus discussed. The remaining three sections then describe improvements to the system.

Firstly, Section 5.2 examines the grapheme-phoneme alignment step in FOKS's construction, crucial to providing accurate reading and alternation frequencies for its core error modelling. We improved its accuracy and efficiency by adapting its unsupervised alignment method into a semi-supervised method. Secondly, Section 5.3 investigates micro-structural aspects of dictionary translations, with the goal of improving lookup and ultimately retention of the word once found. Thirdly, Section 5.4 combines FOKS's phonemic error-correction with a novel graphemic error-correcting search, making use of the graphemic neighbourhood modelling from Chapter 4. Indeed, the lack of such correction in FOKS and other dictionaries was a significant motivation for developing those models.

Finally, we look to log analysis in Section 5.5 to evaluate the effect of these changes where possible.

5.1 FOKS: the baseline

Intelligent lookup

Our focus on FOKS requires a recap of its contribution to dictionary search in Japanese. We can summarise the problems learners of Japanese (and native speakers) face when encountering a new word in the following way: words must be input into a computer by pronunciation, but the pronunciation is unknown when the word is unknown. For this reason, users normally have to fall back to a slower and more imprecise lookup method based on visual analysis. This is the basic situation, but this description is incomplete.

There are many situations when the learner has partial knowledge of the word they are looking up. For example, the context in which the word occurs may give clues as to the word's meaning – in fact the entire inferring-from-context strategy (Section 3.1) is based on fleshing out this partial knowledge over many exposures to a word. Equally commonly, a kanji compound may be encountered where the learner knows something about each of the kanji involved in the compound. Just as they might guess the meaning by combining the meanings of both elements, they can guess the pronunciation in the same way. In Japanese, such guesses are likely to be wrong for one of three reasons:

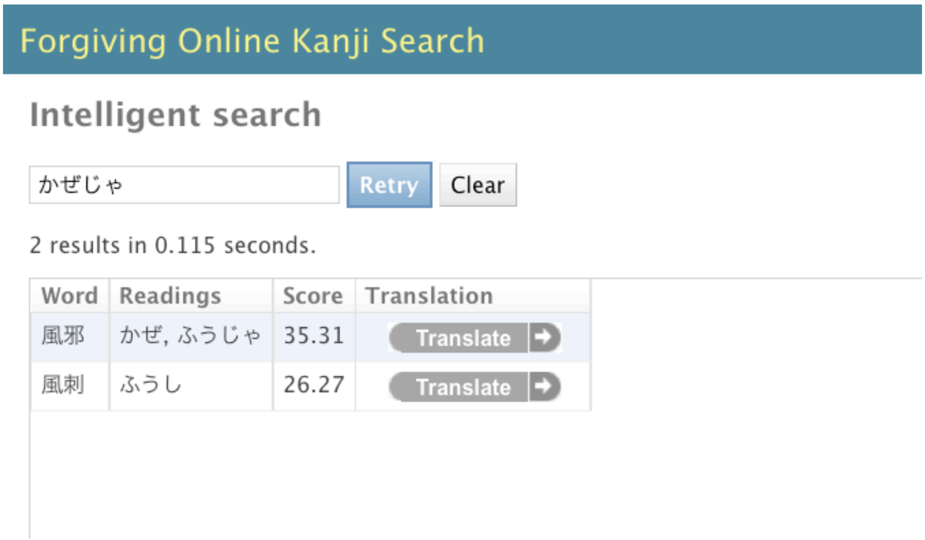
1. Each kanji may have several readings, yet a particular compound typically only has a single valid reading. Choosing the correct reading for each kanji is difficult.

2. When readings are combined into a whole, one or more combination effects can occur. For example, 日 *nichi* "sun" + 本 *hoN* "origin" combine to form 日本 *nippoN* "Japan". The most prominent effects are *sequential voicing* and *sound euphony*, which we discussed in more detail earlier in Section 2.3.

3. Some compounds have non-compositional readings, i.e. their pronunciation is not based on any of their components, but is rather special to that compound. For example, 山車 *dashi* "festival float" is non-compositional in this way.

When a learner guesses incorrectly and searches by this pronunciation in a typical dictionary, they will not find the desired word, and will have to resort to slower and more complex searching. However, using the FOKS dictionary, they can search using the expected pronunciation. Even if they choose the wrong reading, or mistakenly apply combination effects, FOKS will seamlessly correct the error and find the word they were looking for quickly and accurately from within the same interface. They may equally use the interface to intentionally search for words by an implausible but compositional reading; it will correct for this as if it were simply an error.

For example, suppose the user wishes to look up 風邪 *kaze* "common cold". He or she may know the kanji 風 *kaze/fū* "wind", and also 邪 *yokoshima/ja* "evil, wicked", and thus guess that the reading for 風邪 is *kazeja*, one possible combination of readings. However, the correct reading *kaze* is non-compositional. Figure 5.1 depicts the results of this search in the FOKS interface. Despite the incorrect guess, FOKS still lists the target word with the correct reading in its list of candidates for the guessed reading.



Forgiving Online Kanji Search

Intelligent search

かぜじゃ

2 results in 0.115 seconds.

Word	Readings	Score	Translation
風邪	かぜ, ふうじゃ	35.31	Translate
風刺	ふうし	26.27	Translate

Figure 5.1: FOKS search results for the reading *kazeja*. The first result 風邪 *kaze* "common cold" is the desired word.

Architecture

FOKS is best termed a dictionary *interface*, rather than a dictionary *resource*; it serves as an advanced indexing scheme constructed on top of an existing dictionary. This underlying dictionary resource could be arbitrary, but in practice is a combination of one or more of the freely available EDICT family of Japanese-English word-level dictionaries maintained by the Electronic Dictionary Research and Development Group.[1]

This section provides an overview of the architecture of FOKS, as described by Bilac (2005), and thus shows how FOKS is able to recover from incorrect reading guesses. The construction of FOKS has three main stages, as shown in Figure 5.2: grapheme-phoneme (GP) alignment, canonisation and reading generation. The simplest way to explain is to step through FOKS's construction.

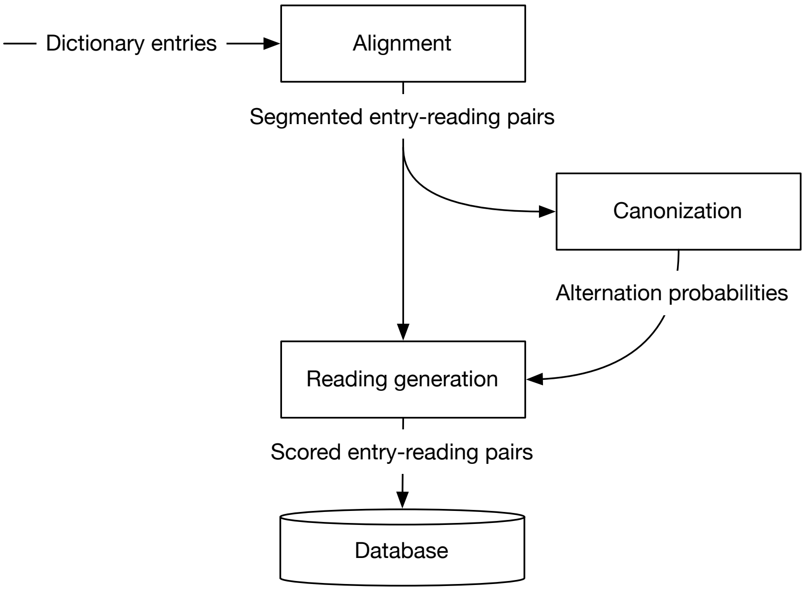

Dictionary entries
Alignment
Segmented entry-reading pairs
Canonization
Alternation probabilities
Reading generation
Scored entry-reading pairs
Database

Figure 5.2: The architecture of the original FOKS interface.

For each entry in the dictionary which contains kanji, FOKS constructs an exhaustive list of plausible guesses for the word's reading, scored by an estimate of their likelihood (or plausibility). In order to generate these readings, it needs to split words like 神社 *jiNja* "shrine" into their component segments by aligning word-reading pairs, a pro-

[1] http://www.edrdg.org/

cess known as grapheme-phoneme alignment, using an unsupervised method described by Baldwin and Tanaka (1999a). The result of this step is a series of GP-aligned words. Once GP-aligned, our example becomes 神|社 ↔ *jiN|ja*.

These alignments are used in two ways. Firstly, they are fed into a canonisation process which recognises any reading alternations which have occurred. For example, 社's reading *ja* is recognised to be *sha* altered by sequential voicing. If r is the segment reading, r_c is its canonical reading, and s is its kanji form, then GP-alignment alone provides a frequency distribution for $\Pr(r|s)$, whereas canonisation converts this into further distributions for both $\Pr(r|r_c, s)$ and $\Pr(r_c|s)$.

Secondly, the alignment for each word is used as the basis for reading generation. Error models use the frequency distributions from canonisation to generate a series of plausible readings for each grapheme segment, usually in the form of an alternation from an existing reading. These error models will be discussed further in the following section. Usually the correct reading occurs naturally as one of the plausible readings suggested; in rare cases where this does not occur, the correct reading is inserted into the generated list. Each word's readings are then weighted by the word's corpus frequency $\Pr(w)$, pruned using a plausibility threshold, and then stored in the database. From here, lookup is a simple matter of querying against the plausible readings in the database – this is facilitated by a straightforward web interface.

The overall lookup model for a word w given a reading query r_w is as follows:

$$\begin{aligned}
\Pr(w|r_w) &\propto \Pr(r_w|w)\Pr(w) \\
&= \Pr(w)\Pr(r_{1\ldots n}|s_{1\ldots n}) \\
&\approx \Pr(w)\prod_i^n \Pr(r_i|s_i) \qquad (5.1)
\end{aligned}$$

Firstly, Bayes rule shows the results can be ranked by $\Pr(r_w|w)\Pr(w)$; in this term, $\Pr(w)$ is simply modelled by corpus frequency of words. We then break the reading and words up into GP segments, and approximate with independence of GP-segments, giving us the final equation given in Equation 5.1. This equation relates plausible (mis)readings for kanji with plausible (mis)readings for words.

Error models

We suggested three main errors made by learners in our recap of lookup using FOKS: inadequate choice of kanji reading, incorrect application of alternation rules for sequential voicing and sound euphony, and non-compositional readings. A more thorough list of error types encountered through extensive log analysis is discussed in Bilac (2005:25). Several of these extra error types are corrected by the original FOKS interface, including:

1. *Palatalisation errors.* For example, misreading 亜流 *aryū* "epigone" as *arū*.
2. *Character/suffix co-occurrence errors.* For example, misreading 激しい *hageshī* "violent" as *kibishī* due to common suffix with 厳しい *kibishī* "severe".
3. *Insufficient knowledge of proper nouns.* Many proper nouns have difficult or archaic readings. For example, the word 上野 has at least 13 distinct readings as proper nouns, including *agano*, *ueno*, and *uwano* for place names, as well as *uehara*, *kamitzuke*, *toshi* for person names.

The last of these errors, insufficient knowledge of proper nouns, is better described as the problem of *non-compositional readings*.

Limitations

Language learners often have useful information they can use to constrain a query, but are unable to express it in traditional electronic dictionaries. In particular, most only allow partial but correct queries through wildcards. FOKS takes a unique approach amongst Japanese dictionaries in allowing the user to express their partial knowledge – in this case about kanji pronunciation – in terms of a noisy query, from which it then attempts to recover the original word. Despite success with this approach, the original system contained several limitations.

A number of other errors were not corrected within FOKS, such as those due to:

1. *Phonetic confusion of words or characters.* Confusing a kanji with a homophone, and then applying a different reading borrowed from the homophone.

2. *Graphemic similarity of words or characters.* Confusing a kanji with a near-homograph, and then using a reading of the neighbour. For example, 闇 *yami* “darkness” being misread as *oN* due to visual similarity with 音 *oN* “sound”.

3. *Semantic similarity of words or characters.* Confusing a word with a near-synonym, and then using a reading of the synonym. For example, 火事 *kaji* “fire” being misread as *kasai* due to similarity with 火災 *kasai* “(disastrous) fire”.

Phonetic, graphemic and semantic links or “associations” are all increasingly the focus of new dictionary interfaces, as discussed in Section 3.3. It is not surprising that they cause errors in general, but it is at least unexpected that all three forms of proximity affect search by pronunciation in measurable ways. The magnitudes of these effects are given figures in Bilac *et al.*’s (2004) log analysis: 0.8% of errors were due to graphemic similarity, whereas 0.3% of errors were classified as either grapho-phonetic similarity, semantic similarity or suffix co-occurrence errors. These figures are simply lower bounds though, since only errors corrected by existing error models were logged and analysed. More generally, such errors would not have resulted in successful search, and thus would not be included in the log analysis.

These cases indicate that users sometimes have partial information about the word they wish to find, which they express through their search (intentionally or otherwise). In the well known tip-of-the-tongue phenomenon (Brown and McNeill 1966), the rememberer can recall parts of the words pronunciation but not the entire word. Partial information searches are aimed at allowing the learner to still find their word, even in such cases. One significant limitation to FOKS is that it only allows partial knowledge to be expressed in one manner – through pronunciation. Information about form or meaning is ignored. For example, if any kanji are found in the query, FOKS limits the search to exact matches.

When FOKS does correct for misreadings, there is often lack of transparency in the correction method. For example, log analysis determined that 塵芥 *chiriakuta* “garbage” was reached by the query *chiNke*, but the type of error made by the user was unclear even to an expert (Bilac 2005:27).

Finally, an important limitation for faster experimentation is the build time for FOKS; the grapheme-phoneme alignment cycle in particular is unsupervised, and takes a long time to complete, as does the offline reading generation. If the longest word length is L, and there are W words, then the time complexity for GP alignment is $O(W^2 2^{2L})$. That is, it scales quadratically with dictionary size, and exponentially with longest word size. This effect is increased as dictionaries increase their coverage and thus their size.

FOKS rebuilt

With many of these limitations in mind, we rebuilt FOKS with several significant changes. Firstly, many offline steps such as reading generation were moved to become online steps occurring at query-time. This had the benefit of faster database rebuilds for easier experimentation, but also that plausible misreadings generated by the system could be easily reverse engineered to explain how they were reached.

Beyond these basic architectural changes, several other improvements were made to FOKS, which form the focus of this chapter. In Section 5.2 we examine the grapheme-phoneme alignment algorithm and develop variants suitable for faster experimentation. Section 5.3 then looks at so-called "microstructural" improvements to word translation to improve usability, and extends coverage over place names. Section 5.4 then uses our newly developed graphemic similarity models to provide error-correcting search by grapheme. We conclude with log analysis of these changes in Section 5.5.

5.2 Grapheme-phoneme alignment

Overview

This section considers improvements to FOKS's grapheme-phoneme alignment algorithm, however these improvements have potential for broader application beyond FOKS. In our description and analysis of this problem, we situate them in this wider context.

The grapheme-phoneme ("GP") alignment task aims to *maximally* segment the orthographic form of an utterance into morpho-phonemic units, and align these units to a phonetic transcription of the utterance. Maximal indicates the desire to segment grapheme strings into the smallest meaningful units possible. Taking the English example word *battleship* and its phonetic transcription /bætlʃɪp/, one possible alignment is:

b	a	tt	le	sh	i	p
b	æ	t	l	ʃ	ɪ	p

Note that alignment in general is many-to-many. In the example above, *tt* aligns to /t/, *le* aligns to /l/ and *sh* aligns to /ʃ/. Equally it might be possible for some letters to align to an empty string. This task is challenging for any language without a one-to-one correspondence between individual graphemes and phonemes, as is the case with English (Zhang *et al.* 1999), Japanese (considering graphemes as kanji characters), and indeed most languages with a pre-existing writing system.

Aside from FOKS, GP alignment is a prerequisite for many applications. For example, the alignment process, and its resulting aligned GP tuples, are a precursor to achieving automated grapheme-to-phoneme mappings for text-to-speech systems such as MITALK (Allen *et al.* 1987), Festival (Black *et al.* 1999) and SONIC (Pellom and Hacioglu 2001). Further uses include accented lexicon compression (Pagel *et al.* 1998), identification of cognates (Kondrak 2003) and Japanese-English back-transliteration (Knight and Graehl 1998; Bilac and Tanaka 2005).

There are several successful approaches to Japanese GP alignment, notably the iterative rule-based approach taken by Bilac *et al.* (1999), later followed by Baldwin and Tanaka's (1999a) unsupervised statistical model based on TF-IDF. Although these models have high accuracy, their iterative approach has a high computational cost, making them impractical for many real-world applications. For the statistical models, this is partially a consequence of their strongly unsupervised nature. We thus explore the use of the EDICT and KANJIDIC electronic dictionaries (Breen 1995) as means of constraining the alignment search space and reducing computational complexity.

This section examines in detail Baldwin and Tanaka's (1999a) GP alignment method, and alters it to achieve comparable alignment accuracy at a much lower computational cost. To achieve this goal, we split the task of GP alignment into a pure alignment subtask and an okurigana detection subtask, and compare algorithm variants of pre-existing approaches for both.

In Japanese, the GP-alignment problem is simplified somewhat by the convenience of using the syllabic kana script as the phonemic representation, though we will continue to use romanised forms in our examples. A simple example is given in Figure 5.3, where several possible alignments are shown. Kana are convenient in this context because they can occur in both the grapheme and phoneme string, as in the *suru* suffix shown. Whereas kanji in the grapheme string serve as wildcards, kana serve as inflexible pronunciations which constrain the number of possible alignments.

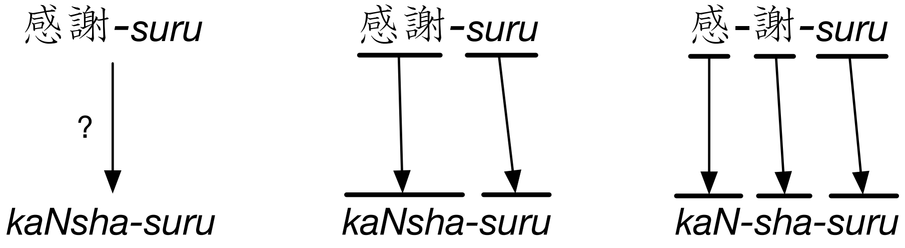


Figure 5.3: The dictionary entry for 感謝する *kaNshasuru* "to give thanks, be thankful", with two of its potential alignments shown.

There are four main word-formation effects in Japanese which complicate alignment, each of which we discussed in Section 2.3. They are: okurigana, sequential voicing, sound euphony and grapheme gapping. The first case, okurigana, describes inflectional suffixes and how they may change. The remaining three cases describe effects which can occur when kanji form compounds. The last case, grapheme-gapping, needs little discussion: it occurs very rarely (under 0.1% of the evaluation set) and is productive only in very limited forms, as noted by Baldwin and Tanaka (1999b). It thus requires no special handling. In general however, each of these effects makes alignment more difficult because they add variability to kanji pronunciation.

In the remainder of this section, we firstly describe the existing algorithm in detail before going on to describe our proposed improvements. Finally, we evaluate our results, discuss the improvements and their re-integration back into FOKS.

Existing algorithm

Overview

We now examine the baseline GP-alignment algorithm used by FOKS, namely Baldwin and Tanaka's (1999a) unsupervised iterative algorithm. A high-level depiction of this

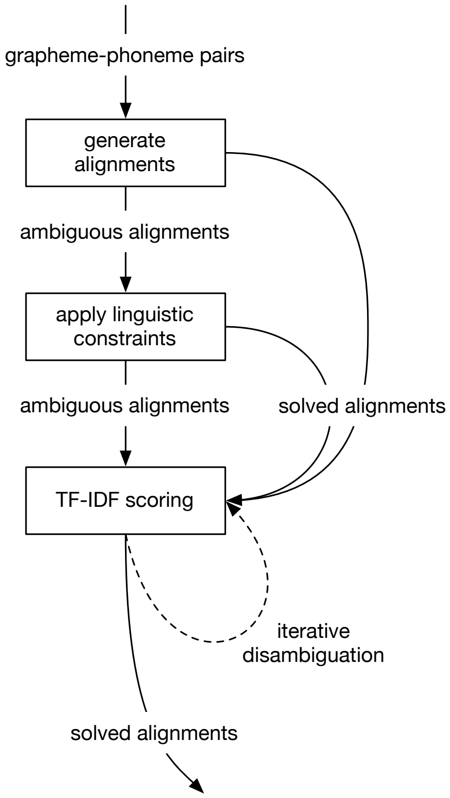


Figure 5.4: The TF-IDF based alignment algorithm

algorithm is given in Figure 5.4. Firstly all potential segmentations and alignments for input entries are created. In general, each entry may have potential segmentations and alignments per segmentation numbering exponentially in the entry's length. As in any alignment task where two strings of length l and m respectively need to be aligned, there are 2^{lm} possible alignments before applying constraints (Brown *et al.* 1993).

Fortunately, some simple linguistic constraints avoid this worst-case number of alignments to consider. Alignments must be strictly linear, each grapheme must align to at least one phoneme, and kana in the grapheme string must align exactly to their equivalents in the phoneme string. Further constraints used to prune entries include matching okurigana to pre-clustered variants and forcing script boundaries (except kanji to hiragana boundaries) to correspond to segment boundaries.

Based on the linguistic constraints, we can reasonably expect to have uniquely determined some number of alignments for any sufficiently diverse data set.[2] The uniquely determined alignments and the remaining ambiguous alignments are both used separately to seed frequency counts for the TF-IDF model.

TF-IDF is a family of models originally developed for IR tasks, combining the TF (term frequency) and IDF (inverse document frequency) heuristics (Salton and Buckley 1988). In the GP alignment task, they mediate the tension between oversegmenting and undersegmenting. The TF value is largest for the most frequently occurring GP pair given any grapheme; an oversegmented alignment produces rarer segments with lower frequency, penalising the TF score. The IDF value on the other hand is largest for segments which occur in a wide variety of contexts, and penalises undersegmenting.

TF-IDF Alignment

We use a modified version of the TF-IDF model which takes into account the differing level of confidence we have in our frequency counts between solved (freq_s) and ambiguous (freq_u) alignments (Baldwin and Tanaka 2000). For each alignment, we count the occurrence of each grapheme segment $\langle g \rangle$, of each aligned grapheme-phoneme segment pair $\langle g, p \rangle$, and of the same pair with one additional pair of context on either side $\langle g, p, c \rangle$. For any frequency lookup, the w_s and w_u constants provide a weighting between information from solved and ambiguous alignments:

$$\mathrm{wtf}(x) = w_s \times \mathrm{freq}_s(x) + w_u \times \mathrm{freq}_u(x) \tag{5.2}$$

To score a potential alignment, we calculate the TF and IDF scores for each grapheme-phoneme segment pair and multiply them together as in Equations 5.3-5.5. The score for the whole alignment is the average of the scores for every pair which contains a kanji character, since these are the non-trivial pairs. The constant α is intended as a smoothing factor for the TF and IDF scores. It must be assigned such that $0 < \alpha < w_u \leq w_s$.

$$\mathrm{TF}(g,p) = \frac{\mathrm{wtf}(\langle g,p \rangle) - w_u + \alpha}{\mathrm{wtf}(\langle g \rangle)} \tag{5.3}$$

[2] Notable exceptions to this are dictionaries of 4-kanji proverbs, such as the 4JWORDS electronic dictionary, whose entries' grapheme forms lack kana to help eliminate possible alignments.

$$\mathrm{IDF}(g,p,c) = \log(\frac{\mathrm{wtf}(\langle g,p \rangle)}{\mathrm{wtf}(\langle g,p,c \rangle) - w_u + \alpha}) \tag{5.4}$$

$$\mathrm{score}(g,p,c) = \mathrm{TF}(g,p) \times \mathrm{IDF}(g,p,c) \tag{5.5}$$

Once all potential alignments have been scored, the highest-scoring alignment is chosen to disambiguate its entry. Its counts are removed from the unsolved pool and added to the solved pool, and algorithm reiterates with updated counts. In this way entries are iteratively disambiguated until no more remain, and the algorithm is complete.

The iterative algorithm is effective but extremely expensive, with two main components to the cost. The first is the number of potential alignments per entry, of which there are exponentially many. In particular, long entries with many kanji and no kana to constrain them have prohibitively large numbers of possible alignments. These cases bloat the number of potential alignments to be rescored on each iteration so much that including them makes our main algorithm infeasibly expensive: the longest few entries together have the same number of potential alignments as the entire rest of the dictionary entries together. The only way to tackle this component is to find additional constraints which will reduce the number of alignments.

The second cost is in the scoring loop. Suppose there are n alignments pairs, each with p possible alignments. Then the cost of the iterative rescoring loop is $\mathrm{O}(n^2p^2)$. Even having removed the problem cases above, if p is still high on average, the problem will prove intractable for suitably large n. For example, the evaluation set used by Baldwin and Tanaka (1999a) has 5000 word-reading entries, yet the EDICT dictionary has over 170,000 entries at the time of writing, representing an expected increase in computation time of three orders of magnitude. Although this could be mitigated by simply breaking the input down into smaller subsets for processing, it is desirable to process all the data in the same iterative loop, since this gives greatest consistency of alignment.

Strategies to better constrain alignments and alternatives to iterative scoring form the basis for our attempts at modifying the algorithm.

Modified algorithm

The modified algorithm diverges from the unsupervised algorithm in three main respects. Firstly, we separate out okurigana handling into a separate step after alignment, benefiting both efficiency and error measurement. Secondly, a reading model is introduced based on the KANJIDIC electronic dictionary[3] and is used to disambiguate the majority of remaining cases before the TF-IDF model is reached. Thirdly, we provide a maximum alignment size cutoff above which we use a simplified non-iterative alignment algorithm which meets resource constraints for problem cases. We discuss these changes below.

Separating okurigana handling

The okurigana handling in the original algorithm involves pre-clustering okurigana alternates, and attempting to restrict alignments to match these alternates wherever possible. Whilst this constraint does help reduce potential alignments, it also limits the application of the stronger constraint that script boundaries in the grapheme string must correspond to segment boundaries (i.e. every occurrence of a kanji–hiragana script boundary must be considered as a potential okurigana site). If okurigana detection is left as a post-processing task, we can strengthen this constraint to include all script boundaries, instead of omitting kanji-to-hiragana boundaries. This in turn provides a larger gain than the original okurigana constraint, since more entries are fully disambiguated.

The GP-alignment task is then split into two parts: a pure alignment task, which can be carried out as per the original algorithm, and a separate okurigana detection task. This redesign also allows us to separately evaluate the error introduced during alignment, and that introduced during okurigana detection, and thus allows us to experiment more freely with possible models.

Short and long entries

Ultimately, any method which considers all possible alignments for a long entry will not scale well, since potential alignments increase exponentially with input length. We can

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

however extend the applicability of the algorithms considered by simply disambiguating long entries in a non-iterative manner.

The number of potential alignments for an entry can be estimated directly from the number of consecutive kanji. Our approach is to simply count the number of consecutive kanji in the grapheme string. If this number is above a given threshold, we delay alignment until all the short entries have been aligned. We then use the richer statistical model to align all the long entries in a single pass, without holding their potential alignments in memory.

Although long entries were not an issue in our evaluation set, a threshold set experimentally to 5 consecutive kanji worked well using the EDICT dictionary as input, where such entries can prove difficult.

Reading model

For the pure alignment task, we added an additional reading model which disambiguates entries by eliminating alignments whose single kanji readings do not correspond to those in the Kanjidic and KANJD212 electronic dictionaries. These dictionaries list common readings for all kanji in the JIS X 0208-1990 and JIS X 0212-1990 standards respectively, covering 12154 kanji in total. Effectively, we are applying the closed world assumption and allowing only those alignment candidates for which each grapheme unit is associated with a known reading. Only in the instance of over-constraint, i.e. every GP alignment containing at least one unattested reading for a grapheme unit, do we relax this constraint over the overall alignment candidate space for the given grapheme string.

A simple example of disambiguation using the reading model is that of 一両 *i-chi-ryo-u* "one vehicle" as shown in Figure 5.5. Since only one of the potential alignments is compatible with the known readings, we then select it as the correct alignment. As an indication of the effectiveness of the reading model, our initial constraints uniquely determine 31.1% of the entries in the EDICT dictionary.[4] The reading model disambiguates a further 60.6% of entries, effectively decreasing the input to the iterative alignment algorithm by an order of magnitude, to the remaining 8.3%.

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

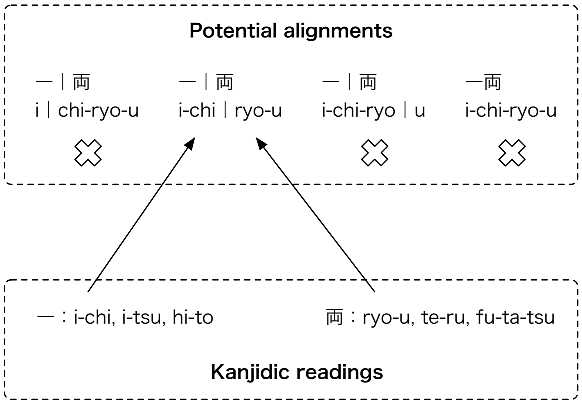


Figure 5.5: Disambiguation using the reading model

Heuristic variants

We could continue to use the original TF-IDF model over the residue which is not disambiguated by the reading model, although the type of input has changed considerably after passing through the reading model. Since the reading model is likely to fully disambiguate any entry containing only single kanji segments, the only remaining ambiguous models are likely to be those with solutions containing multi-kanji segments (which do not occur in either KANJIDIC or KANJD212); an instance of a multi-kanji segment is our earlier example 風邪 *kaze* "common cold". With this in mind, we compare the original TF-IDF model (our baseline) with similar models using TF only, IDF only, or random selection to choose which entry/alignment to disambiguate next.

Okurigana detection

We similarly wish to determine what form of okurigana detection and realignment model is most appropriate. Since the majority of entries in the EDICT dictionary (our main experimental data set) which contain potential okurigana sites (i.e. kanji followed by hiragana) do contain okurigana in some form, we use as our baseline the simple assumption that every such site is an instance of okurigana. In this manner, the baseline simply removes every kanji-to-kana segment boundary. As a small enhancement, the boundary is not removed if the tailing kana segment is one of the hiragana particles *no*, *ga* or *ni*, which frequently occur alone.

We consider three alternative okurigana models to compare to our baseline, of increasing complexity and expected coverage. Firstly, the Kanjidic dictionary contains common okurigana suffixes for some kanji with conjugating entries. Thus our first model uses these suffixes verbatim for okurigana detection. The coverage of okurigana suffixes in Kanjidic is somewhat patchy, so in our second model, in addition to Kanjidic suffixes, we also perform a frequency count over all potential okurigana sites in the EDICT dictionary, and include any occurrences above a set threshold as okurigana.

Finally, most instances of okurigana are due to verb conjugation. As well as taking straight suffixes from the previous models, this final model harvests verbs from EDICT. Most verb entries in EDICT have a tag marking them as *ichidan*, *godan* or *suru* verbs.[5] The verb type and stem allow us to conjugate regular verbs variously, giving us a large number of new okurigana suffixes not present in the previous models. In order to improve accuracy, all three methods fall back to the baseline method if they do not detect any okurigana.

Evaluation

Having teased apart the alignment and okurigana detection algorithms, we are in a position to separately evaluate their performance. Our test set for the combined task consists of 5000 randomly chosen and manually aligned examples from EDICT, from which we then separated out an individual evaluation set for each sub-task.

Since we are also interested in efficiency, we provide execution time as measured by elapsed time on a Pentium 4 desktop PC. Our emphasis however is on the *relative* time taken by different algorithms rather than the exact time as measured.

In the following subsection we first evaluate alignment and okurigana detection separately, then we evaluate okurigana detection, and finally we assess performance over the combined task.

[5] The tagset for EDICT verbs is larger than this, but the additional tags largely mark subclasses and exceptions of the three main classes, which we ignore for the sake of simplicity.

	Random	TF	IDF	TF-IDF
Iterative	47.8	23.7	94.7	93.4
Single-pass	47.3	23.6	90.5	90.8
Iterative + kanjidic	94.4	92.9	98.0	97.9

Table 5.1: Alignment accuracy across models.

	Random	TF	IDF	TF-IDF
Iterative	0:10	24:10	22:47	21:54
Single-pass	0:10	0:11	0:09	0:10
Iterative + kanjidic	0:12	0:27	0:24	0:24

Table 5.2: Alignment execution time across models in minutes and seconds.

Alignment

We first compare the accuracy of the three main alignment algorithm variants along with several scoring variants for each, as given in Table 5.1. The *Iterative* methods pre-populate the frequency distributions with all potential alignments, and on each iteration they rescore all potential alignments and resolve the single best. Since rescoring every potential alignment each iteration is expensive, we also provide a *Single-pass* variant for comparison. The *Single-pass* method begins with the same alignment model, but uses only a single scoring round which determines the best alignment for every entry at once. The *Iterative + kanjidic* method uses readings from kanjidic as soft constraints on potential alignments; in practice this results in a significant reduction in the number of ambiguous alignments. After some experimentation, parameter values of 0.05 for α, and 2.5 for w_s and w_u were found to yield the best results, and were hence used to generate the results we discuss here.

For each of the non-random heuristics, we expect that the iterative version will achieve higher accuracy than the non-iterative version, since the statistical model is rebuilt each iteration adding the best example from the last. As such, this represents a time/accuracy trade-off, a fact confirmed by our data (see Table 5.2). The gain (2% in the case of TF-IDF, 4% for IDF alone) comes at the cost of an order of magnitude larger execution time, which also increases exponentially with the number of input entries.

In contrast, the Kanjidic model consistently achieves a very high accuracy regardless of the heuristic chosen. A large number of entries are immediately disambiguated by the Kanjidic model, thus initially improving accuracy and then facilitating use of more accurate statistics in the iterative algorithm without significant penalty to efficiency. We also expect the Kanjidic model's execution time to scale more moderately with the number of input entries than the original iterative algorithm, since a far lesser proportion of the entries require iterative disambiguation.

Comparing the individual heuristics at this stage, a surprise is that the IDF heuristic attains equivalent results to the TF-IDF heuristic, suggesting that broad occurrence of $\langle g, p \rangle$ pairs is a good indicator of their alignment probability. The TF heuristic in comparison performs worse than simply choosing randomly, suggesting that the proportion of times a grapheme occurs as the current $\langle g, p \rangle$ pair is a very poor indication of its alignment probability. From a different viewpoint, if TF guards against over-segmentation, then over-segmentations are not an issue in this task, where our goal is the maximal segmentation of the GP pair.

Okurigana detection

We now compare the performance of our okurigana detection algorithms. All the algorithms we compare are linear in the size of the input and thus run in much less time than the alignment phase, thus efficiency is not a significant criteria in choosing between them. The accuracy found by each model is shown in Table 5.3.

Model	Accuracy
Simple	98.1%
Kanjidic	98.3%
Co-occurrence	97.7%
Verb conjugation	97.7%

Table 5.3: Okurigana detection accuracy across models

Interestingly, the simple baseline model which assumes that every potential case of okurigana *is* okurigana performs extremely well, beaten only by the addition of the Kanjidic common okurigana stems. Adding more information to the model about valid okurigana occurrences even reduces the accuracy slightly over our test data.

Rather than indicating blanket properties of these models, the results suggest properties of our testing data. Since it consists entirely of dictionary entries without the common hiragana particles which would occur in open text, this greedy approach is very suitable, and suffers few of the shortcomings which it would normally face.

In open text, we would consistently expect additional language features between lexical items which would break the assumptions made by our simple model, and thus reduce its accuracy dramatically. In contrast, the full verb conjugation model would then be expected to perform best, since it has the most information to accurately detect cases of okurigana even in the presence of other features.

Combined task

Selecting the two models which performed best on our test data, we can now evaluate the pair on the combined task. For the alignment subtask, the IDF heuristic with Kanjidic was used. For the okurigana detection subtask, the simple algorithm is used. The results are shown in Table 5.4.

Status	**Count**	**Percentage**
Correct	4809	96.2%
Incorrect	191	3.8%
→ Gapping	6	0.1%
→ Alignment	163	3.3%
→ Okurigana	22	0.4%

Table 5.4: Best model accuracy for the combined task

A final accuracy of 96.2% was achieved, with the errors caused mostly in the alignment subtask. As predicted, grapheme gapping was a source of errors only in a small percentage of cases, justifying its exclusion from our model. This level of accuracy is equivalent to that of earlier models, yet it has been achieved with a much lower computational cost. Examples of incorrect alignment are given in Figure 5.6 below.

<table>
<tr><td>a.</td><td><i>Output</i></td><td>挟 | 撃 | <i>chi</i></td><td><i>hasa</i> | <i>miu</i> | <i>chi</i></td></tr>
<tr><td></td><td><i>Correct</i></td><td>挟 | 撃 | <i>chi</i></td><td><i>hasami</i> | <i>u</i> | <i>chi</i></td></tr>
<tr><td></td><td></td><td colspan="2">“pincer attack”</td></tr>
<tr><td>b.</td><td><i>Output</i></td><td>赤-<i>N</i> | 坊</td><td><i>akaN</i> | <i>bō</i></td></tr>
<tr><td></td><td><i>Correct</i></td><td>赤 | <i>N</i> | 坊</td><td><i>aka</i> | <i>N</i> | <i>bō</i></td></tr>
<tr><td></td><td></td><td colspan="2">“baby”</td></tr>
</table>

Figure 5.6: Examples of incorrect alignment in the combined task

Example (b) shows a typical alignment error, where one kanji has been attributed part of the reading of another. Example (b) on the other hand gives an error in okurigana detection, where the *N* kana is erroneously detected as an okurigana suffix of the 赤 kanji.

Improved alignment

We have decomposed the GP alignment task into an alignment subtask and an okurigana detection subtask, and explored various algorithm variants for use in both. In particular, the iterative IDF heuristic with a Kanjidic reading model provided the best accuracy in significantly less time than the original algorithm. For the okurigana detection subtask, a simple model outperformed more complicated models of conjugation due to peculiarities of dictionary entries as input to alignment.

The modified algorithm was suitable for use with the FOKS system, and was thus adopted as part of the build process for the new FOKS interface. Ideally, this work would have applicability to open text as well as dictionary interfaces. However, the basic unsupervised method relies heavily on getting a representative sample of readings from the input before attempting alignment. One way to circumvent this would be to bootstrap the method with alignments from an existing representative sample, say the EDICT dictionary, and then use the bootstrapped algorithm to align new pairs.

Okurigana detection remains the harder problem, for tasks which require it. The verb-conjugation model, despite its relatively poor performance for dictionary entries, suggests itself as the most fruitful approach to accurate detection for open text, and could easily be extended. In particular, the addition of conjugation suffixes of high-frequency irregular verbs would be a straightforward way to boost accuracy.

5.3 Usability enhancements

This section describes three basic enhancements to the dictionary, namely: the addition of a huge library of place names, and their display; the separation of the senses of polysemous kanji and words; and the ability to explain to a user how a query worked, and how the correct reading is structured. We discuss each of these in turn.

Place names and homographs

As part of error analysis of FOKS, Bilac *et al.* (2004) found that insufficient knowledge of proper nouns was a common error type, as discussed in Section 5.1. FOKS is especially useful for these proper nouns, since their pronunciation is idiosyncratic. However, FOKS relies on ENAMDICT for place names, which has two disadvantages. Firstly, ENAMDICT has limited coverage of place names in Japan. Secondly, it gives no cues as to relationships between places, only giving each place name a transliteration as its gloss.

We address both of these problems by constructing a simple gazetteer resource from data mined from Japan Post.[6] This resource provides a large number of place names not in ENAMDICT, but also provides some hierarchical structure which we can use to distinguish between the many homographs encountered in this exercise, structure which ENAMDICT does not provide. 114591 place names were mined; of these, roughly 69% had unique written forms, whereas 31% did not. Figure 5.7 shows the distribution of homography over the 79139 unique written forms mined.

The extremes of this distribution also suggest the limits of this data set. The most used place name found was 本町, with 315 places using this name, taking pronunciations *hoNchō*, *hoNmachi* or *motomachi*. However, this is better termed a suffix than a place name in its own right. More generally, there are a number of valid but high-frequency place names, such as 上野, with 76 matches. In reality, the senses of such place names – that is, the places they actually refer to – themselves vary in frequency. For 上野, the most prominent amongst

[6] The Japan Post Gazetteer: `http://www.csse.unimelb.edu.au/~lljy/datasets/#gazetteer`

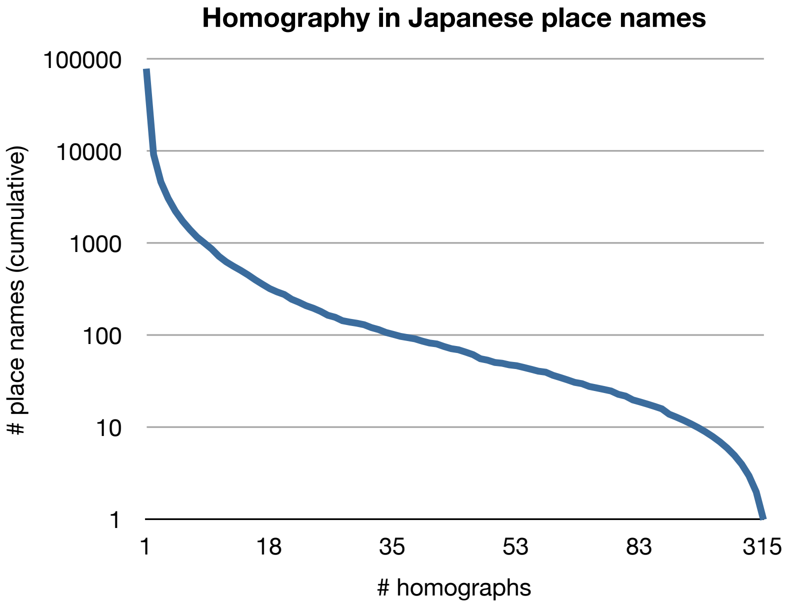


Figure 5.7: The distribution of place-name homography in Japanese. The vertical axis shows how many place names had at least n homographs, out of the 79139 unique place names encountered.

these senses is likely to be the *Ueno* area of Tokyo. FOKS currently lacks a useful way of determining and displaying the relative importance of different place name senses. Instead, we simply group these names by region, so that the user can find the place they were looking for provided they know its broad region. The more difficult task of determining the most prominent place name senses remains as future work. For each of these place names, we generated an automatic transliteration using Kakasi[7] to serve as a gloss.

Although the inclusion of this larger number of place names will allow the user to find them in the dictionary, it could also pollute the search results for common words. This is avoided in FOKS in two ways. Firstly, FOKS is primarily an indexing method for kanji compounds. Even in the case of 本町 above with 315 matching places, only a single entry is placed in the search results for a matching query. Secondly, users can filter by the type of query they are performing, or choose a default filter to apply to all queries. This provides an easy way to either focus on or exclude place names from results as desired. Search results

[7] http://kakasi.namazu.org/

aside, we have also made improvements to a word's translation page, which we now discuss.

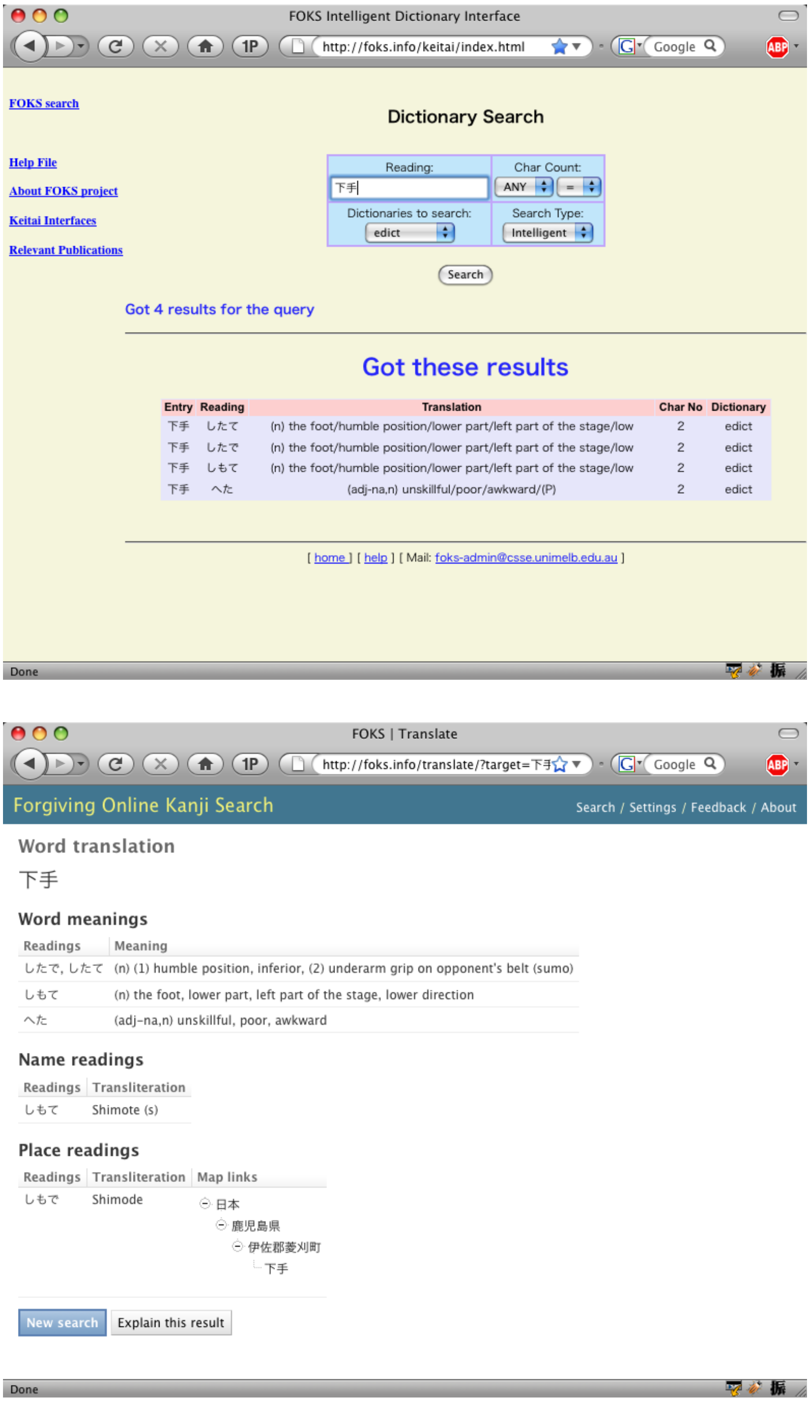
Sense separation

Both the new and old FOKS interfaces are targeted at achieving successful lookup of a word, in particular words containing kanji. However, the large amount of homography in Japanese means that the same word or compound might be used for several different purposes, and the two interfaces differ in how they present this information to the user. In the old interface, a gloss is shown for each word whose surface form matches exactly that of the successful query. When a word has multiple readings for the same gloss, this leads to repetition of the gloss. If many proper nouns are included in the translation results, they can pollute the results, depending on the purpose of the user's search.

The new interface takes a different approach in allowing the user to visualise the complex relationship between surface form, readings and senses. For purposes of comparison, Figure 5.8 provides screenshots of both systems for the translation of compound 下手. Firstly, the three main categories of sense – general word, person/company name and place name – are clearly demarcated in separate sections. This allows the user to skim to the section they care about. The general word section in turn separates out reading-senses (senses which occur with a specific reading), and groups them by pronunciation. For example, *shitate* and *shitade* are both listed as readings of the sense "humble position". The two sections of proper nouns both list not translations but transliterations (i.e. romanisations) of the proper nouns. The section for place names also provides a hierarchical display of the place name, showing country, prefecture, ward and town; in this case 下手 *shimode* is located in Kagoshima Prefecture. Clicking on any of the levels in the place hierarchy makes a Google Maps query, locating the place quickly and conveniently.

Query and reading explanation

The original form of FOKS search bases itself around plausible misreadings of kanji and kanji compounds, and indeed reverse engineers the most prominent word-formation effects in order to determine what a plausible reading would look like. However, there is a lack of transparency for users in how FOKS works, and why it gave them a particular search



FOKS Intelligent Dictionary Interface
http://foks.info/keitai/index.html
Google
FOKS search
Help File
About FOKS project
Keitai Interfaces
Relevant Publications
Search
[home] [help] [Mail: foks-admin@csse.unimelb.edu.au]
Done

Dictionary Search

Reading: 下手
Char Count: ANY =
Dictionaries to search: edict
Search Type: Intelligent

Got 4 results for the query

Got these results

Entry	Reading	Translation	Char No	Dictionary
下手	したて	(n) the foot/humble position/lower part/left part of the stage/low	2	edict
下手	したで	(n) the foot/humble position/lower part/left part of the stage/low	2	edict
下手	しもて	(n) the foot/humble position/lower part/left part of the stage/low	2	edict
下手	へた	(adj-na,n) unskillful/poor/awkward/(P)	2	edict

FOKS | Translate
http://foks.info/translate/?target=下手
Google
Forgiving Online Kanji Search
Search / Settings / Feedback / About
New search
Explain this result
Done

Word translation

下手

Word meanings

Readings	Meaning
したで, したて	(n) (1) humble position, inferior, (2) underarm grip on opponent's belt (sumo)
しもて	(n) the foot, lower part, left part of the stage, lower direction
へた	(adj-na,n) unskillful, poor, awkward

Name readings

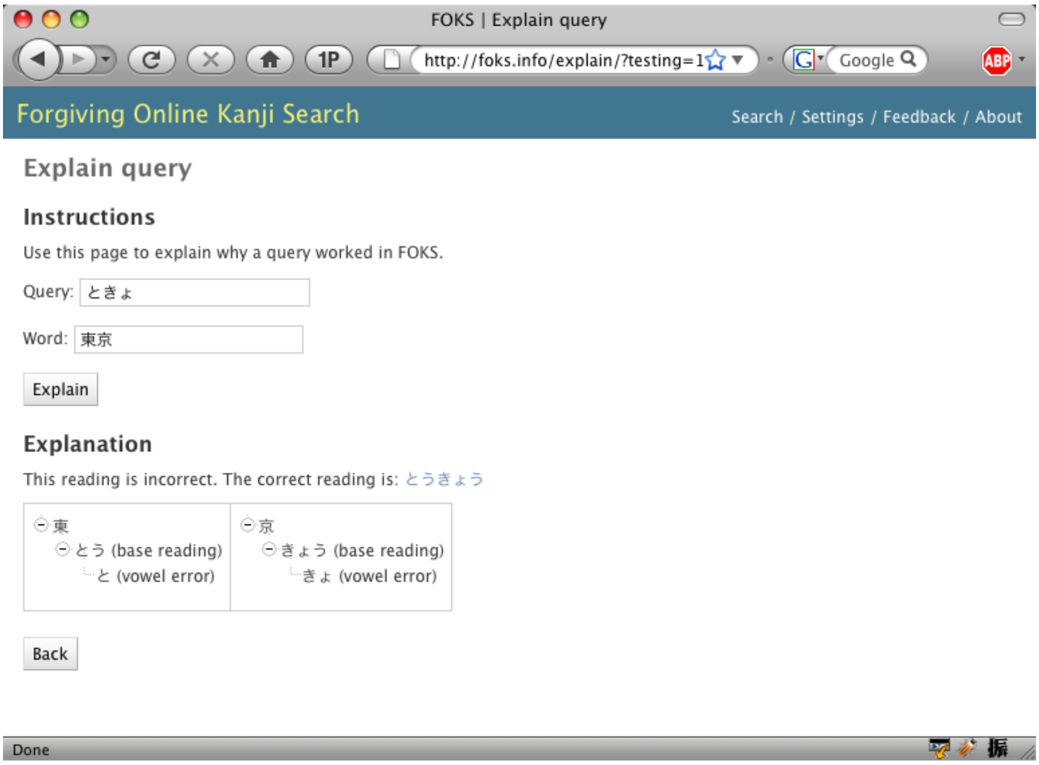
Readings	Transliteration
しもて	Shimote (s)

Place readings

Readings	Transliteration	Map links
しもで	Shimode	日本 鹿児島県 伊佐郡菱刈町 下手

Figure 5.8: The old and new translation pages shown for the compound 下手.

result. Since the new FOKS architecture generates the misreadings at query-time, it can also reverse-engineer queries to determine the relationship between the misreading and the target.


FOKS | Explain query
1P
http://foks.info/explain/?testing=1
Google
ABP
Forgiving Online Kanji Search
Search / Settings / Feedback / About
Explain
Back
Done

Explain query

Instructions

Use this page to explain why a query worked in FOKS.

Query: ときょ

Word: 東京

Explanation

This reading is incorrect. The correct reading is: とうきょう

東	京
とう (base reading) と (vowel error)	きょう (base reading) きょ (vowel error)

Figure 5.9: FOKS explaining the relationship between the query *tokyo* and the word 東京 *Tōkyō*.

Figure 5.9 gives an example for a common erroneous query, searching for 東京 *Tōkyō* using the reading *tokyo*; this is a misreading because both vowels should be long, but are instead short vowels in the query. The explanation firstly segments the reading into per-kanji readings, then explains the erroneous reading for each kanji as a correct base reading changed due to a vowel length error. The explanation also notes that the reading is incorrect, lists correct readings for the compound, and offers explanations for how the correct readings are derived. This last point is especially useful for learners who may be unsure how the reading of a long word can be attributed to readings of its parts.

In some cases, such as 海豚 *iruka* "dolphin", the correct reading is non-compositional. In these cases, the explanation cannot provide a segmented per-kanji reading; it instead simply states that the reading is correct but non-compositional, and thus special to this compound.

To recap our usability enhancements, we have extended the dictionary entries with a large number of proper nouns in the form of place names, yet managed these additional candidates through the use of search filters and appropriate translation structure to avoid them burdening search for common words. We have provided an augmented word translation page, which organises word readings and senses in a manner which both offers the user all the information about the word without overwhelming them. Finally, we have provided transparency to the search, by explaining to users how queries are reached, and providing them with the tools to better study and understand word formation principles in Japanese. Whilst we offer no strong evaluation for these changes, we make the case that they improve the value of the dictionary as a study tool, and may also improve retention of looked-up words.

5.4 Intelligent grapheme search

Overview

In Section 5.1 we gave an overview of FOKS, and discussed the shortcomings of the old system. They came in two forms: errors in searches by reading which could not be corrected, and searches by grapheme which were limited to exact matches. Having discussed and evaluated carefully various models of graphemic similarity, we are now in a position to provide this form of error correction to FOKS searches.

When we originally developed our graphemic neighbourhood models, we intended to provide correction for mistaken searches by reading. Our motivation came from Bilac *et al.*'s (2004) log analysis of FOKS, which picked up a small number of these cases. Although the number itself was low, the log data itself was highly constrained: it only contained successful searches through FOKS, which only corrected for reading errors. Thus, to appear in the logs at all, such grapheme-confusion cases needed to coincide with some other form of reading error which FOKS *did* support. We wouldn't normally expect grapheme-confusion and other forms of reading error to coincide, so the true prevalence of this error type might be significant. However, allowing search for a word using a reading of a near-homograph represents a trade-off, since the many readings available for any kanji would swell the potential candidate list for any search enormously. An alternative view of this problem is that the link between a compound's reading and that of near-homographs is tenuous; attempting to correct such errors forces us to cast our net too wide. Either the results will appear, but will be lowly ranked, or they will displace more frequent error types, and thus decrease search performance.

For these reasons, we extended search by grapheme – which had been exact match only – into an error-correcting search, with the potential to both correct errors based on graphemic similarity and also to allow intentional search of unknown characters, using known neighbours. This section describes the details of this new error-correction model.

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 substitute readings for kanji in our query. A unigram approximation leads us to Equation 5.6 below, where q is the query given by the user, and w is the desired word:

$$\begin{aligned} \Pr(w|q) &\propto \Pr(w)\Pr(q|w) \\ &= \Pr(w)\prod_i \Pr(q_i|w, q_0 \ldots q_{i-1}) \\ &\approx \Pr(w)\prod_i \Pr(q_i|w_i) \end{aligned} \tag{5.6}$$

The final line of Equation 5.6 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.

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, we expect that users will tend to confuse unknown characters with characters that they are already familiar with. This is a reasonable assumption 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:

$$\Pr(q_i|w_i) \approx \frac{\Pr(q_i)s(q_i, w_i)}{\sum_j \Pr(q_{i,j})s(q_{i,j}, w_i)} \tag{5.7}$$

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 $[0, \infty)$.

In order to maximise the accessibility of this form of search, we must find the appropriate trade-off between providing enough candidates and limiting the noise in the candidates present. 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 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.

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. In contrast, grapheme-based search may have only one exact match, so additional secondary candidates need not be in direct competition with existing search practices.

Finally, we can consider the theoretical accessibility improvement of this form of search. 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, we get the accessibility curve found in Figure 5.10. Our baseline is a single kanji accessible

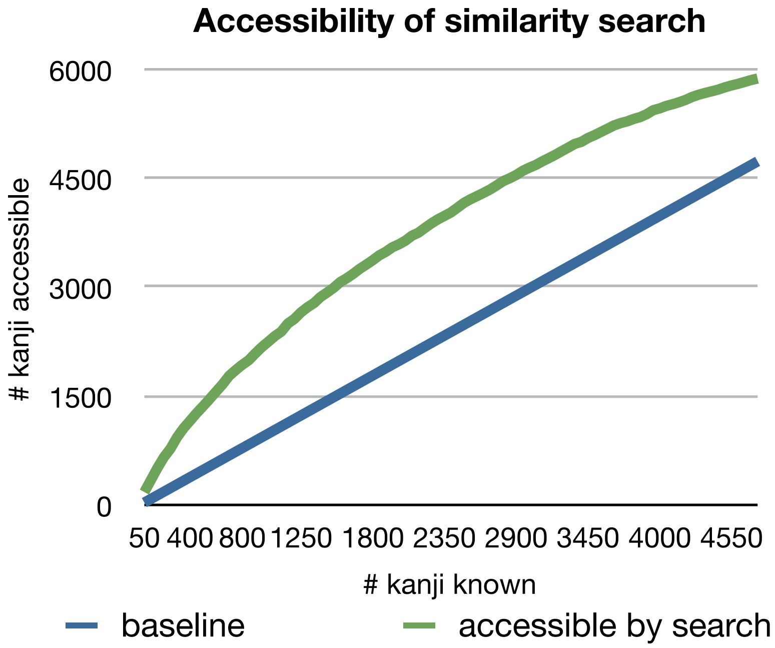


Figure 5.10: The potential accessibility improvement of kanji similarity search

for each kanji learned. Given that the actual number of usable neighbours is small and our precision within top-5 candidates of 0.228, we will need to expose the user to a larger set of candidates in order to get this level of improvement.

In order to determine how this form of search is actually used by learners, we now look to log analysis.

5.5 Log analysis

This chapter has described a broad series of improvements to the FOKS dictionary, extending it with the new intelligent grapheme search, and providing a large number of usability enhancements. Many of these enhancements are difficult to evaluate directly. Dictionary search takes relatively unconstrained and free-form input from users, and this can be hard to interpret in the resulting logs. We address this problem in Chapter 6 in a more constrained testing application, where we provoke errors from our users and can thus eval-

Language	**Visits**		**Cumulative time on site (hours)**	
Japanese	42798	79.5%	1785	66.1%
English	6781	12.6%	623	23.0%
Chinese	1164	2.2%	148	5.5%
Turkish	765	1.4%	20	0.7%
Other	2301	4.3%	126	4.7%
Total	53809	100.0%	2702	100.0%

Table 5.5: FOKS usage as reported by Google Analytics in the period from October 2007 to March 2009, broken down by browser reported language.

uate our error models more rigourously. Nonetheless, the broad usage patterns of FOKS can still be examined. The new system with grapheme error-correction was implemented in October 2007, and our results are based on logs taken from then until late March 2009.

Over this period, Google Analytics was used to measure site traffic and get a better understanding of user location and language background. The majority of queries originated from within Japan, which is unsurprising since both native speakers and advanced learners are likely to be based there. Table 5.5 shows the breakdown of site usage by browser reported language. Note that the browser reported language may differ from a user's actual first language; for example a native German speaker may be running Japanese browser software if they are an advanced learner, or working in a Japanese computing environment. Nonetheless, it gives us an indication of usage patterns. 80% of visits are Japanese-language, perhaps indicating a large pool of native-speakers as users, with English language visits at 13%. Smaller pools of Chinese and Turkish speakers are also significant users. However, when measured by time actually spent on the site, the share given to Japanese language users reduces to 66%, since they spend less time on average on the site than other language groups.

We now turn to query logs. Any interpretation of such logs bases itself on a model of user behaviour. Our basic assumptions are that a user is looking for a particular word whose kanji form is available, known or recognisable to them. When a user performs a search, they are thus presented with a list of scored word-reading results, each with a "Translate" button next to it. When a user clicks on this button, FOKS captures their decision and displays a detailed translation of the word. This is presumed to be the word the user intended to find,

	Full query		Partial query	Total
	Intelligent	*Simple*		
By grapheme	997	16756	81018	98771
By phoneme	754	4262	22912	27928
Total	1751	21018	103930	126699

Table 5.6: Number of FOKS searches logged between October 2007 and March 2009, arranged by search type.

and thus considered a successful search. We call such queries *full queries*, and those where the user does not select any word for translation *partial queries*, after Bilac *et al.* (2004). Partial queries give us little useful information, whereas full queries allow us to compare the query with the word found and determine if any error-correction was used.

This model represents only one possible series of user interactions, but we assume it is the dominant user behaviour. Other behaviours include: clicking on translations of unknown words mid-search out of curiosity, whether or not their desired word is present in the results; looking for a word of known pronunciation but unknown meaning, thus clicking on several translations to determine which matches the known meaning; searching to determine the pronunciation of a word of known form; searching to determine the form of a word of known pronunciation; and others. Distinguishing between these behaviours would be an interesting exercise, but is not the focus of this thesis, hence our assumption of a dominant behaviour.

Using these assumptions, basic search statistics can be compiled, as shown in Table 5.6. Roughly 29% of grapheme searches and 30% of phoneme searches are full queries, the remainder are partial queries. Of the full queries, we divided them into queries which were correct (i.e. exact-matches), and those which contained some error which our error models corrected. The former type of query we call "simple", the latter "intelligent". Focusing on the full queries, searches by grapheme are over 5 times more prevalent than those by phoneme, perhaps indicating the common practice of copying and pasting an unknown word into the dictionary. However, they enjoy a comparable share of the intelligent searches.

Since one feature of the rebuilt architecture was the ability to reverse-engineer the relationship between a query by reading and a word, we can give an automated breakdown of the prevalence of reading error types. This breakdown is given in Table 5.7. Inadequate

Error type	**Frequency**		**Example**	
			Query	*Word*
Choice of reading	547	72.0%	*noriru*	乗る *noru* "to get on"
Vowel length	92	12.1%	*tokyo*	東京 *tōkyō* "Tokyo"
Non-compositional reading	59	7.8%	*umibuta*	海豚 *iruka* "dolphin"
Sequential voicing	40	5.3%	*hayahaya*	早々 *hayabaya* "early"
Sound euphony	13	1.7%	*fukuki*	復帰 *fuk-ki* "return"
Palatalisation	9	1.2%	*seNko*	選挙 *seNkyo* "election"
Total	760	100.0%		

Table 5.7: Automatically determined error types for intelligent queries by reading.

choice of reading the most prevalent error type, accounting for 72% of errors. Vowel length errors account for a further 12%, followed by the other error types in smaller proportions. This drop-off differs somewhat from Bilac *et al.*'s (2004) earlier log analysis of the original system, in that we encounter a far smaller proportion of gemination errors, voicing errors and errors due to non-compositional readings. This could be due to subtle changes in search result ranking of the new system, or even in the demographic of its users. Nonetheless, each of the broad error types is successfully corrected for by the new system.

Of the 997 grapheme-similarity searches, the forms of analysis open to us are limited. We can however test the assumption inherent in our error model, namely that kanji will be mistaken for their more frequent but visually similar neighbours. Looking at every intelligent grapheme query, and every target kanji t and query kanji q which differ, let us measure $\log(\frac{\Pr(q)}{\Pr(t)})$. This log-ratio indicates the relative frequency of kanji errors compared to the kanji they are derived from. If strongly positive, this would suggest either that people confuse kanji for their more frequent neighbours or that people use more frequent neighbours to query for rare kanji. If strongly negative, the result would be problematic: it would seem to suggest that people use rare kanji to query for common ones, a highly unlikely scenario. Instead, we would surmise either that unexpected confusion errors were occurring in selecting the appropriate search result, or alternatively that the corpus we use to measure kanji frequencies differs significantly from the order in which learners acquire kanji.

Plotting a histogram of the log ratio, Figure 5.11 shows that the distribution is remarkably even, showing a very slight negative log-ratio on average. This suggests that, on

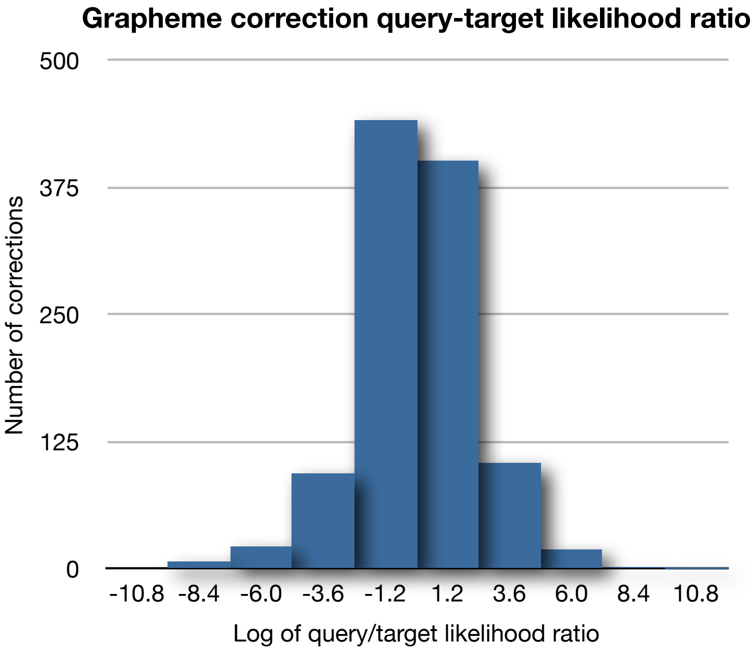


Figure 5.11: Histogram of log query/target likelihood ratio for all corrected grapheme queries, taken on a per-kanji basis.

average, kanji are mistaken for visual neighbours of comparable but not necessarily higher frequency, thus providing evidence for neither of the outcomes discussed above. Instead, a plausible explanation could be that learners tend to confuse kanji pairs on the periphery of their knowledge, rather than mistaking rare kanji for higher frequency known ones.

An alternative hypothesis is that users are distracted by the additional search results, which anecdotally seem to contain individually frequent kanji in rarer combinations, and may thus select a word other than their original target for translation. For example, a search for 海豚 *iruka* "dolphin" also yields the surname 海家 *Umiie* in the search results, which is a rare composition of the highly frequent kanji 海 and 家. Combinations such as this might be interesting enough to distract the user from their original search. Whilst this behaviour is not technically an error, it would break our simplifying assumption that the first word a user selects in search results was the original word used to formulate the query. This case could potentially be investigated through more sophisticated analysis of log behaviour, since it should be identifiable by the user selecting multiple query results for translation, one of which may be an exact match for the original word. Such analysis is beyond the scope of this thesis; we leave it for future work.

5.6 Conclusion

If learning words is the main obstacle to linguistic competency, then the dictionary is the gateway to this competency, since it is the primary resource for unlocking the meaning behind otherwise foreign words. In this chapter, we examined the FOKS dictionary, with the aim of making it a better resource for learners in their autonomous study. We took a holistic approach, taking apart the dictionary and rebuilding it with improved grapheme-phoneme alignment, improved place-name coverage, better word translation, explanations of word readings, and finally, a new form of search which allows users to substitute unknown kanji for their visually neighbours in queries. Our log analysis shows that the number of intelligent graphemic queries was higher than that of intelligent phonetic queries, indicating the adoption of this form of search by the existing user base.

There are many directions which future research could take in improving further upon this form of search. Log data accumulating from intelligent graphemic queries can be stored and used as a means to evaluate improvements to graphemic error correction. Furthermore, such data also serves as yet another means of evaluating orthographic neighbourhood models, which still contain substantial noise and hence potential for improvement. Our automatically generated explanations for both correct and plausibly incorrect readings of kanji compounds could be extended to graphemic errors, identifying the strokes or components which differ between the two characters and highlighting the differences between them. There is also potential for other forms of graphemic relationships such as subsumption to be used as graphemic search methods. For example, 動 could be used to search for 働, not based on their similarity, but based on its strict membership within the latter kanji. Such a search method has already been trialled by Jin (2008), but with further work the comparison between graphemic search methods could help us to better understand what kinds of partial information are most naturally available and most easily expressible in search queries.

Ultimately, at the core of the new FOKS implementation we have discussed lies two main error models: a model of plausible mispronunciation, and a model of plausible misrecognition. Yet the use of these models need not be limited to the dictionary. We now continue by applying these models of plausible error in language testing domain, in order that students can better self-evaluate their study.

Chapter 6

Testing and drilling

In the previous chapter (Chapter 5), we examined the effect of augmenting an existing dictionary with the graphemic error models we developed earlier (Chapter 4). In this chapter, we examine the transferability of both phonemic and graphemic error models to the domain of language testing and drilling, using a new system called Kanji Tester.

We begin in Section 6.1 by discussing why testing is such an attractive application space for the graphemic and phonetic error models we used in search. We then propose criteria with which to situate Kanji Tester amongst other forms of testing. Since Kanji Tester is modelled after the Japanese Language Proficiency Test (JLPT), Section 6.2 describes the JLPT in more detail and justifies our interest in emulating it with our automated tests. With this background complete, Section 6.3 sets out describing the Kanji Tester system in detail, from the perspective of both its use and its implementation. It discusses our user modelling, from the granularity of modelling chosen to the algorithms used for update and for question generation. Section 6.4 then evaluates Kanji Tester through analysis of its usage logs, exploring user demographics, user ability, power users and the overall effectiveness of our error models. Finally, we conclude in Section 6.6 with our findings.

6.1 From search to testing

Testing as an application space

We saw in Chapter 3 that tools for learning vocabulary can basically be divided into two forms: reading aids and tests. These mirror the two ways in which vocabulary can be learned, implicitly or explicitly. So far, we have focused on dictionaries, since they are the simplest reading aid and first point of contact for learners. We now complete our study by considering language testing.

There are several strong reasons to choose testing as our next application area. Without exception, all forms of explicit vocabulary study focus heavily on testing. This is because as learners we are poor at self-evaluation; by testing ourselves we can objectively discover the limits of our knowledge, and thus begin to expand them. Testing also closes another loop: motivation. A strong motivation may provide the impetus for early study, but without evidence the study is paying off, it may wane. Testing provides this evidence.

Aside from its heavy use in language learning, testing is also an attractive secondary application area for the error models we generated for search. In search, the learner provides us with their response to an unknown stimulus, and our task is to recover the original stimulus word. Since user queries are noisy, we are often unable to recover the original word. Such cases can be attributed to one of three issues:

1. *Insufficient user knowledge*: The user did not know enough accurate information about the word to formulate a useful query.

2. *Poor modelling of error prevalence*: The search results contained the desired word, but the rank was too low for the user to find it.

3. *Poor coverage of errors*: The search results did not contain the desired word, despite a correct query formulation by the user.

Unfortunately, we can't distinguish between the first error type, which can't be reduced, and the latter two, which can. In testing however, the system generating the test provides the stimulus to which the user must respond. This more constrained form of interaction means that user responses can be used directly to evaluate our error models and thereby form a basis for their improvement.

The ideal test

It makes little sense to investigate language testing without considering what makes a test worthwhile. We discussed in Section 3.4 the two main quality attributes of tests, namely validity and reliability, and these are certainly desirable properties for any test. However, whilst they are useful for comparing two tests with the same aims, they offer little guidance as to what to test in the first place. For example, many tests from flashcards to large scale accredited language testing can achieve high validity and high reliability, albeit measuring fundamentally different properties of the learner.

In order to make sense of the variety of tests used in language study, we instead propose two scales along which we arrange these and other tests measuring language proficiency. The first is the *scope* of the test: does it attempt to measure all aspects of proficiency, a limited subset, or simply basic knowledge of a few words? The second scale we call the test's *availability*, a measure which encapsulates both its accessibility and any obstacles to taking it: can the test be taken on demand, or only rarely? does it take a long time, or cost money? Figure 6.1 arranges several forms of tests along these scales.

We justify these scales by revisiting the actual purpose of testing. Testing is used to provide information supporting decision making. In the context of language learning, such decisions range from *do I need to keep studying this word?*, to the much broader *what progress have I made?* and *what study methods or classes were most effective for me?* Simpler decisions can already be made quickly and effectively with the use of basic tools such as flashcards. It is the more complex decisions which require much stronger supporting information. Large-scale formal tests of proficiency, such as Japanese Language Proficiency Test (JLPT), are designed with wide scope to meet this need, but currently this scope comes at a great cost to availability. For many learners, the information provided by this form of test comes too infrequently to usefully inform their study.

Imagine an oracle which, when asked, could accurately and instantly determine the proficiency of a learner, across whichever dimensions this proficiency is best measured. Such an

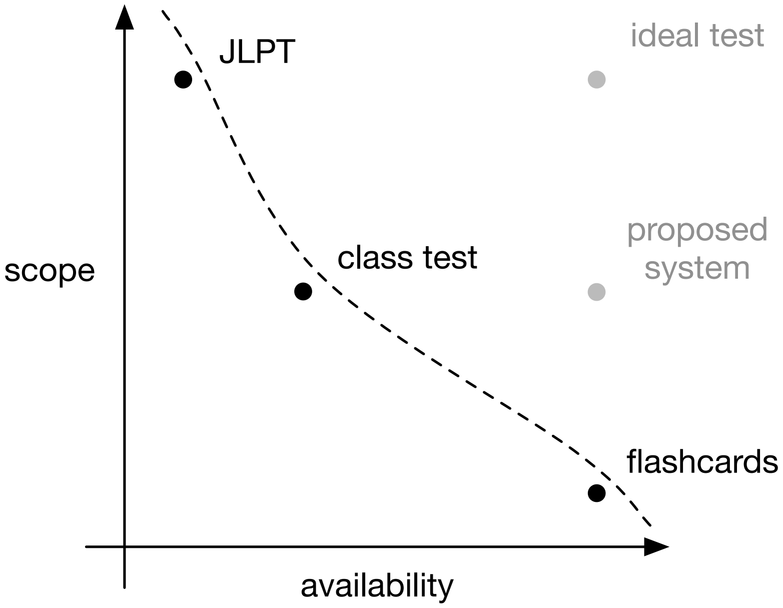


Figure 6.1: The various types of Japanese proficiency testing as they vary in scope and availability. The dotted line represents the current state-of-the-art.

oracle would reduce or eliminate the need for formal testing, and would itself be considered the ideal test. This is indeed the underlying goal of Intelligent Tutoring Systems, where the user's study of content material informs a user model which in turn determines what will be presented to them next based on their current proficiency (Shute and Psotka 1994:10). From another perspective, these platforms offer a form of continuous testing where learning and testing are either combined into a single activity or rapidly alternated between. The more the learner uses the system, the more information it has to use to aid the learner.

Without the pre-existing user data which such a platform can provide our ideal test would share much with existing broad-scope manually-constructed tests such as the JLPT. The main problem with such tests currently is that they cannot be taken more than once without damaging their validity; once a learner has seen a test, their performance on subsequent attempts at the same test may be affected as much by their memory as by their ability. For this reason substantive tests are usually rewritten for every use by expert language instructors, at significant time and cost. The transition to computer-based testing allows postponing of this problem through the manual construction of large question banks, but the fundamental issue remains: if individuals are allowed to test repeatedly, validity suffers.

In contrast, today's automatically generated tests have high availability, but are universally primitive in the forms of questions they can offer, which in turn reduces their validity in assessing language proficiency.

In this chapter we propose a new system called Kanji Tester which generates automated vocabulary tests. Since tests are generated on demand, it meets the ideal test's goal of availability. It sits at an intermediate level in terms of scope, since the types of questions it can ask a learner are constrained by what can be reasonably automated. However, its intelligent error models allow it to generate a different test every time, and thus to approach the scope of some human generated tests. Since the system allows a learner to test and re-test themselves repeatedly, we call its tests *drills*.

To simplify Kanji Tester's construction and to gain it a larger audience, we attempted to emulate the well-known Japanese Language Proficiency Test, which we now discuss.

6.2 Japanese Language Proficiency Test

Overview of the JLPT

The Japanese Language Proficiency Test (JLPT) is a family of examinations run biannually by the Japan Foundation. This section provides an overview; the interested reader should visit the JLPT site[1] run by the Japan Foundation (2009). It currently consists of four separate exams at targeted at different levels of difficulty, from Level 4 (easiest) to Level 1 (hardest), as shown in Table 6.1. Level 1 is designed to be the level required to live and interact in Japanese society, and tests all the kanji in the government proscribed 常用 *jyōyō* "daily use" set which native speakers are expected to know upon completing secondary education.

The test itself consists entirely of multiple-choice questions. It contains three parts, as shown in Table 6.2: Writing and Vocabulary; Listening; and Reading and Grammar. From 2010, the writing section of the test will be merged in to the reading section which will then contain 75% of the total points in the test. Multiple-choice tests such as the JLPT give significant scope for guessing of unknown answers . Since JLPT questions have four potential answers for the learner to choose from, unknown questions may be correctly

[1] `http://www.jlpt.jp/`

Level	# kanji	# words	Listening	Hours of study	Pass mark
4	100	800	Beginner	150	60%
3	300	1500	Basic	300	60%
2	1000	6000	Intermediate	600	60%
1	2000	10000	Advanced	900	70%

Table 6.1: The four levels of the JLPT, with approximate numbers of kanji and words which learners should know when testing at each level, and estimated hours of study required. From Japan Foundation (2009).

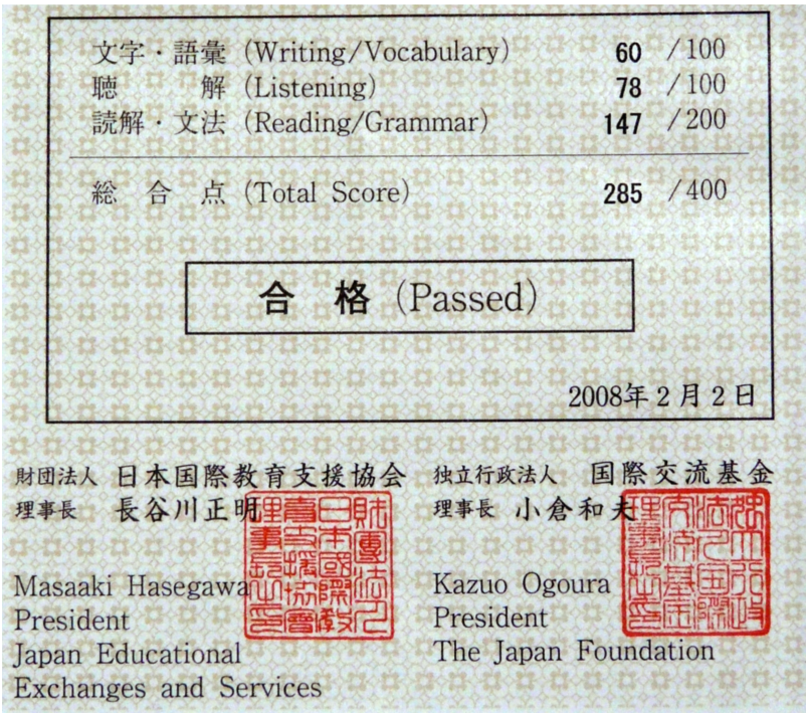
Section	Points	
Writing and vocabulary	100	(25%)
Listening	100	(25%)
Reading and grammar	200	(50%)
Total	**400**	**(100%)**

Table 6.2: Sections of a JLPT exam and their relative weighting. From Japan Foundation (2009).

guessed 25% of the time. However, the standard correction for guessing is not applied to final test scores, but instead a higher pass mark is used to compensate. We note that with four options per question, a learner who knew the correct answer to half the questions and guessed randomly the remainder should score 62.5% on average. 62.5% is thus the theoretically appropriate break-even pass mark for such a test, rather than 50%. This figure is approximated for the JLPT by a 60% pass threshold for levels 4, 3 and 2. For level 1, the pass mark of 70% more than sufficiently corrects for guessing.

It is a criterion-referenced test. Thus, rather than measuring an individual against their peers (*norm-referenced assessment*) or against past performance (*ipsative assessment*), it assesses a learner against fixed proficiency criteria (Begg 1997:22). This means that pass rates for each level vary from year to year according to the proficiency of candidates, and also that an outcome of the JLPT is a certification of proficiency.

Several aspects of the JLPT make it suitable as a reference test for our purposes. JLPT exams are based entirely on multiple-choice questions. Such questions can be easily and objectively marked, although in principle could be constructed so as to be arbitrarily complex. In practice, the Writing and Vocabulary and Reading and Grammar sections both contain question types which plausibly be automatically generated. Together these sec-



文字・語彙 (Writing/Vocabulary)	60	/100
聴　　解 (Listening)	78	/100
読解・文法 (Reading/Grammar)	147	/200
総合点 (Total Score)	285	/400

合格 (Passed)

2008年2月2日

財団法人 日本国際教育支援協会
理事長 長谷川正明

Masaaki Hasegawa
President
Japan Educational
Exchanges and Services

独立行政法人 国際交流基金
理事長 小倉和夫

Kazuo Ogoura
President
The Japan Foundation

Figure 6.2: An example results card from the JLPT 2007 tests.

tions contribute 75% of the test, suggesting that with reasonable coverage of their question types we should be able to estimate performance on the JLPT tests themselves, and through this measure proficiency. Although in this thesis we can only automate a limited number of question types, we nonetheless see opportunity for extending this coverage in the future. Finally, the JLPT provides a cohesive audience for a Japanese testing system: roughly 560000 candidates undertook one of the JLPT levels in 2008, and for many of these candidates significant study is required beforehand. By targeting these users we aimed to collect significantly more data than we would otherwise have been able to.

Question types

Each year, a new JLPT test is written for each of the four levels, and then administered to a large number of candidates. Since the aspects of language proficiency being tested are limited, and each question is multiple-choice, the number of question types used to test each aspect of proficiency is ultimately limited. It is beyond the scope of this thesis to perform a full account of these question types, but instead we focus on those which we believe best test vocabulary knowledge directly. In our view, the most basic aspects of word knowledge are the links between a word's form (or *surface*), its meaning (or *gloss*) and its pronunciation (or *reading*), as given in Figure 6.3.

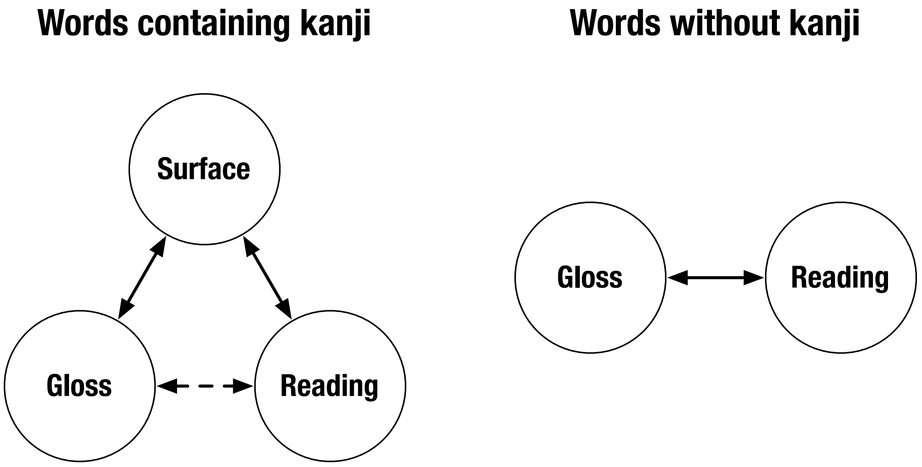
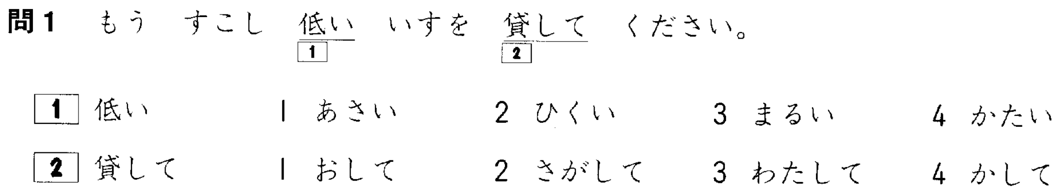


Figure 6.3: Easily testable aspects of word knowledge for words containing kanji, and words without kanji. A dotted line indicates a more tenuous link.

Many vocabulary-related questions in the JLPT interrogate these links. For example, Figure 6.4 asks the learner to select the correct pronunciation for each word in the sentence, thus testing the surface-to-reading link for these words. The same link is tested in reverse direction in Figure 6.5. Both questions make use of expert knowledge on behalf of test authors, using their experience to choose distractors for each question based on plausible misreadings firstly and plausible misrecognition secondly. Yet these are exactly the forms of error models we have for Japanese, and which formed the basis of our improved dictionary lookup in Chapter 5.

Since the JLPT is administered in Japanese, and is designed to accommodate participants from a variety of first-language backgrounds, the links to L1 meaning are not tested explicitly in the JLPT. However, since we choose a restricted audience for Kanji Tester – that of English speakers – we are also able to generate questions which interrogate the links to meaning.

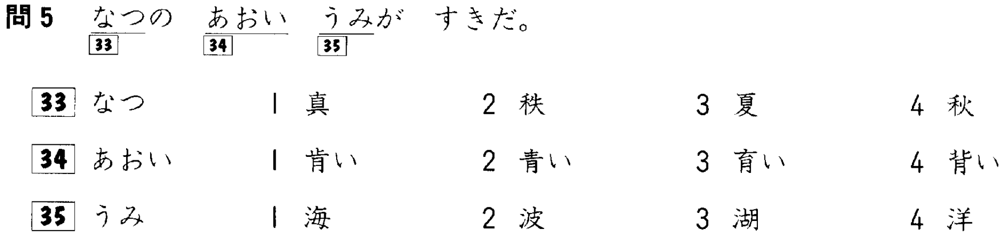
There are many other question types which might plausibly be automatically generated.



問１　もう　すこし　低い　いすを　貸して　ください。
　　　　　　　　　1　　　　　　2

1 低い	1 あさい	2 ひくい	3 まるい	4 かたい
2 貸して	1 おして	2 さがして	3 わたして	4 かして

Figure 6.4: Example question: the learner must select the correct reading for each kanji word in context. Taken from the JLPT level 3 example questions, Writing and Vocabulary section.



問５　なつの　あおい　うみが　すきだ。
　　　33　　34　　35

33 なつ	1 真	2 秩	3 夏	4 秋
34 あおい	1 肯い	2 青い	3 育い	4 背い
35 うみ	1 海	2 波	3 湖	4 洋

Figure 6.5: Example question: the learner must select the correct kanji form for words given by their reading. Taken from the JLPT level 3 example questions, Writing and Vocabulary section.

For example, questions testing grammatical patterns might be generated according to manually created templates akin to Zock and Afantenos's (2007) pattern drills, and similarly for questions focused on correct and appropriate verb conjugation. Likewise, questions based on reading comprehension might be constructed in the following way. Firstly, appropriate texts could be sourced online, for example news articles filtered by difficulty using Sato *et al.*'s (2008) measure. Semantic entailment or automatic summarisation techniques could then be used to generate sentences entailed by the text, to which the learner could give a true or false answer. Likewise, variations using words from the text but which are not entailed could be added as incorrect cases.

Despite the plausibility of these additional question types, they require significant additional labour and expertise in order to automate their generation. Since our scope is limited, we focus instead on the simpler vocabulary-based questions which might benefit most from the error models used earlier in this thesis. The following section describes how the Kanji Tester system presents and implements automatic test generation based on these simple question types.

Potential criticisms of the JLPT

We have described an abstract conception of an ideal test for language learners, and followed on to describe the most frequently used family of proficiency tests for non-native speakers of Japanese, the JLPT levels, which we use as a reference point for our test generation system. However, there remains a gap also between the JLPT levels and the ideal test as we have characterised it. This section focuses on this gap, discussing potential criticisms of the JLPT and our use of it as a reference point.

The attributes of the ideal test include a broad scope, i.e. a goal of determining holistic proficiency in a language. The JLPT does not provide a single test like this, but rather it provides several tests, each of which assesses participants against criteria for a certain level of proficiency. The choice of what levels to accredit and thus how many tests to run is somewhat arbitrary at all levels but Level 1, which is roughly pegged at the level of a native speaker having completed Japanese secondary school education. These arbitrary levels could arguably be replaced by a single test, calibrated in such a way as to provide reliable proficiency scores for a large variety of candidates of differing ability.

The main arguments against such a large test are common to paper-based tests. Firstly, it could be argued that a single test would have to be too long to accurately measure a wide variety of different ability levels, and would contain many questions which would either be far too easy or far too hard for a learner. However, computer-based adaptive testing would allow short tests for learners of a wide variety of levels. We note that the Test of English as a Foreign Language (TOEFL) is a single monolithic test, however it is adaptive in the listening and structure sections (Chalhoub–Deville and Deville 1999:287). Whilst we expect all major proficiency tests to move to this approach in the future, we nonetheless note that the JLPT is successfully in use as a proficiency test. The separation into four levels of testing is useful for us, since it allows us to focus on the more limited levels 3 and 4 in Kanji Tester, rather than having to construct a full test system for assessing native-like Japanese proficiency.

The JLPT can also be criticised for measuring only receptive aspects of proficiency, according to the receptive/productive distinction we introduced in Section 3.4. This largely relates to the use of multiple-choice questions for vocabulary testing, since providing a missing word or pronunciation is harder than choosing from it amongst distractors. In particular, Laufer and Goldstein (2004) argue that this difficulty relates to the graded nature of vocabulary knowledge; by not evaluating productive knowledge directly, the JLPT reduces the depth at which it measures word knowledge. However, we noted in Section 3.4 that breadth and depth of knowledge are correlated, suggesting that objections based on depth of testing may be overcome simply by testing a greater number of words. We further note that multiple-choice questions reduce marker bias, as opposed to productive questions which require the learner to fill in the blank. The use of multiple-choice only makes the JLPT questions better candidates for automation in Kanji Tester.

More broadly though, the form of the test prevents the direct measurement of productive oral ability. This means that, for example, an individual with excellent general vocabulary, grammatical and listening skills but extremely poor pronunciation might fare well on the test, but poorly in the real-world interactions the test aims to mimic. This criticism is common to many proficiency tests, and the merits of including interactive oral testing remain up for debate. Here, we merely hypothesise that for the average learner, their productive oral skills may be well estimated by a combination of vocabulary, grammar and listening skills which the test does cover. Even without such oral testing, the current cost of test-building reduces availability of tests to learners. Introducing more costly oral testing is a trade-off against other quality attributes of tests. In any case, the lack of interactive oral testing makes the goal of automatic test generation more achievable in the medium term.

As this section has shown, most shortcomings of the JLPT are also benefits for attempts to automate JLPT-like test construction. The JLPT thus provides a meaningful yet strong goal to aim for in test generation.

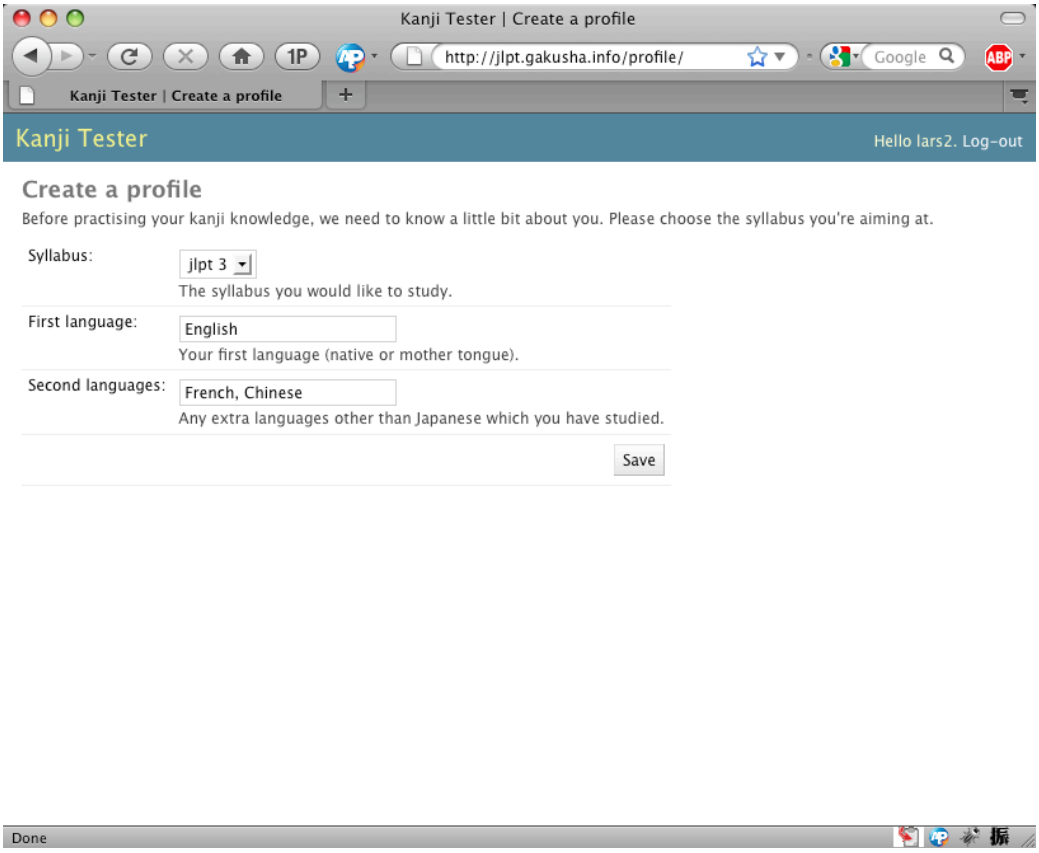
6.3 Kanji Tester

Having described in detail the JLPT examinations from which we draw our syllabi and question types, we now discuss our system for automatically generating tests for users, Kanji Tester. We do this in two parts. Firstly we walk through the system from a user's perspective, describing the system as they use it. Secondly, we describe it from our perspective, looking at its internal architecture and the mechanics of question generation and user modelling.

User interface

When a user accesses Kanji Tester for the first time, they must sign up for an account. This provides us with the means to customise tests to individual users. As part of the sign-up process, the user must create a profile, as shown in Figure 6.6. This includes choosing a syllabus to study, which at the time of writing was either JLPT level 3 or 4. They must also give their first language, and may optionally list any other languages they have studied. The user is then presented with a dashboard encouraging them to take a test. They choose the length of the test, from 10 questions to 50 questions, and click "Take a test". Kanji Tester then generates a new test for them based on their syllabus, as shown in Figure 6.7.

Each question in the generated test (a test *item*) is multiple choice, where the user must select the single correct answer amongst distractors, and is based on a single word or kanji from the syllabus. Although there are many aspects of word knowledge, Section 3.1 discussed Vermeer's (2001) work linking breadth and depth, indicating that well constructed



Kanji Tester | Create a profile
1P
http://jlpt.gakusha.info/profile/
Google
ABP
Kanji Tester | Create a profile
+
Kanji Tester
Hello lars2. Log-out

Create a profile

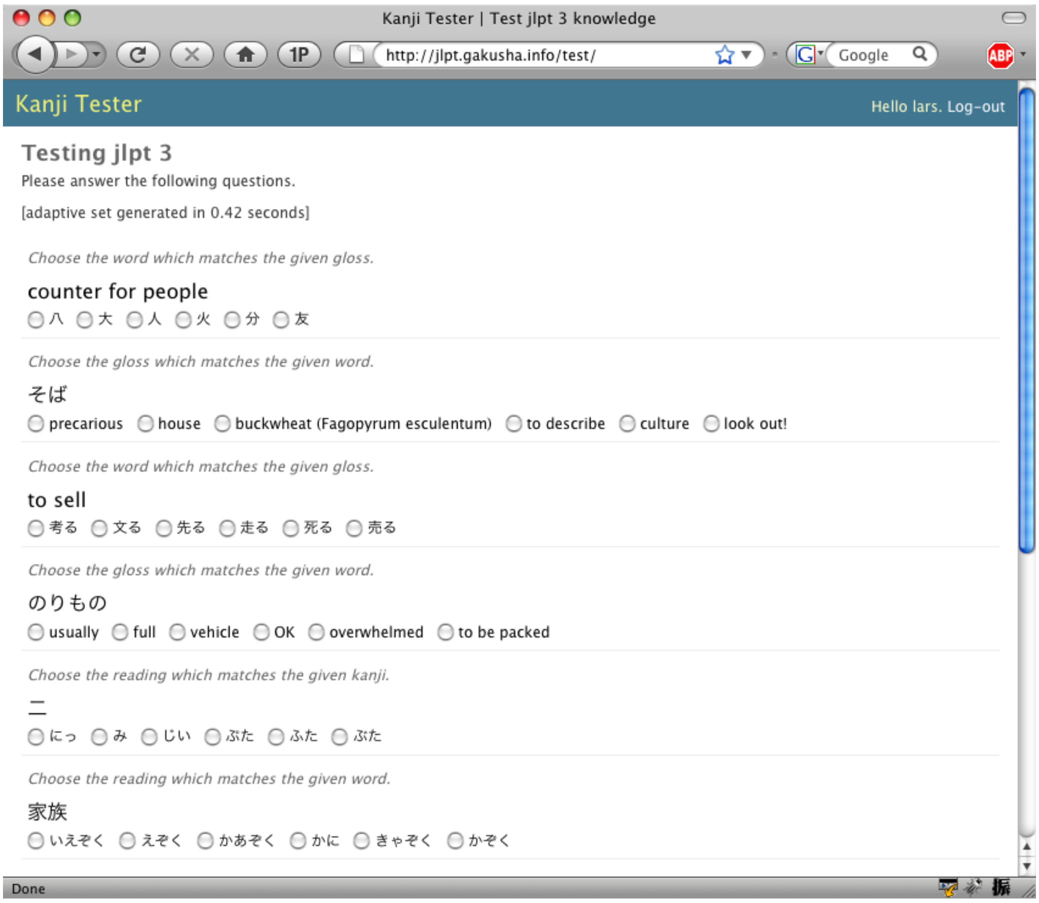
Before practising your kanji knowledge, we need to know a little bit about you. Please choose the syllabus you're aiming at.

Syllabus: jlpt 3
The syllabus you would like to study.

First language: English
Your first language (native or mother tongue).

Second languages: French, Chinese
Any extra languages other than Japanese which you have studied.

Figure 6.6: Creating a profile for Kanji Tester. The user enters a first language, and optionally any second languages they have studied.


Kanji Tester | Test jlpt 3 knowledge
http://jlpt.gakusha.info/test/
Google
Done

Kanji Tester

Hello lars. Log-out

Testing jlpt 3

Please answer the following questions.

[adaptive set generated in 0.42 seconds]

Choose the word which matches the given gloss.

counter for people

○八 ○大 ○人 ○火 ○分 ○友

Choose the gloss which matches the given word.

そば

○ precarious ○ house ○ buckwheat (Fagopyrum esculentum) ○ to describe ○ culture ○ look out!

Choose the word which matches the given gloss.

to sell

○考る ○文る ○先る ○走る ○死る ○売る

Choose the gloss which matches the given word.

のりもの

○ usually ○ full ○ vehicle ○ OK ○ overwhelmed ○ to be packed

Choose the reading which matches the given kanji.

二

○にっ ○み ○じい ○ぶた ○ふた ○ぶた

Choose the reading which matches the given word.

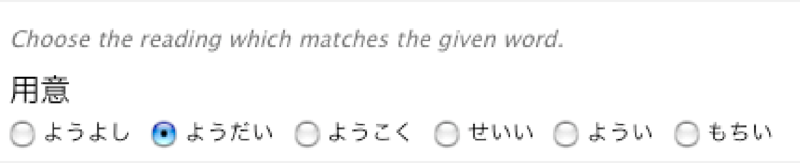
家族

○いえぞく ○えぞく ○かあぞく ○かに ○きゃぞく ○かぞく

Figure 6.7: A sample test generated by Kanji Tester for JLPT 3.

breadth testing was sufficient to measure proficiency. For this reason, only a few basic but crucial aspects of word knowledge are tested.

Figure 6.3 shows the types of links available between the three forms of word knowledge, each of which represents an aspect of knowledge which can be tested, and each of which generates a different question type depending on the direction the link is traversed. For example, surface-to-reading yields a question where the user is presented the surface as stimulus, and asked to identify the correct reading (Figure 6.8); gloss-to-surface would use the gloss as stimulus, and ask the user to identify the correct surface. In the case of words without kanji, the reading and the surface components are the same, since kana are syllabic; this reduces the available links to only the gloss-reading link.



Choose the reading which matches the given word.

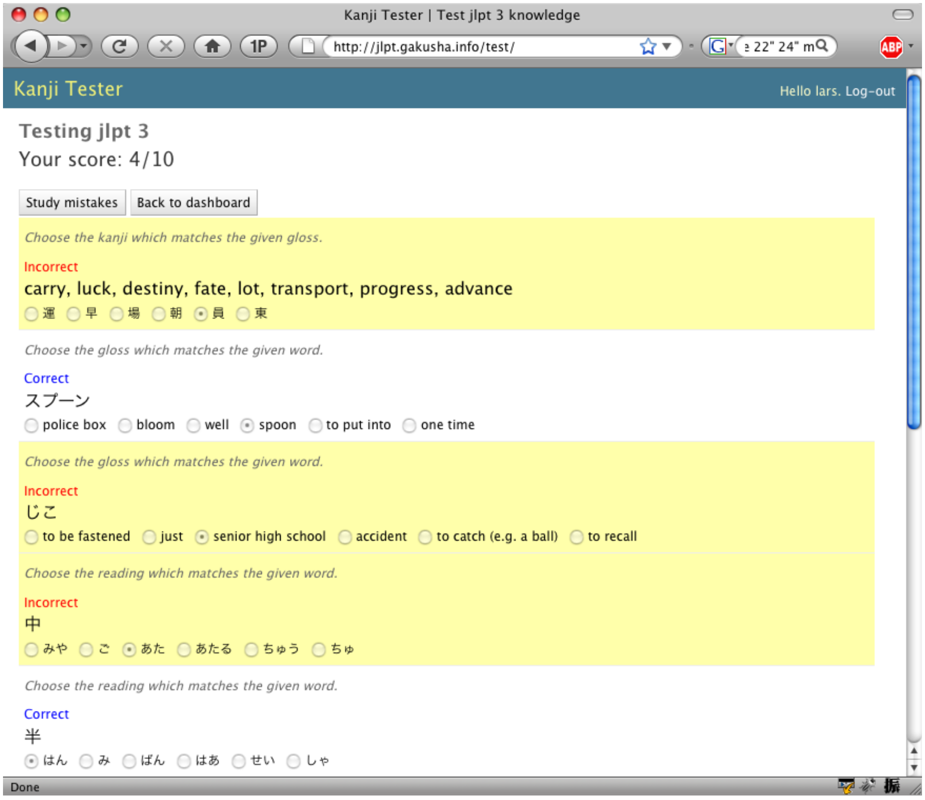
用意

○ ようよし ● ようだい ○ ようこく ○ せいい ○ ようい ○ もちい

Figure 6.8: An example question measuring the surface-to-reading link for the word 用意 *yōi* “preparation”.

The user selects their answers to each question in the test and then clicks “Check answers”. They are then provided with feedback on their test performance, as shown in Figure 6.9, which lists their score and highlights mistakes. If the user clicks on “Study mistakes”, they are provided with a vocabulary list of any word or kanji they made a mistake on, complete with multiple readings and glosses. This study list is tied to the user’s syllabus in two ways, firstly in the sense that it can only contain words and kanji which the user was tested on and are hence in their syllabus, and secondly in the sense that only aspects of word and kanji knowledge within the chosen syllabus are shown in the study view. In particular, a word with multiple pronunciations and kanji forms may have some of these excluded from the study view (and from testing) if they are not part of the syllabus. Ideally, word senses would be similarly restricted, however the JLPT syllabi do not specify words or kanji to this level.

Once the user returns to the dashboard, they are presented with two main statistics on their performance: their cumulative accuracy, and their accuracy on the most recent



Kanji Tester | Test jlpt 3 knowledge
http://jlpt.gakusha.info/test/
Kanji Tester
Hello lars. Log-out
Done

Testing jlpt 3

Your score: 4/10

Study mistakes | Back to dashboard

Choose the kanji which matches the given gloss.

Incorrect

carry, luck, destiny, fate, lot, transport, progress, advance

○ 運 ○ 早 ○ 場 ○ 朝 ● 員 ○ 東

Choose the gloss which matches the given word.

Correct

スプーン

○ police box ○ bloom ○ well ● spoon ○ to put into ○ one time

Choose the gloss which matches the given word.

Incorrect

じこ

○ to be fastened ○ just ● senior high school ○ accident ○ to catch (e.g. a ball) ○ to recall

Choose the reading which matches the given word.

Incorrect

中

○ みや ○ ご ● あた ○ あたる ○ ちゅう ○ ちゅ

Choose the reading which matches the given word.

Correct

半

● はん ○ み ○ ばん ○ はあ ○ せい ○ しゃ

Figure 6.9: The feedback given to a user immediately upon completing a test.

test. Ideally, if their performance is improving through study, their most recent test score should be higher than their cumulative average, increasing motivation for the user. Their accuracy on these tests is designed to itself be a proficiency estimate of receptive vocabulary knowledge. Cumulative accuracy is thus the best estimate of the user's average proficiency over the entire period they have used the system, true to their ability but slow to respond to changes. The last test taken provides a dynamic estimate which responds quickly to changes in proficiency but which is far noisier.

Note that in our heavy use of multiple-choice questions, we do not allow users to omit answers and neither do we use the standard correction for guessing in the user results, which takes the form of a penalty for incorrect answers. This approach was taken for several reasons. Firstly we note that no such correction is taken on the JLPT, which we emulate, although participants may omit answers. Secondly, the standard correction formula[2] assumes that in cases where the learner doesn't know the answer, the learner guesses randomly amongst the available options. This assumption is clearly false in cases of partial knowledge, and the modelling is complex to determine the appropriate penalty for such cases (Espinosa and Gardeazabal 2007). Thirdly, using a penalty and allowing users to omit answers would reduce guessing, and whilst this would decrease the number of explicit errors (as opposed to omitted answers), these errors are necessary to feed our adaptive error models. Finally, such corrections are typically aimed at determining the participant's ability in terms of their "true score" on a given test; we instead use far more fine-grained user modelling to assess a users's proficiency, making correction unnecessary from an ability estimation perspective.

User modelling

Kanji Tester shares many aspects of Intelligent Language Tutoring Systems, such as Heift and Nicholson's (2001) German Tutor, with the main difference residing in Kanji Tester's restricted focus on vocabulary rather than grammar. A large similarity though resides in the reliance of such systems on an accurate user model to base user interaction on. A range of user characteristics can be modelled, including goals, knowledge, background,

[2] The standard correction for guessing estimates a user's corrected score s on a test with k options as $s = r - \frac{w}{k-1}$, where r is the number of correct responses and w is the number of incorrect responses. This correction is effectively a penalty of $\frac{1}{k-1}$ for each incorrect answer.

preferences, interests, individual traits and environment (Brusilovsky 2001). Kanji Tester has two of these key elements available to it, namely the user's first and second language background, and their vocabulary knowledge, as displayed in their test performance.

Each question a user answers is based on a particular item in the syllabus, a word or a kanji, and requires distractors to be generated as plausible alternatives to the the correct answer. The plausibility of the distractors is crucial; if users can systematically guess the correct answer without knowledge of the item in question, the test will have no validity and will not be able to measure proficiency. Choosing an appropriate user model is thus crucial to generating useful distractors.

In order to construct distractors for each question with similar credibility to what a teacher would select, we make heavy use of both the plausible misreading and the plausible misrecognition employed in our rebuilt version of FOKS. However, with a continuous influx of user responses we have the potential to improve these models, making them respond and adapt to user mistakes. In particular, the system can increase its difficulty by increasing the frequency of distractors or error types which users have made mistakes on. To do this we need to perform at least limited modelling of user knowledge and behaviour.

There are several layers of granularity at which such user modelling could potentially occur, both in terms of errors and in terms of users, as shown in Figure 6.10. Users could be modelled: (a) globally, thus adapting to the user population as a whole; (b) in groups, for example by first language background; or (c) individually, adapting to each user. The larger the user grouping, the greater the benefit to users who are representative of that group, and the greater the penalty to minorities and individuals with error profiles which differ from their group. If users are considered individually, data sparseness becomes an issue, and the level of adaptivity may be limited.

A good example is provided by vowel length errors. Since English has no vowel length distinction, native speakers of English may be less sensitive to vowel length in Japanese, and thus make more errors. A global shared model would adapt to the background of the majority of the learner base, in our case English speakers, thus penalising learners from other backgrounds who might not make these errors. If our model is based instead on language background, then other language groups are not penalised, but individual native English speakers who are attuned to vowel length differences still are. If we model errors for each

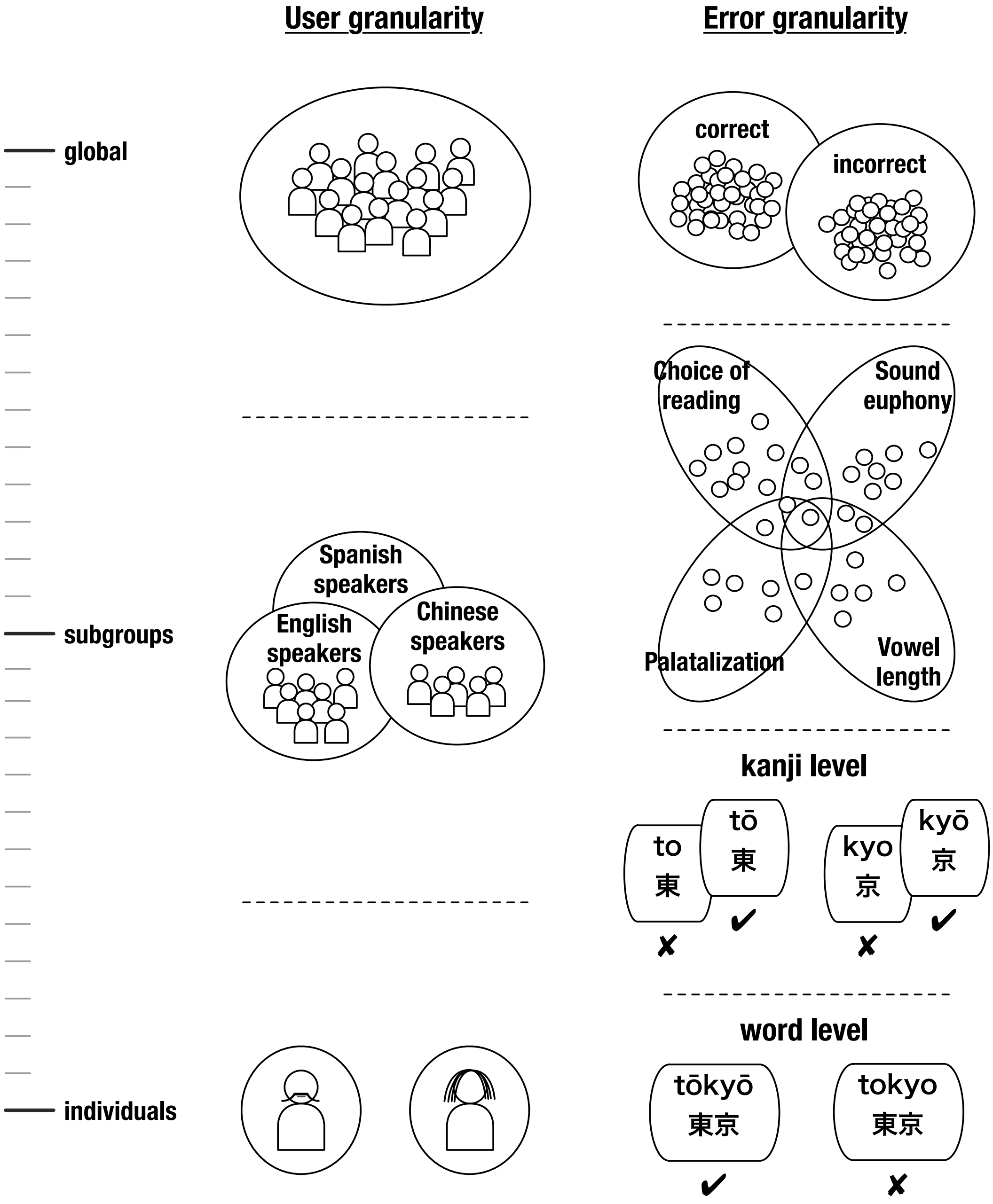


Figure 6.10: Examples of user and error models of varying granularity considered for use by Kanji Tester. On the left, user models can be maintained globally (treating all users identically), in aggregate by some common attribute (e.g. first language), or individually. Similarly error prevalence can be modelled globally, by broad error type, by kanji or by word. The actual configuration of Kanji Tester as reported models users as individuals and errors at the kanji level.

user individually, no user is penalised by the behaviour another, but they cannot benefit from other users either. For Kanji Tester, we chose this last form of error model, as advocated by Rich (1983), because out of our three main options it makes the least assumptions about each individual user's behaviour.

For any level of user model, there are four possible granularities of error models which Kanji Tester could potentially use. Suppose the user chooses the incorrect reading *tokyo* for the word 東京 *tōkyō* "Tokyo". The most coarse model simply registers this as an error, without caring about the error type or the word involved, and notes that the learner is now more likely to be wrong again in the future. A slightly finer grained model would register the error as one of *incorrect vowel length*, but would still ignore the details of the word involved. It would note that the learner is more likely to make vowel length errors in the future. A more detailed model at the kanji level would register that the learner had in fact made two errors, misreading 東 *tō* as *to* and 京 *kyō* as *kyo*, and would estimate the likelihood of the learner misreading either kanji this way as increased. The most fine-grained model would consider the error as a specific to the word 東京, and increase the likelihood of this misreading for this word only.

The main question here is the extent to which error occurrences can be generalised into useful broader trends. In Kanji Tester, we model error trends at the kanji level and use these per-kanji models in sequence when kanji occur in sequence as words. This is convenient since error models from FOKS are applied at the per-kanji level, and thus transfer directly.

Error models from FOKS take the form of probability distributions. For a word containing kanji, with r as its guessed reading, k as its kanji form, and k' as a plausible misrecognition, the models specify $\Pr(r|k)$ and $\Pr(k'|k)$. For the misrecognition model, we used the same error model we used for FOKS, incorporating stroke edit distance as the similarity measure. In Kanji Tester, we simply use a static snapshot of these models as prior distributions, and maintain for each user u their own dynamic posterior distributions $\Pr_u(r|k, R)$ and $\Pr_u(k'|k, R)$, where R is the ordered sequence of their previous responses. Note that for simplicity this snapshot removes any distinctions between error types, and simply considers them as error candidates with different likelihoods. This is aimed at providing a reasonable trade-off between data sparseness and adaptability for each individual user. Thus, in the earlier case of 東 *tō* being misread as *to*, our kanji-level reading model would only note it as an incorrect reading rather than as a vowel length error. The user model would then update itself so as to register the likelihood of 東 being misread this way in the future, rather than, say, modelling vowel length errors in general as increasing in likelihood for this user.

The following two sections describe in more detail firstly how our user models are used to generate questions, and secondly our update algorithm for refining our error models from user responses.

Generating questions

In order to better describe how Kanji Tester generates test questions, a brief overview of its architecture is useful. Kanji Tester provides an online web-based user interface from which users can choose a syllabus and take tests of 10, 20 or 50 questions in length. Each test is generated on a question-by-question basis, using three key components of Kanji Tester: its learner syllabi, its user models and its question plugins. These correspond to the dotted boxes in Figure 6.11's architectural overview, and are designed to be extensible. Each of these three components has a crucial role in question generation.

Each syllabus supported by Kanji Tester is prepared for use in the form of a paired word and kanji list. Each word on the word list is matched to a word from a reference lexicon – in our case JMdict – in a semi-automatic fashion. Once a user has chosen a syllabus to study, every question generated for them is seeded by a word or kanji from that syllabus's list. This is the first step in Kanji Tester's question generation algorithm Figure 6.12. The next choice in question construction is what type of question to generate. Individual questions are generated by question plugins, and for each test only a single plugin can generate each question type.

Tests silently alternate between two types, *control* and *adaptive*. This mirrors the two types of question plugins available, *simple* and *adaptive*. Simple plugins represent a baseline effort; their question generation methods are fundamentally similar to adaptive plugins, but are static and unchanging. On the other hand, adaptive plugins make use of user models which they refine with each user response. Control tests are limited to using only simple plugins, whereas adaptive tests make use of adaptive plugins wherever possible.[3] The con-

[3] The main exception to this the "Random gloss" plugin, which is a simple question plugin used in both

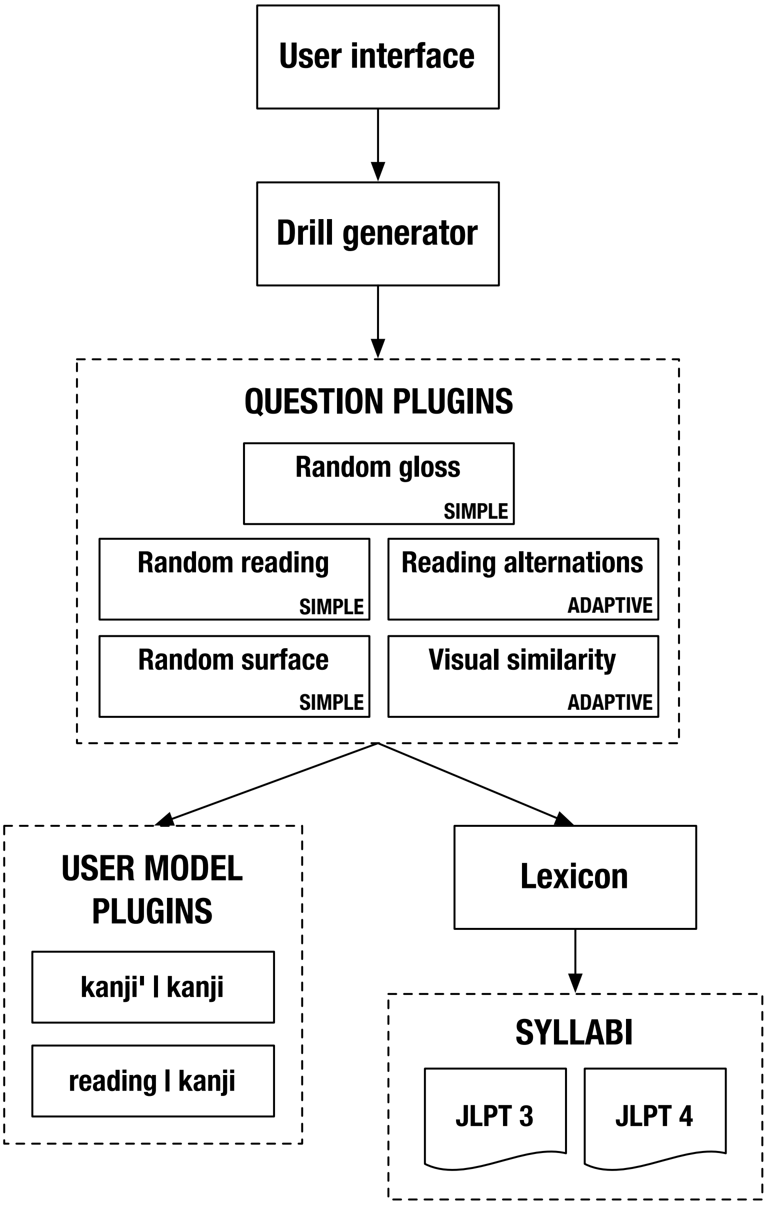


Figure 6.11: The high-level architecture of Kanji Tester. Dotted lines indicate areas designed for easy extension.

1. Randomly select a seed item from the user's syllabus.
2. Randomly select a question plugin, of the available plugins which are capable of using the given seed item.
3. Within the question plugin:
    - 3.1. Randomly select a question type from those the plugin can generate.
    - 3.2. Generate a pool of valid distractors, optionally using one of the available user models.
    - 3.3a. [Simple plugins] Randomly sample distractors from the pool of available distractors with *uniform probability*.
    - 3.3b. [Adaptive plugins] Randomly sample distractors from the pool of available distractors with *proportional probability*.
4. Render the resulting question to the user.
5. [Adaptive plugins] Update user models with the user's response.

Figure 6.12: The algorithm used by Kanji Tester to generate each test question.

trol/adaptive distinction is designed to allow the comparison of the two methods, with the expectation that adaptive tests should be more difficult than their control counterparts.

Regardless of whether the test is a control test or an adaptive one, a fixed set of question plugins is then available to generate questions with. If the seed item is a word containing kanji, any plugin from that set can be used. If the item instead uses no kanji, then only the simple gloss plugin can be used. If several plugins are available, one is chosen randomly from amongst them. This corresponds to step 2 in Figure 6.12. If the chosen plugin can generate more than one type of question, it then chooses randomly between them, and for the final question type it generates a pool of distractors. From this pool, five are randomly chosen to be presented alongside the correct answer. In simple plugins, distractors are chosen with uniform probability; in control plugins, they are chosen with probability equal to their estimated likelihood of eliciting an error. The question is then presented to the user, and their response recorded. If the plugin was adaptive, the user's response is finally fed back into the user model, which updates itself according to the update rule we discuss.

Update algorithm

The basic goal of our approach to testing is to attempt to maximise the number of genuine errors we provoke from the user. By making them aware of gaps in their knowledge, we provide them with the opportunity to correct these gaps. We are aware that difficult tests can reduce study effort (Marso 1969), an effect reinforced by more recent work by Ponsoda *et al.* (1999) on manipulating the difficulty of computer-adaptive tests, and this suggests that we may wish the tests to be easier for learners. However, multiple choice test difficulty can always be reduced by using less plausible distractors. On the other hand, finding *more plausible* distractors to increase test difficulty for the same target words is far more cumbersome. We take the reasonable approach of simply maximising difficulty (and hence user errors), knowing that even with good distractors our tests may err on the easy side.

In order to do this, we model each user's knowledge and discriminative ability using the information we have available: our prior expectation of what common errors will be made, and the user's actual responses to previous questions. This section is concerned with making

control and adaptive tests, since no adaptive counterpart exists.

use of the latter information.

Our update algorithm takes an error model, the question it generated and the user's response to that question, and uses these elements to construct a new updated error model that better fits the learner's observed behaviour. More intuitively, it should ensure that distractors which successfully confused a user are presented more frequently than distractors which did not. There is a large space of algorithms which could perform this function. The main trade-off any such algorithm faces has to do with the rate of adaptivity. If the algorithm adapts swiftly to user input, it is also more likely to over-fit noise; if it adapts too slowly, the strength of the model will rely entirely on its priors rather than on per-user information.

Our user-level granularity tells us what error model to update with each user response, and our error-level granularity gives us a concrete probability distribution. We now describe the update algorithm employed in Kanji Tester, which compares each user response to the probability distribution(s) which generated it and updates the distribution(s) so that the response is considered to be more likely to occur in the future.

We start with basic definitions:

$$\begin{aligned} w &= \text{the word the question is based on, } w = k_1 \dots k_n \\ O &= \text{the distractor space for the word, } O = O_1 \times \cdots \times O_n \\ D &= \text{the options shown to the user, } D \subset O \\ c &= \text{the user's chosen answer, } c \in D \end{aligned}$$

We displayed a random subset D of possible options O to the user, and they chose option c as their answer. According to our user model, each $d \in D$ had its probability calculated firstly by:

$$\begin{aligned} \Pr(d|w) &= \Pr(d_1 \dots d_n | k_1 \dots k_n) \\ &\approx \prod_{i=1}^{n} \Pr(d_i|k_i) \end{aligned} \tag{6.1}$$

and then normalised $\forall d \in D$ to get $\Pr(d|D)$. In this way, we already know the value $\Pr(c|D)$. We now wish to determine the posterior distribution $\Pr'$ for $(C|D)$. We base our update rule on the constraint:

$$\forall_{\{d: d \in D \setminus \{c\}\}} \Pr{}'(c|D) \geq \Pr{}'(d|D) + \epsilon \tag{6.2}$$

That is, the user chose c because it was better than any other option by a margin of ϵ. If we assume that the user did not guess at random, but chose their answer because it was the most likely one to be correct out of the options available, then this rule simply ensures that in the posterior distribution their answer becomes the most likely out of the given options. It does this by enforcing a margin of ϵ in the posterior distribution $(C|D)$ used in the next iteration. Note that ϵ serves to smooth the update of the error model, and the value chosen represents a particular tradeoff between quickly adapting even to noise and random guesses, or slowly adjusting to learner input. We used an ϵ of 0.2, which we intended as an intermediate value.

The use of an ϵ margin was also intended to indicate that not every question we ask the user yields new information. If the ϵ margin already exists between the user's choice and the other options, then the current model adequately predicted the user's response, and no change is needed. If the margin does not exist, we update the error model using the following steps:

1. Let $m = \max_{\{i:d_i \neq c\}} \Pr(d_i|D) + \epsilon$
2. Define the posterior distribution $(C|D)$ as follows:
   - $\Pr'(d_i|D) = \Pr(d_i|D)$ if $d_i \neq c$
   - $\Pr'(c|D) = \max\{\Pr(c|D), m\}$
   - Normalise $(C|D)$ such that $\sum_i \Pr'(d_i|D) = 1$

Now we know the posterior distribution of $(C|D)$, yet our error model is stored in terms of $(C_i|D_i)$. In other words, $\forall d \in D$, we know the sequence probability $\Pr'(d|D) = \Pr'(d_1 \ldots d_n|D)$, but still need to define $\Pr'(d_i|D_i)$. That is, we need to distribute the difference between the prior and posterior distribution for $(C|D)$ to each $(C_i|D_i)$. We do this by firstly defining the constant Δ:

$$\begin{aligned} \Delta &= \frac{\Pr'(d|D)}{\Pr(d|D)} \\ &\approx \prod_{i=1}^{n} \frac{\Pr'(d_i|D_i)}{\Pr(d_i|D_i)} \end{aligned} \tag{6.3}$$

We reach Equation 6.3 above by replacing $\Pr'(d|D)$ and $\Pr(d|D)$ with their sequence approximations from Equation 6.1. We are now close to being able to redistribute this probability mass. We add the additional approximation that the mass should be distributed evenly amongst sequence components:

$$\frac{\Pr'(d_i|D_i)}{\Pr(d_i|D_i)} = \frac{\Pr'(d_j|D_j)}{\Pr(d_j|D_j)} \quad \forall i, j \in 1 \ldots n \tag{6.4}$$

Combining this with Equation 6.3, our final update rule emerges:

$$\Pr'(d_i|D_i) = \Delta^{\frac{1}{n}} \Pr(d_i|D_i) \quad \forall i \tag{6.5}$$

This update rule has some desirable properties. For example, only distractor components displayed to the user have their likelihood changed. This is true at the kanji level, though the likelihood of unseen kanji sequences may still change. Suppose the user had to identify the word meaning "composition, writing". The correct answer is 作文, which they must identify amongst distractors generated by our (kanji$'$|kanji) misrecognition model. They make an error and choose 非人 as their answer. In response to their answer, probability mass is moved away from the other word-level (mis)recognition candidates displayed – including the correct answer – and distributed to (非人|作文). In turn, this additional probability mass is distributed by our update rule into increased likelihoods for pairs (非|作) and (人|文). This algorithm thus provides a principled method for updating user error distributions at the kanji level based on their answers to test questions.

In order to distribute probability mass to reading parts, each word in the syllabus containing kanji must be accurately GP-aligned to its reading. We used the unsupervised method described in Section 5.2 to automatically align each syllabus, and then manually corrected any alignment errors encountered. The only exception to our update rule lies in our (reading|kanji) user model, in the case of non-compositional readings. If the user must guess the reading of a word where the correct answer is non-compositional, as in the case of 山車 *dashi* "festival float", a correct answer from the user will have have no way of attributing the reading of the whole to the reading of its kanji parts, and thus no way to increase the likelihood of the correct reading in place of an incorrect reading. In this rare and special case, no update takes place. However, this is not problematic since the correct reading must always be amongst the available answers to a question.

6.4 Evaluation

In Section 6.1 we discussed our goal of replicating human constructed tests, and then discussed our method of using the error models from FOKS to generate plausible distractors for different question types. After deploying the Kanji Tester system in November 2008, roughly one month before the 2008 JLPT tests, we advertised it on several mailing lists and used a small Google AdWords campaign to try to gain users who would be studying for the test. Since Google trends indicated that the search term "JLPT" encountered heavy traffic in Singapore, we focused our limited AdWords budget on Singaporean users in an attempt to gain participants in a short timeframe. After the JLPT season was over, we analysed log data collected between November 2008 and February 2009 in an attempt to better understand this new system, how it was used, and the extent to which it helped users. This section discusses the details of this analysis.

User demographics

During the period under analysis, 225 users completed a minimum of one test, responding to 17065 questions in total (75.8 questions per user on average). In order to use the system, each user first had to enter their first and second language background. Table 6.3 shows that, although most users had English as their first language, we had users from 36 other language backgrounds, together accounting for 60% of the user population. Note that this includes 19 people of Japanese first language background, who may have been language educators experimenting with the system. For the remainder of this chapter we exclude these users from our analysis, since their responses are unlikely to be characteristic of typical learners.

Of the two syllabi available to users, 71% of users chose the more advanced JLPT level 3, whereas 29% of users chose JLPT level 4. As with any web-based system, participation rates were not even between users. Figure 6.13 gives the number of users that have taken at least n test, for increasing n. The majority of users completed less than ten tests in total. A

Language	**# Users**	
English	90	41.3%
German	28	11.9%
Japanese	19	8.7%
Marathi	12	5.5%
Indonesian	9	4.1%
Vietnamese	6	2.8%
Chinese	6	2.8%
Other (30 more)	50	22.9%
Total	219	100%

Table 6.3: Distribution of active users by first language.

small handful completed many more.

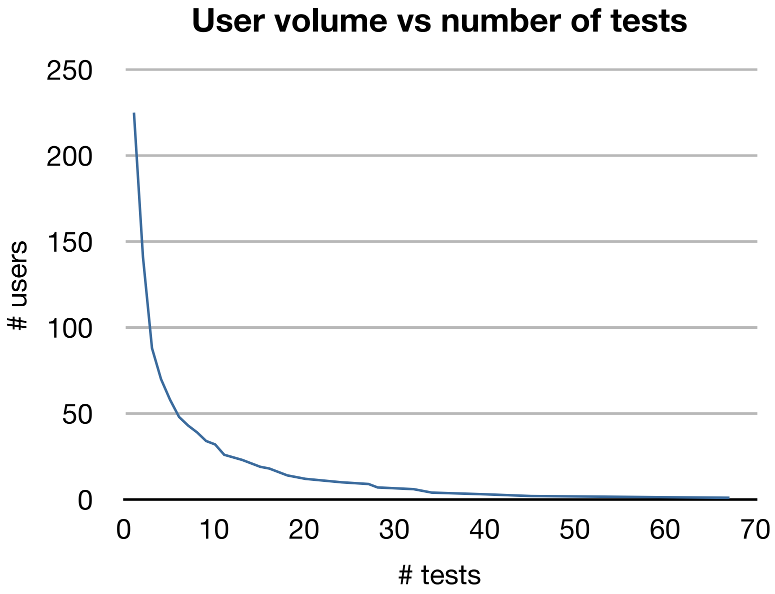


Figure 6.13: Number of users (reverse cumulative) set against number of tests.

Also of interest is how these users made use of the system. Did they test themselves repeatedly over a short space of time, or return to the site occasionally over longer periods to determine their progress? To measure this, we ordered the tests each user took sequentially by their timestamp, and measured the time between sequential tests. Figure 6.14 shows a reverse cumulative histogram of the mean time between tests for each user, with time on a

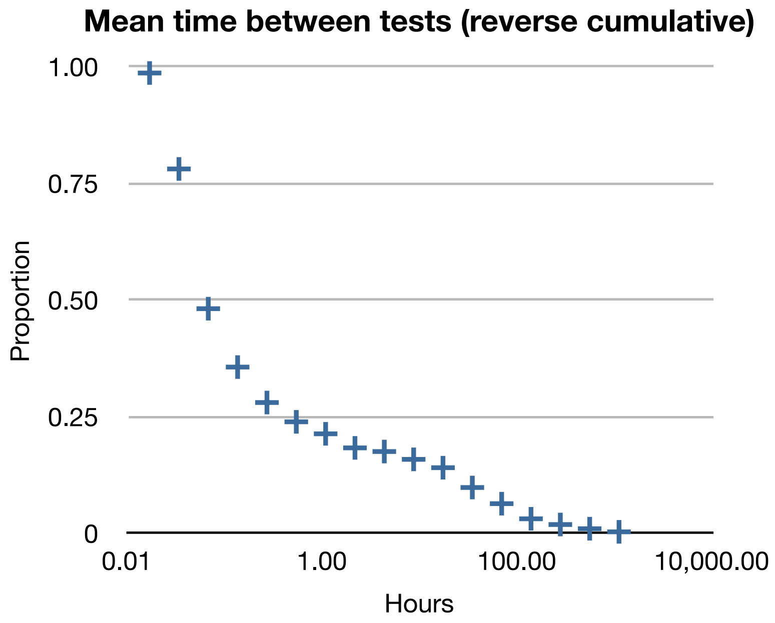


Figure 6.14: The proportion of times between tests taken sequentially by a user, expressed as the proportion of such times of magnitude $> x$ hours (reverse cumulative histogram).

log scale. It shows that the vast majority of tests are taken in quick succession to the previous test, within a few minutes. This suggests that many users used the system as a drill rather than a proficiency test, and thus that we will be unable to measure long-term progress for most users due to the short time frames.

User ability and improvement

Kanji Tester is designed to help learners self-evaluate their proficiency; for this reason, items for tests were chosen randomly from the user's chosen syllabus, as discussed in Section 6.1. Over the course of many such tests, Kanji Tester gains an increasingly accurate picture of a user's knowledge level. If we take our best estimate of their ability, i.e. their average accuracy across all responses, we get scores as shown in the histogram in Figure 6.15. The majority of users in both tests had mean accuracy greater than 0.75 over all tests taken. This figure is representative of their likely performance on the equivalent portion of the JLPT, namely multiple choice questions of this form. However, the JLPT is designed to test various facets of Japanese knowledge, so it is unlikely that we could accurately predict performance on the whole test from just receptive vocabulary knowledge.

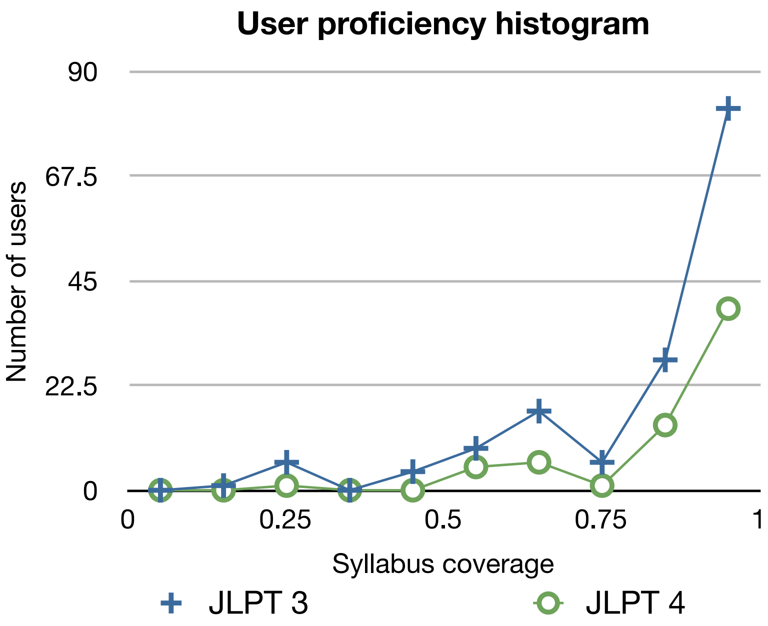


Figure 6.15: Mean score per user as a histogram.

We are of course interested in how this proficiency is related to usage of Kanji Tester, since the premise of our work is that better self-evaluation will allow users to make better decisions about their study patterns, ultimately improving their proficiency. If we compare our best estimate of ability against the number of tests each user takes, we get the results shown in Figure 6.16. Here, we see again that most users take few tests, and already score quite well. A linear trend line shows a very weak positive correlation between usage and proficiency estimates.

Beyond static estimates of user ability, user tests are also distributed in time, and thus can serve to measure change in ability over time. We now examine changes in ability in several ways. Firstly, we can use the number of tests taken as an estimate of study time, and compare study time and ability. Figure 6.17 takes this approach, and shows a very weak negative correlation between ability and study time as measured.

Although it is theoretically possible that increased study with poor study habits could reduce ability, a more likely interpretation of this graph is that one or both of our measures

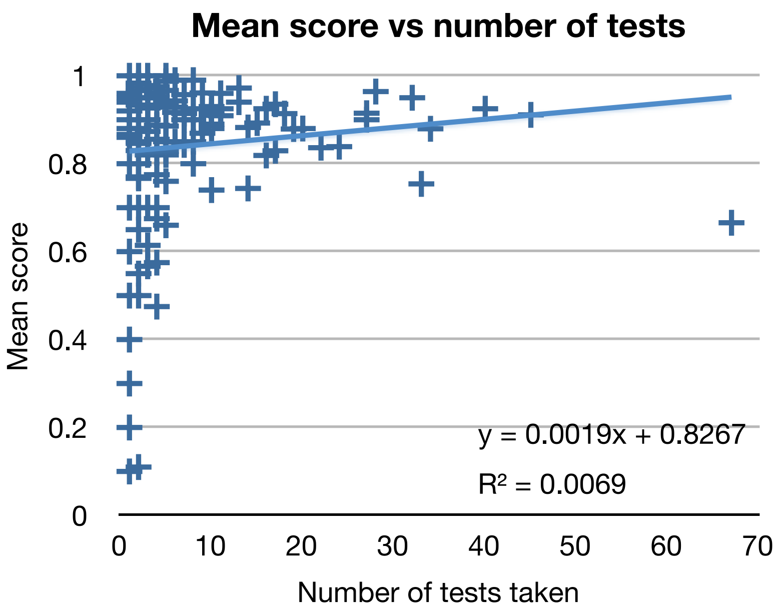


Figure 6.16: Ability, as measured by mean score for each user, plotted against usage, as measured by the total number of tests each user took.

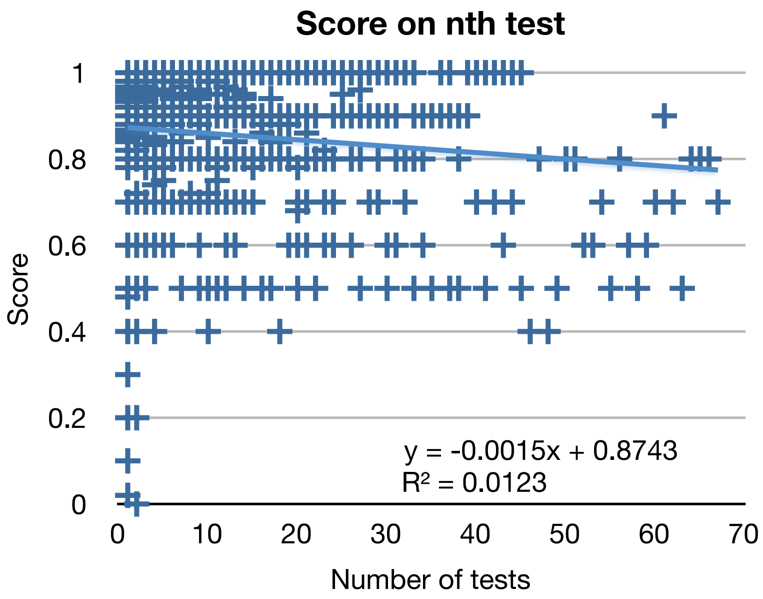


Figure 6.17: Ability, as measured by test score on the nth test, plotted against study time, as measured by number of tests taken n, for all users.

lack full validity. For example, we saw in Figure 6.14 that most tests are taken in rapid succession. If users lose interest or get tired (*user fatigue*), using test scores to measure ability could lose validity, since we may be partially measuring user attentiveness instead. Similarly, if tests are taken in rapid succession in a single session, the number of tests taken may not be a good measure of broader time spent studying.

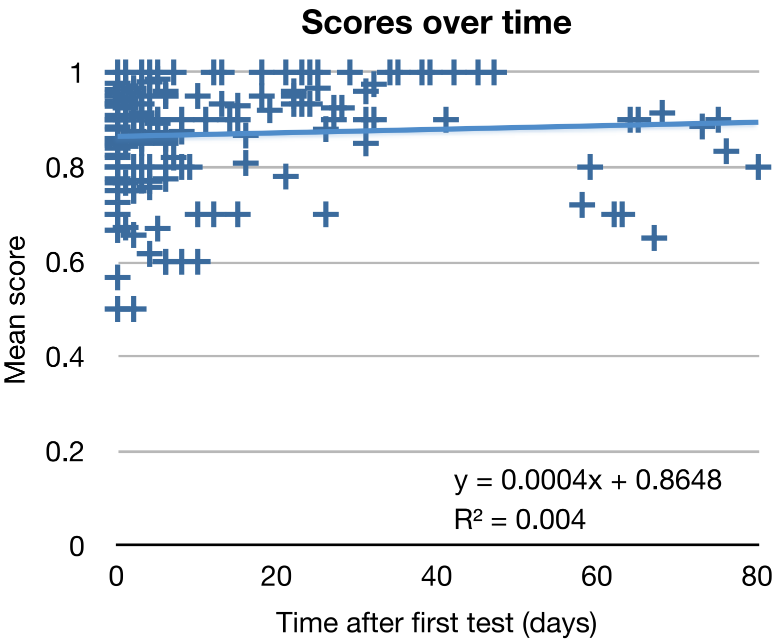


Figure 6.18: User scores plotted against time in days since their first test.

To address these concerns about measurement validity, we now plot user scores on tests against real time in days since their first test (Figure 6.18). Each point represents a user's mean score on a single day in which they took tests. The linear trend-line shows a very weak positive correlation between ability and time passed, indicating that users are in general improving in proficiency over time.

Compared to flashcard-style progression through a syllabus where users are continuously shown the same items until they are reliably learned, users of Kanji Tester encounter new items constantly, since items are chosen randomly from the syllabus. However, with sufficient tests, many items are encountered and tested several times. One way to determine if, general improvement aside, users are improving on the actual items they are tested on is to compare their responses to such items over time. Since data is quite sparse, we looked at only users who had encountered at least one item multiple times, and compared the mean score on their first encounter to such items to the mean score on the last encounter. The former should measure the knowledge they had before using the system; the latter represents any change.

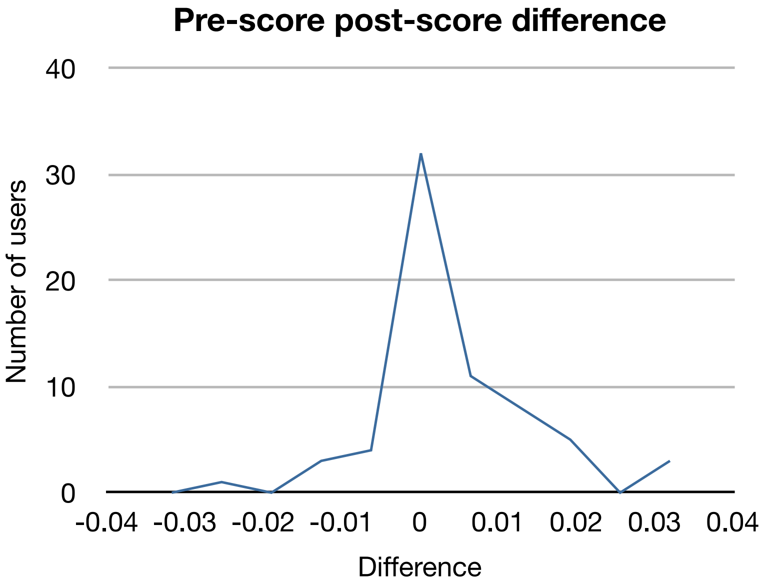


Figure 6.19: The difference in scores between the first time a stimulus was seen and the last time it was seen, expressed as a histogram.

The results of this analysis are shown in Figure 6.19. They show that on average there is a slight improvement in accuracy on stimulus tested more than once, although for many users there is no difference. This in turn suggests either that few users use the provided "study view" and carefully study their mistakes, or that the majority of stimuli encountered twice were those which users already knew.

To eliminate these potential effects, we modelled learner knowledge as a finite state machine, where each word or kanji is either *known* or *unknown* for a given user when tested. A correct answer upon subsequent encounter with the stimulus indicates a transition to the *known* state, an incorrect answer a transition to *unknown*. We populated the model with response data by examining again all stimulus pairs that an individual user encountered at least twice, and ordered that user's responses temporally into a sequence such as

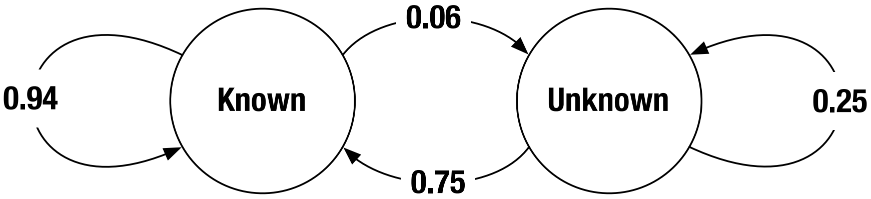


Figure 6.20: State transition diagram for known and unknown items. In this context "known" mean the user answered the most recent question about the item correctly.

⟨*incorrect*, *correct*, *correct*⟩. We used the first response in each sequence to determine the starting state, and each subsequent response as a transition. Assuming that each stimulus item had an equal chance of being learned (or not) after each test, we combined data from all stimulus items into a single state transition model, given in Figure 6.20.

The results are encouraging: known items have only a 6% chance of becoming unknown, but unknown questions have a 75% chance of becoming known. However, this model has limits on its predictive capacity. Exploring its performance characteristics, we find that the long-run likelihood of getting any item correct is $\frac{0.75}{0.06+0.75} = 0.93$, even after it has been tested many times. A previously unknown item reaches this level after it has been tested just three times. This artefact, a residual 7% error rate, suggests limits to the model. In particular, changes in user response do not always indicate changes in user knowledge, due to two main effects. Firstly, some correct answers are actually correct *guesses*. After a correct guess, the learner will not be prompted to study the item, and is thus less likely to improve their knowledge for subsequent tests. Secondly, knowledge of a kanji or word is graded and multi-faceted. In reality, a learner may easily recall one aspect of word knowledge (e.g. its meaning) but not another (e.g. its pronunciation). Success or failure on questions based around the same word may purely reflect the different aspects of word knowledge being tested, rather than indicating changes in word knowledge.

Ideally, both of these problems could be overcome through the addition of multiple states to our model. In the first case, states could be added for each of main aspect of knowledge, in particular for the links we showed earlier in Figure 6.3. In the second case, states could be added indicating two, three or four successive correct answers, in order to

User #	First language	JLPT	# responses	# tests	Mean accuracy	Pre/post difference	Time period
305.	English	L3	1160	24	0.839	0.056	5 days
251.	English	L4	860	18	0.914	0.056	12 hours
191.	Shona	L3	670	67	0.666	0.087	66 days
167.	German	L3	450	45	0.911	-0.058	62 days
253.	German	L4	440	16	0.925	0.033	27 days
Average	[English]	[L3]	75.7	4.99	0.880	0.110	3.7 days

Table 6.4: Basic user statistics for the 5 users with the most responses.

capture both the graded nature of knowledge and the weaker relationship between a correct answer and item knowledge due to guessing. Unfortunately, since Kanji Tester tests words and kanji randomly from the syllabus, data sparsity issues prevent us from examining these richer models.

Power users

Having examined estimation of user ability across the entire user population, we now restrict our focus to so-called "power users", the top 5 users in terms of number of tests taken. We provide some basic descriptive statistics for these users in Table 6.4, comparing them to the general user population.

These users differ in nearly every aspect, giving little indication of what motivated these users to use the system more than others. Their first languages include English and German, the largest two language groups, but also Shona, a Bantu language. They also include both syllabi supported by the system, JLPT levels 3 and 4.

Each user performed 6-15 times more tests than the population average, but varied widely as to the length of test they preferred: User 305 used almost exclusively tests with 50 questions, and User 191 used tests of 10 questions. Our power users vary around the mean accuracy level, with User 191 in particular having a very low accuracy compared to the population average. All of these users continued for long enough to be tested randomly on the same item twice, and thus can be given a pre/post difference score averaged across such items. User 167 was the only user amongst the five to show negative improvement; the other four all registered improvement, although less than the population average. This would seem counter-intuitive: if increased use is beneficial to users, then we would expect a greater-than-average improvement in the ability of power users. However, many users in the general user pool never encountered an item more than once, and those that did typically encountered few such items. The smaller this number of items, the noisier the pre-post estimate is for that user is. This suggests that the population average on this measure may be skewed in magnitude due to data sparsity.

Figure 6.21 gives an alternative view of each of the five power-users, plotting both the total number of unique stimulus items each was exposed to at the nth test, and the number they answered correctly on the last encounter. The graphs should be interpreted with an understanding of two caveats. Firstly, since tests taken differed in size according to each user's preferences, the number of items actually tested per data point can vary from one to the next. This could manifest itself in differing slopes at different points in time. Secondly, this effect could be compounded since the number of tests taken is only a rough indication of the passage of time: the time between two neighbouring tests could be a matter of minutes or weeks. With these provisos, we note that the "Tested" lines display the asymptotic behaviour we would expect from randomly choosing stimulus: they gradually approach the total number of kanji or words in each syllabus, but with each test the likelihood of re-encountering a previously seen item increases, hence the decreasing slope as the asymptote is approached.

The ratio between the number tested and the number correct when last tested is also interesting to observe between users. At any point in time, the ratio can be considered an estimate of the user's "true knowledge" of the syllabus's vocabulary at that point in time. For some users, for example user 251, a gap develops between the two lines for kanji knowledge, indicating a number of errors made. The gap later closes when those kanji are retested on later encounters. For user 191, a large gap emerges between the two lines and widens with increasing coverage of the syllabus, indicating the lower proficiency reached at the point of final testing. Interestingly, the final ratio of each user's lines visually agrees well the mean accuracy measurements for each user in Table 6.4, particularly on their word graphs: user 191 performs most poorly, followed by user 305. The remaining three power users are comparably high. User 191, who according to our pre/post ratio estimates improved the most over the testing period, also had the most room to improve. In comparison, the other

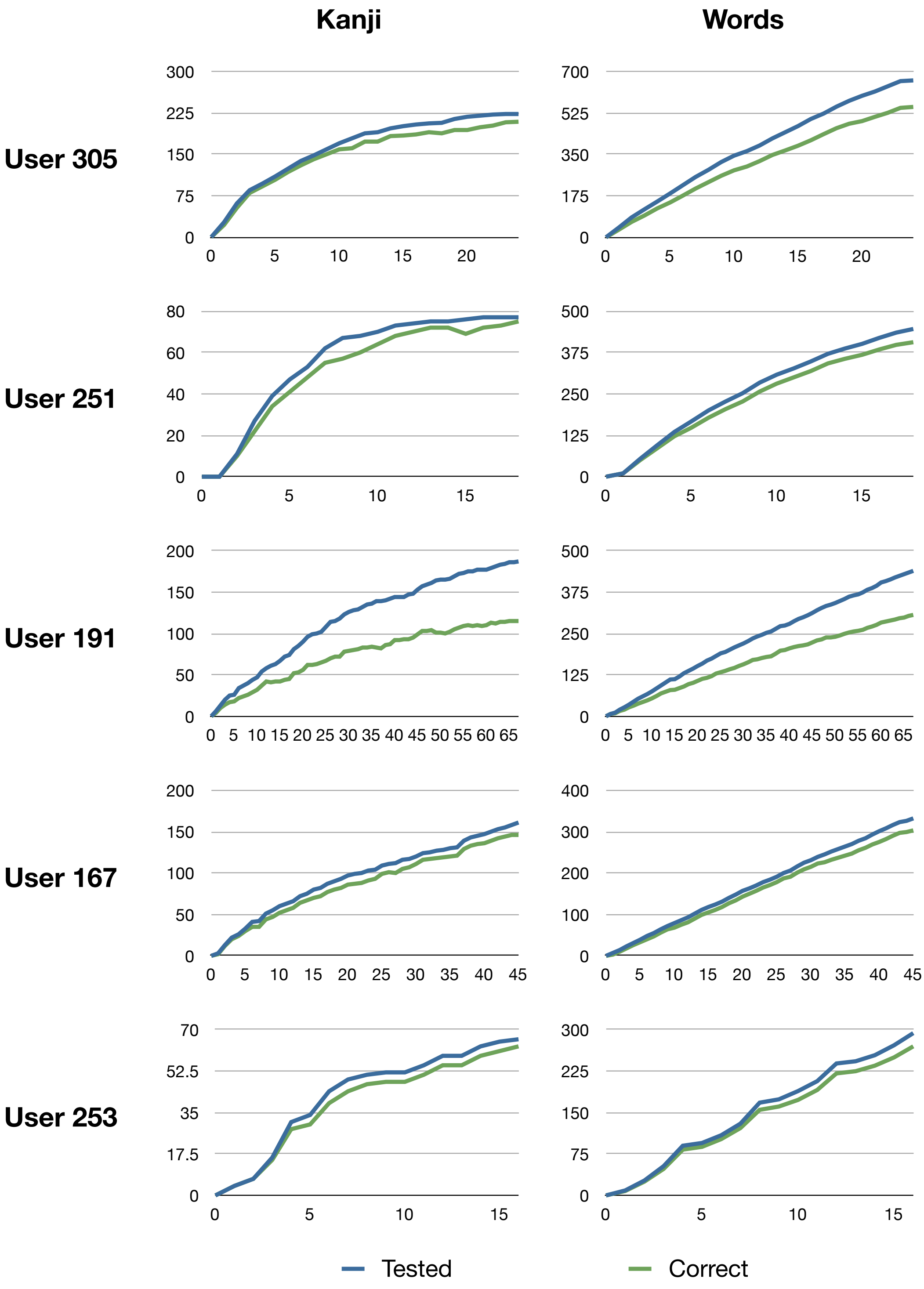


Figure 6.21: Accuracy and exposure for words and kanji by the top-5 power users. Each graph shows the total number of unique items tested after the nth test, and the number of items responded to correctly upon last encounter.

Name	Type	# questions	
Random glosses	Simple	11580	57.1%
Visual similarity	Adaptive	2567	12.7%
Reading alternations	Adaptive	2346	11.6%
Random surfaces	Simple	1948	9.6%
Random readings	Simple	1831	9.0%
Total		20272	100.0%

Table 6.5: Plugin name, type and number of questions generated.

users already had good knowledge of their syllabus, perhaps offering an explanation why they too did not show the same level of improvement, despite heavy use of the system.

Measuring adaptiveness

Kanji Tester is designed to generate authentic tests in two main ways: firstly, we make use of error models borrowed from the new FOKS system (Chapter 5). Secondly, we update these error models with user responses, allowing them to adapt the test to each user individually. As discussed in Section 6.3, Kanji Tester maintains two error models for each individual, namely a kanji reading model and a kanji misrecognition model, both of which are updated continuously with the user's responses. Recall that each user's tests alternate between *control* and *adaptive* versions, both of which consist of randomly generated questions. Control and adaptive tests consist of multiple choice questions of the same type, each with distractors selected randomly from the same pool. However, for control questions the distractors are sampled from the pool with uniform probability; for adaptive questions the distractors are sampled according to the likelihood of eliciting a user error predicted by the user's error model.

The control/adaptive distinction is designed to allow us to better evaluate the utility of having adaptive user models, and by having both forms of testing draw on the same distractor pools we set a strong baseline from which to improve. In this section, we examine the types of questions used and use the control/adaptive question distinction to measure the extent and utility of our adaptive user models.

Recall that each question in a test is generated by a plugin, with each plugin responsible

Item type	Mean # exposures
Words	12.0
Kanji	12.4
Kanji combined	45.7

Table 6.6: Mean number of exposures across all users by item type. The Kanji Combined type counts words containing kanji as exposures for those kanji.

for a limited range of question types. There are two types of plugins, namely *simple* plugins which do not adapt to the user, and *adaptive* plugins which maintain an error model which adjusts to user responses. Our plugins use the terms *gloss* to refer to a word's translation in English, *surface* to refer to a word's visual form, and *reading* to refer to its pronunciation.

Table 6.5 shows the distribution of questions generated by each question plugin. It is quickly apparent that the majority of questions are generated by the "Random glosses" plugin, covering some 57.1% of questions. This occurs because many words, especially at earlier levels of proficiency, are not yet represented by their full kanji form. This prevents us from using our more sophisticated adaptive question types, which centre around kanji misreading or misrecognition. In these cases, only the reading-meaning relationship could be tested, hence the large coverage of this plugin in particular. This is an immediate blow to adaptivity, since the adaptive plugins need user responses in order to tune the user models.

Now, our two adaptive user models are $(\text{reading}|\text{kanji})$ and $(\text{kanji}'|\text{kanji})$. In order for these models to adapt to an individual user, they must face questions generated using these distributions, i.e. generated by their corresponding adaptive plugins. However, Table 6.6 shows that across all users, each word has only 12 exposures on average, kanji not much higher at 12.4. Since each syllabus has many words and kanji, this means that on average each user won't encounter each word or kanji more than once. This is similarly a big problem for adaptivity at the per-user per-kanji level, although the situation is slightly improved because kanji may occur in many words, as the Kanji combined figure shows.

Given these figures, we should expect very little adaptivity from Kanji Tester in its generation of questions. However, we can nonetheless measure effectiveness of the adaptive plugins in comparison to their simple counterparts by comparing adaptive plugins to their simple counterparts. Recall that for each user, every second test completed was a control test which used simple plugins only. Simple plugins used the same error candidates, but ignored their probability distribution and simply used a uniform distribution across candidates.

Figure 6.22 provides error rates for each plugin, and shows that questions based on surface form are the easiest, whereas those based on reading are the hardest. Comparing the simple and adaptive plugins, the gap between the two reading question plugins is substantial. Using a two-tailed Student's t-test between error vectors shows the difference is significant to the 99% level. This suggests that learners make many more reading errors when the distractors are plausible misreadings. On the other hand, simple and adaptive surface questions are roughly comparable in error rate, with their difference significant only to a 20% level. This could indicate either that learners make few errors of this type, or that our priors are too weak for these models. Note that questions based on gloss have no adaptive counterpart, since no appropriate error model could be easily constructed which could guarantee that distractor glosses were actually incorrect.

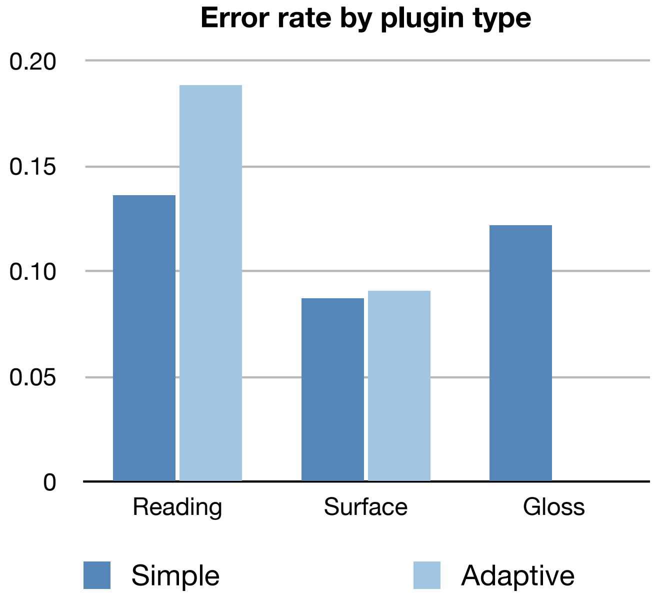


Figure 6.22: Error rates compared for simple and adaptive plugins.

In general it appears that adaptive questions are able to generate more user errors compared to their simple counterparts, but how was this possible if they adapted little to each user? The more difficult adaptive questions are generated because the user models use rebuilt FOKS error models for priors. Since per-user error models adapt little, they are in general similar across all users, but still effective in generating distractors due to these priors.

Error models

We used two main error models: our reading alternation model (reading|kanji), and our grapheme alternation model (kanji′|kanji). A significant motivation for constructing Kanji Tester was the opportunity to test these models proactively rather than reactively. FOKS attempts to recover from user queries, but if users are conservative in their querying, we may encounter only limited forms of errors. Since Kanji Tester is a more constrained environment, we can attempt to provoke errors by introducing distractors to users, and thus obtain greater coverage over error forms. This greater coverage comes at a cost: users of dictionaries have at least partial knowledge of the word, whereas the worst-case scenario for a test question is where the user has no applicable knowledge, and guesses near-randomly. Nonetheless, the difference in error patterns between the two applications is worth examining.

We first consider the reading alternation model, which provoked significantly more errors than average. Using the same query explanation feature of FOKS, we can recover the types of errors made by users. Table 6.7 provides the distribution of error types users made, and is an interesting comparison with our earlier analysis of FOKS queries (see Table 5.7). Whilst choice of reading remains the most common error type, non-compositional reading errors are significantly reduced. This is most likely because the syllabi available for testing contain few proper nouns, which are often non-compositional. The other main difference is the increase in voicing errors compared to vowel length errors. Although some voicing errors were cases of sequential voicing, such as choosing *tekami* as the reading for 手紙 *tegami* "letter", many also mistakenly voiced initial constants, for example choosing *dzuki* as the reading for 月 *tsuki* "month".

There are two plausible reasons why these additional voicing errors are being made by learners. Firstly, they may have encountered kanji or words in larger compounds first, where sequential voicing was correctly applied. From these encounters, they may assume mistakenly that the reading of the item outside the compound is also voiced this way. For example, if the learner knows the place name 新橋 *shiNbashi*, they may mistakenly read 橋 "bridge" as

Error type	**Frequency**	
Choice of reading	174	76.0%
Voicing error	23	10.0%
Vowel length	14	6.1%
Non-compositional reading	9	3.9%
Sound euphony	8	3.5%
Palatalisation	1	0.4%
Total	229	100.0%

Table 6.7: Error types for adaptive reading questions.

Word/kanji			**# Errors**	**Users chose**
急度	*kitto*	"without fail"	7	急言/多計/意文/急来/思度/悪言
出る	*deru*	"to appear"	6	千る/高る/南る/者る/計る
意	*i*	"feelings"	5	思/春
長い	*nagai*	"long"	4	大い/英い/足い
題	*dai*	"subject"	4	員/貸/買
代	*dai*	"price"	4	貸/会/仕
仕方	*shikata*	"way"	4	生方/仕走/工去/行方
小父さん	*ojisaN*	"uncle"	4	小入さん/少父さん/小大さん

Table 6.8: Top 8 words or kanji by number of grapheme substitution errors.

bashi instead of its correct reading *hashi*. Secondly, they may be confused by the graphemic similarity between pronunciation strings, which are provided in kana. In Japanese, voicing is marked in hiragana and katakana by a diacritic. For example, *ha*, *ba* and *pa* (unvoiced, voiced, semi-voiced) are written as は, ば *a*nd ぱ. Since these diacritics are small, learners could easily visually mistake voiced and unvoiced variants of these kana. Such errors are not explicitly modelled as reading alternations; incorporating knowledge of them into future systems would prove interesting.

We now examine grapheme substitution errors. Table 6.8 shows the most common errors made by users of Kanji Tester. On the right of each word, we see distractors which different users mistook for the correct answer. From these results, we can see mixed levels of apparent similarity between the user errors and the correct answer. For example, pairs such as 小父さん and 小大さん seem visually similar, yet others such as 出る and 者る do not. The presence of the latter type of pair suggests that in some cases, the user simply guessed randomly amongst alternatives.

This may have happened in our reading results as well, but each reading result is also a plausible misreading, whereas noise in our graphemic distance models means that some neighbours generated by the model will share little visual similarity. An alternative explanation of the same problem is that some regions in grapheme space are visually dense, and others visually sparse. Sparse items have few plausible neighbours, yet we must still generate a sufficiently large distractor pool for such items for use in testing. Thus, even a very accurate graphemic distance model is forced to occasionally include neighbours beyond a plausible misrecognition threshold.

6.5 Discussion

In this chapter we have discussed a system allowing users to test their knowledge of Japanese vocabulary, within the scope of two supported syllabi. The goal of the system was to allow repeatable, accurate self-evaluation by replicating limited aspects of human tests. Have we succeeded?

We can measure success in several ways. We have clearly built a system that allows rapid self-evaluation, assured by our random sampling over the user's chosen syllabus and question generation which is at least minimally resistant to random guessing. However, any system with these features would achieve this aim without requiring significant error modelling. On the other hand, have we replicated human tests?

Our reading alternation questions are certainly successful, eliciting many more errors than their control counterparts. Our grapheme alternation questions however were comparable to their control counterparts, and thus do not match a human generated standard. There are several problems we face generating such questions. Firstly, our best graphemic distance model – stroke edit distance – still has significant noise when compared with human judgements. This noise must be reduced if we are to generate questions with more difficult distractors. Secondly, the natural sparsity of some areas in the visual similarity space means that there are simply not enough motivated distractors to generate difficult questions for all kanji or all words. This is solved in human-generated tests by the use of plausible non-kanji, made up of real components in combinations which do not occur in real-life. Ultimately, the only way to solve the data sparsity problem is to find a way to automatically generate distractors using such non-kanji; as yet no such automated system exists, though use of Chinese, Korean or archaic variants could alleviate this concern.

We drew a distinction between different forms of testing in Section 6.1, from flashcards to human-generated tests, and claimed that Kanji Tester would provide the availability of flashcards, yet with validity and scope closer to human tests. In its current form, Kanji Tester serves as a useful intermediary between these two forms of testing. Since it is not yet as powerful as human-generated tests, it cannot replace them. Yet neither can it replace flashcards, for the simple reason that flashcards provide a progression through the syllabus which facilitates learning. In comparison to flashcards, Kanji Tester is superior in the level of self-assessment, yet inferior as a drilling tool. Could this be improved?

Whilst flashcards will continue to be useful, a future system could certainly benefit by simply providing a linear progression through the syllabus, returning repeatedly to items responded to incorrectly. For example, it could use Leitner's (1972) spaced repetition method, as discussed in Section 3.1. This linear progression would ensure full eventual coverage of the user's syllabus. However it would not be useful for testing the user's vocabulary knowledge on the whole syllabus. For that, the current random sampling of the syllabus gives better validity. Although the current method of testing ability has not been explicitly evaluated against current forms of paper testing. It cannot either be used by teachers to make decisions about their students without such formal comparisons.

For these reasons, the suggested future system would thus provide two modes of testing. One which provided an alternative to flashcards, as a means of vocabulary study. The other, purely as a means of vocabulary self-assessment, yet evaluated formally against other tests to prove its validity. This formal assessment would allow it to bridge the divide between learner tools for self-use, and tools to aid language educators in rapid assessment of their students.

Overall, the current Kanji Tester system provides a compelling example of increasing possibility of generating intelligent learning tools for learners, based on strongly motivated error modelling. The better we can understand learners, the better we can adapt to their needs. Applied linguists have long looked at the minutiae of learner errors to better understand how they acquire language; our work does the same, but uses those errors to support learners in their self-study and self-discovery of language.

6.6 Conclusion

Nearly all forms of language learning can be construed as testing. In this chapter we have bound together all aspects of our work and embedded them in a vocabulary testing application, Kanji Tester, which approximates the well known JLPT test family in its coverage and testing of vocabulary. Kanji Tester makes use of both intelligent kanji (mis)reading models from the FOKS dictionary (Chapter 5), and a new kanji (mis)recognition model generated from the novel stroke edit distance metric (Chapter 4). Furthermore, we explored how these error models might change over time in response to user input. In particular, the adaptive reading model achieved a statistically significant increase in the number of user errors elicited.

The data sparsity issues we encountered suggest two possible directions the existing system could take in the future. Firstly, Kanji Tester could attempt to move to a user model of coarser granularity, for example by pooling user models for learners from the same first language background. Secondly, Kanji Tester could divide itself into two applications: a drilling application, which would revisit stimulus according to a spaced repetition schedule instead of randomly, and a testing application, meant for occasional authoritative testing. This would better match the patterns of current use, and by revisiting incorrect items more frequently in drill mode data sparsity issues would be alleviated. Fortunately, the current body of user responses serves as a useful data set on which to evaluate new and as yet untested user models, without having to deploy them and collect new user data over time. Over longer time periods, it may also serve as a useful data set for measuring learner improvement.

More broadly, Kanji Tester has limited itself to vocabulary testing and focused on kanji-based errors. There are many additional aspects to proficiency which could be tested, and a template-based systems such as Zock and Afantenos's (2007) pattern drills seem a good partner for our approach. A current shortcoming of Kanji Tester is its lack of intelligent gloss-based questions. A large barrier to such questions currently lies in the need to avoid using too-near-synonyms as distractors, since their use poses the risk of unintentionally generating questions with more than one valid answer. If this risk could be avoided through use of semantic similarity measures, stronger testing would be possible.

Chapter 7

Conclusions

7.1 Conclusions

This thesis has investigated the use of linguistic error modelling to support language learners in their autonomous self-study of Japanese vocabulary. Acquiring sufficient vocabulary is perhaps the most difficult task in language learning because the breadth of knowledge to be acquired is enormous, too much to be taught explicitly. The most commonly advocated solution to acquiring large amounts of vocabulary is to read widely, yet Nuttall (1996) found a system of positive feedback not only in successful reading, but in unsuccessful reading too. This indicates some sort of tipping or boundary point beyond which successful readers propel themselves towards fluency, and before which reading is of limited use and can even reduce motivation. Laufer (1997) suggests this tipping point lies at the 3000 word family mark for vocabulary knowledge. The central problem in vocabulary learning is thus to allow learners to reach this mark, and to support their early attempts at reading.

This thesis has made three main contributions towards solving this problem. Firstly, noting the potential for lexical relationships to support vocabulary growth, it examined in depth graphemic relationships between words and characters based on similarity of visual form, and compared several novel similarity metrics for Japanese kanji. In order to do this, it has also generated three useful data sets for the evaluation of such models, not including the additional data which will continue to be generated over time by both the FOKS and

Kanji Tester systems as they are used. In its evaluation, it has found the stroke edit distance and tree edit distance metrics to most effectively match human judgements, and based on the available data has identified stroke edit distance as the preferred distance metric due to its relative efficiency in comparison to its tree-based cousin.

The second main contribution involved the transformation of stroke edit distance into a confusability model for Japanese kanji, and the incorporation of this model into an existing dictionary to support search-by-similarity. Several other minor enhancements were made to the FOKS dictionary as part of this rebuild, including the addition of a more scalable grapheme-phoneme alignment algorithm. Upon post-hoc analysis of log data, our new search method was found to have been quickly adopted by users, to the extent that searches by similarity outnumbered the previous form of intelligent reading search during the same log period, indicating their utility.

Finally, we combined existing phonemic and graphemic error models applied these to the new area of adaptive testing in the form of the Kanji Tester site. By emulating the authoritative JLPT test, Kanji Tester aimed to allow users better self-evaluation through rapid and repeatable proficiency testing. Through analysis of user responses over a several month log period, we established that use of adaptive testing increased the difficulty of test questions significantly over a control baseline.

These theoretical and practical contributions combined to form a platform for second-language vocabulary growth in Japanese.

Graphemic similarity

Four new graphemic distance models were proposed for Japanese kanji, based on radicals, pixels, tree-based structure and strokes. These distance models were evaluated on three main data sets: explicit human similarity judgements, expert judgements from a commercial flashcard set, and native speaker judgements from a candidate pool experiment.

The first experiment showed that for low-to-medium similarity pairs, shared radicals and similar layout were most important for measuring similarity. Furthermore, the results suggested that non-speakers of Japanese do not differ fundamentally in their judgements to native speakers, but rather their judgements gain in consistency as their experience with such characters increases. The flashcard data set consisted of expert-selected high-similarity pairs, and here radical methods performed the most poorly across all methods of measurement. Instead, stroke and tree-based methods modelled most closely the expert judgements, followed by pixel comparison of kanji images. A distractor pool experiment eliciting native speaker judgements confirmed this ordering, suggesting that for high-similarity pairs, both form and layout of components were important. The success of the stroke-based metric was explained in terms of its success in fuzzy matching of structural and form-based features, and due to its efficiency in comparison to other metrics, it was established as the preferred metric given our data.

Dictionary search

We examined in detail how an error-correcting dictionary, FOKS, could be extended and augmented to increase its accessibility and thus its support for early reading. As part of a complete rebuild, we modified the dictionary's grapheme-phoneme alignment algorithm, showing that use of a kanji reading resource allowed far faster alignment times through stronger alignment constraints. Furthermore, its TF-IDF scoring function was shown to perform better when reduced to the IDF component alone, due to the importance of segmenting maximally in the GP alignment task.

We extended the dictionary's coverage of place names by making use of a simple gazetteer mined from Japan Post, and improved the display of translations to accommodate richer word information and yet remain manageable for users. We then made FOKS's internal error models transparent to users by providing a query explanation tool, which also serves to explain the derivation of correct kanji compound readings.

Finally, and most significantly, we incorporated our stroke-based grapheme distance metric into the dictionary to form a new error-correcting search by visual form. This innovative form of search allows users to search for visually similar words using their known neighbours. For example, a user could use the query 補左 to find the word 補佐 *hosa* "help", based on the similarity between 左 and 佐. Query logs showed that this new form of search is actively used, and is thus providing benefit for learners.

Vocabulary testing

We demonstrated how linguistic error models can be transferred from the dictionary search domain to the language testing domain, where they can actively provoke user errors rather than passively correcting for them. The reading and graphemic error models from the rebuilt FOKS system were used to develop Kanji Tester, an adaptive testing system which generates multiple choice questions which are potentially unique for every user and every test.

We discussed various methods for grouping users and errors at differing levels of granularity, ultimately choosing to model users as individuals and errors at the kanji level. We then proposed an update rule for this form of error modelling, and applied this rule to make tests adapt to user responses, increasing the likelihood of distractors which previously provoked an incorrect response.

Finally, we performed extensive log analysis to determine the relationship between use of the system and user scores on the system, which serve as estimates of receptive vocabulary knowledge. On the whole we found a weak positive correlation between time and test scores, suggesting that users of the system improved in ability the longer they used the system. We also compared adaptive and control question types, and found that adaptive questions caused more errors than control questions, especially for the reading error model. Data sparseness limited the adaptivity of error models to users, however strong prior distributions meant that tests were sufficiently difficult nonetheless.

7.2 Future work

Many open issues remain for future investigation. The following sections describe these issues, aligning them roughly to our main areas of contribution.

Lexical relationships

In this thesis we focused on graphemic similarity as a novel and under-utilised relationship between words, and measured this similarity through a range of distance metrics which we evaluated. Our best performing metric, stroke edit distance, still contains significant noise which impedes better lookup and testing. However, this area of study is no longer data poor. We have contributed five data sets for evaluating graphemic similarity: two experiments, flashcard distractors, and two sets of log data. These provide a strong foundation for the development of better performing graphemic distance metrics.

A simple direction to consider is the appropriate costs to use for tree and stroke edit distance calculations; we used unit costs, but if some features prove to be far more important than others, variable costs might provide a better result. More broadly, there is also scope to examine a broader range of OCR techniques for Japanese kanji, and to develop distance metrics from these techniques.

Any such metric used generates a topology on the space of kanji. Interesting aspects of these topologies should be explored, and explained with reference to models of kanji perception. For example, orthographic density about a word has already been explored in psycholinguistics; this could be extended to examine and explain differences between low and high-density regions of orthographic space.

The limits of visual neighbour accessibility should be explored to determine the drop-off in neighbour availability as kanji grow more visually distant from one another. Visual neighbours that are close enough can be considered linked; the resulting network may have interesting properties worth exploring, and comparing with Yamamoto and Yamazaki's (2009) recent work on network properties of kanji compounds. There are also other graphemic relationships other than similarity, for example containment. These alternative relationships may prove more accessible to learners than similarity, or at the very least complementary, and should thus be examined.

Beyond graphemic relationships, there is much work to be done in mapping out, making sense of and unifying the various forms of semantic relationships, since doing so will greatly aid lexicography, and will provide meaningful new methods of understanding words and their relationships to each other.

Dictionary search

In this thesis we took an existing dictionary, FOKS, and rebuilt it to increase its accessibility, however we believe there remain improvements worth investing in. Some of these opportunities lie in correcting known error types, such as errors due to mistaken voicing of kana; others are in helping learners understand the mistakes they have made, for example by providing a spatial explanation of how two similar kanji differ.

We have already mentioned the need for improved graphemic distance models, but any other form of lexical relationship can be used to augment a dictionary in useful ways. For example, semantic relatedness could be used to provide nearby words to users for comparison.

The most interesting area to pursue is the ability to provide users with the means to extend the dictionary in useful ways, thus "crowdsourcing" dictionary enhancements. The Tatoeba project[1] already does this for example sentences, so that representative examples of word usage can be provided by users themselves. This could be combined with semantic similarity measures between words, so that very similar words could have example sentences provided which demonstrate the difference between them. Choosing between near-synonyms is extremely difficult for learners, since they lack contextual knowledge about when each word is appropriate, and what additional meaning a word may connote. An appropriate dictionary extension would help solve this difficult problem, and would thus alleviate the difficulties associated with lexical gridding issues between a learner's first and second languages. There have been several other recent attempts to render dictionaries far more useful and useable, for example, use of semagrams as a rich semantic encoding of words in dictionaries (Moerdijk *et al.* 2008). Tapping into user populations to crowdsource this additional richness would also be an approach worth exploring.

An alternative approach for improving dictionary accessibility would be to conduct formal user studies of dictionary search, for example using task timing and verbal protocols (Ericsson and Simon 1985) to elicit hard data on dictionary lookup performance. Such data could provide a principled means of discriminating between existing dictionary offerings based on accessibility, or could be used to look for particular points of difficult which remain for learners.

[1] `http://tatoeba.org/`

Proficiency testing

We have shown that adaptive vocabulary testing is possible, and argued for its future role in replacing human-constructed tests. Kanji Tester is an important first step, but there is still a significant gap between human tests and automated tests. Several open research areas could help close this gap.

Firstly, we know that vocabulary knowledge is graded, since there are many aspects of word knowledge, and that productive tests are more difficult than receptive tests, such as Kanji Tester. If Kanji Tester incorporated both forms of test question, the distinction between these two types or levels of knowledge could be analysed and solidified through extensive log analysis.

Secondly, having argued for the importance of vocabulary knowledge, we could attempt to estimate the size of a user's vocabulary knowledge directly, for example using a word familiarity database based on the method of Amano and Kondo (1998). Providing this estimate to learners would allow them to set vocabulary size goals and measure progress towards achieving such goals.

We also noted the various levels and granularities of user and error models available for use, but only experimented with a single of these combinations. By using the existing user response data, we could evaluate different user model combinations, and determine what model best predicts actual user behaviour. This in turn would provide useful insights as to the extent to which user errors and misconceptions are shared across different cross-sections of the learner community.

There is significant scope for better receptive question generation, through new and different error models. Existing work on transliteration, such as that by Pervouchine *et al.* (2009), could be leveraged to generate plausible distractors for katakana loan-words. Learners are also known to confuse words which are semantically similar (Mondria 2007). Semantic similarity measures, as discussed in the previous section, could thus be used to generate error-provoking gloss distractors based on semantic similarity. For this purpose, Japanese WordNet (Isahara *et al.* 2008) could be used, choosing siblings or cousins as semantic-similarity-based distractors.

The ability to hand-write Japanese kanji is a productive skill requiring recall of form rather than pronunciation, one which we have not discussed at length in this thesis. Modern hand-writing dictionary interfaces such as those discussed in Section 2.2 suggest a way that this skill could be incorporated automatically into tests. Tests requiring both input of kanji by form and input of typed kanji by pronunciation could provide interesting data on the relative productive abilities of learners with respect to form vs sound recall.

Nearly all our work on Japanese can be easily extended to Chinese, especially work on graphemic error modelling, since the Chinese writing system uses similar characters to Japanese kanji. The Pinyomi dictionary (Yencken *et al.* 2007) is a straightforward candidate for such improvement. Its Japanese-Chinese dictionary interface based on lookup-by-transliteration could easily support search by similar kanji or hanzi. Doing so would involve rebuilding one of the similarity models to encompass both hanzi and kanji, but the improvement in accessibility could be significant. Such multilingual models could also be useful in monolingual testing, since they could alleviate the sparse visual neighbour problem.

Finally, recall that in Chapter 3 we noted our limited coverage of the Chinese- and Japanese-language literature, especially as relates to native speaker development. We encourage future multi-lingual surveys into these areas, as techniques designed to aid second language learners may equally prove useful for native speakers, especially child learners during their development or adults facing rare words.

7.3 Summary

In summary, this thesis argued for better support for second language learners in their vocabulary study and early reading experiences, and indeed provides support in both areas. Firstly, it developed new linguistic error models based on graphemic proximity of words and kanji. Secondly, it extended an existing dictionary, improving its usability and accessibility through graphemic-error correcting search. Finally, it used error models to provide an adaptive testing interface for learners to self-evaluate their study progress. Together, these applications form a platform to assist learners in their autonomous self-study of vocabulary, and thus in their ultimate progression towards fluency.

Bibliography

Allen, Jonathan, Sheri Hunnicut, and Dennis Klatt. 1987. *From Text To Speech, The MITALK System*. Cambridge, UK: Cambridge University Press.

Amano, Shigeaki, and Tadahisa Kondo. 1998. Estimation of mental lexicon size with word familiarity database. In *Proceedings of the 5th International Conference on Spoken Language Processing*, volume 5, 2119–2123, Sydney, Australia.

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 *Proceedings of the COLING 2004 Workshop on Enhancing and Using Electronic Dictionaries*, 36–39, Geneva, Switzerland.

Baayen, Harald. 1993. Statistical models for word frequency distributions: A linguistic evaluation. *Computers and the Humanities* 26.347–363.

Backhouse, A. E. 1996. *The Japanese Language: An Introduction*. Melbourne, Australia: Oxford University Press.

Baddeley, A. D. 1997. *Human memory: Theory and Practice*. Boston: Allyn and Bacon, revised ed edition.

Baddeley, Alan. 2003. Working memory and language: an overview. *Journal of Communication Disorders* 189–208.

Baldwin, Timothy, and Hozumi Tanaka. 1999a. The applications of unsupervised learning to Japanese grapheme-phoneme alignment. In *Proceedings of the 1999 ACL Workshop on Unsupervised Learning in Natural Language*, 9–16, College Park, USA.

——, and ——. 1999b. Automated Japanese grapheme-phoneme alignment. In *Proceedings of the International Conference on Cognitive Science*, 349–354, Tokyo, Japan.

——, and ——. 2000. A comparative study of unsupervised grapheme-phoneme alignment methods. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*, 597–602, Philadelphia, USA.

Barker, David. 2007. A personalized approach to analyzing 'cost' and 'benefit' in vocabulary selection. *System* 4.523–533.

Begg, Andy. 1997. Some emerging influences underpinning assessment in statistics. In *The Assessment Challenge in Statistics Education*, ed. by I. Gal and J. B. Garfield, chapter 2, 17–25. Amsterdam, Netherlands: IOS Press.

Bell, Timothy. 1998. Extensive reading: Why? and how? *The Internet TESL Journal* 4.

Bilac, Slaven, 2005. *Automatically extending the dictionary to increase its coverage and accessibility*. Tokyo Institute of Technology dissertation.

——, Timothy Baldwin, and Hozumi Tanaka. 1999. Incremental Japanese grapheme-phoneme alignment. In *Information Processing Society of Japan SIG Notes*, volume 99-NL-209, 47–54.

——, ——, and ——, 2003. Modeling learners' cognitive processes for improved dictionary accessibility. Handout at *10th International Conference of the European Association for Japanese Studies*, Warsaw, Poland.

——, ——, and ——. 2004. Evaluating the FOKS error model. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2119–2122, Lisbon, Portugal.

——, and Hozumi Tanaka. 2005. Direct combination of spelling and pronunciation information for robust back-transliteration. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, 413–424, Mexico City, Mexico.

Bille, Philip. 2005. A survey on tree edit distance and related problems. *Theoretical Computer Science* 337.217–239.

Black, Alan W., Paul Taylor, and Richard Caley, 1999. *The Festival speech synthesis system*. System documentation.

Breen, Jim. 1995. Building an electronic Japanese-English dictionary. In *Proceedings of the 1995 Japanese Studies Association of Australia Conference*, Brisbane, Queensland.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19.263–311.

Brown, Roger, and David McNeill. 1966. The "tip of the tongue" phenomenon. *Journal of Verbal Learning and Verbal Behavior* 5.325–337.

Brusilovsky, Peter. 2001. Adaptive hypermedia. *User Modeling and User-Adapted Interaction* 11.87–110.

Bull, Joanna, and Colleen McKenna. 2004. *Blueprint for Computer-Assisted Assessment*. London, UK: Routledge Falmer.

Carreiras, Manuel, Manuel Perea, and Jonathan Grainger. 1997. Effects of orthographic neighborhood in visual word recognition: Cross-task comparisons. *Journal of Experimental Psychology: Learning, Memory and Cognition* 23.857–871.

Chalhoub–Deville, Micheline, and Craig Deville. 1999. Computer adaptive testing in second language contexts. *Annual Review of Applied Linguistics* 19.273–299.

Clark, Stephen, and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of 20th International Conference on Computational Linguistics*, 282–288, Geneva, Switzerland.

Coady, James. 1997. L2 vocabulary acquisition through extensive reading. In *Second Language Vocabulary Acquisition*, ed. by James Coady and Thomas Huckin, 225–237. Cambridge, UK: Cambridge University Press.

Cohen, William W., Pradeep Ravikumar, and Stephen E. Feinberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, 73–78, Acapulco, Mexico.

Craik, Fergus I. M., and Endel Tulving. 1975. Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology: General* 104.268–294.

Cruse, D. A. 1986. *Lexical semantics*. New York: Cambridge University Press.

Demaine, Erik D., Shay Mozes, Benjamin Rossman, and Oren Weimann. 2007. An optimal decomposition algorithm for tree edit distance. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming*, 146–157, Wrocław, Poland. Springer.

Dycus, David. 1997. Guessing word meaning from context: Should we encourage it? *Literacy Across Cultures* 1.1–6.

Ericsson, K. Anders, and Herbert A. Simon. 1985. *Protocol Analysis: Verbal Reports as Data*. Cambridge, Massachusetts: MIT Press.

Espinosa, María Paz, and Javier Gardeazabal. 2007. *Optimal Correction for Guessing in Multiple-Choice Tests*. Technical report, Department of Foundations of Economic Analysis II, University of the Basque Country.

EUGENIO, BARBARA DI, and MICHAEL GLASS. 2004. The kappa statistic: A second look. *Computational Linguistics* 30.95–101.

FELLBAUM, CHRISTIANE. 1998. *Wordnet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1st edition.

FERRET, OLIVIER, and MICHAEL ZOCK. 2006. Enhancing electronic dictionaries with an index based on associations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, 281–288, Sydney, Australia.

FLORES D'ARCAIS, GIOVANNI B., HIROFUMI SAITO, and MASAHIRO KAWAKAMI. 1995. Phonological and semantic activation in reading kanji characters. *Journal of Experimental Psychology: Learning, Memory and Cognition* 21.34–42.

FRASER, CAROL A. 1999. Lexical processing strategy use and vocabulary learning through reading. *Studies in Second Language Acquisition* 21.225–241.

GASKELL, M. GARETH, and NICOLAS DUMAY. 2003. Lexical competition and the acquisition of novel words. *Cognition* 89.105–132.

GAUME, BRUNO, KARINE DUVIGNAU, LAURENT PRÉVOT, and YANN DESALLE. 2008. Toward a cognitive organization for electronic dictionaries, the case for semantic proxemy. In *Proceedings of the workshop on Cognitive Aspects of the Lexicon*, 86–93, Manchester, UK.

GOH, CHOOI-LING, MASAYUKI ASAHARA, and YUJI MATSUMOTO. 2005. Building a Japanese-Chinese dictionary using kanji/hanzi conversion. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, 670–681, Jeju Island, Korea.

HALPERN, JACK (ed.) 1999. *The Kodansha Kanji Learner's Dictionary*. Tokyo, Japan: Kodansha International.

HAMBLETON, RONALD K., and RUSSELL W. JONES. 1993. Comparison of Classical Test Theory and Item Response Theory and their applications to test development. *Educational Measurement: Issues and Practice* 12.38–47.

HANDKE, JÜRGEN. 1995. *The Structure of the Lexicon: Human vs Machine*. Berlin: Mouton de Gruyter.

HARRIS, ZELLIG S. 1954. Distributional structure. *Word* 10.146–162.

HEIFT, TRUDE, and DEVLAN NICHOLSON. 2001. Web delivery of adaptive and interactive language tutoring. *International Journal of Artificial Intelligence in Education* 12.310–324.

HEISIG, JAMES W. 1985. *Remembering the Kanji I: A Complete Course on how Not to Forget the Meaning and Writing of Japanese Characters*. Tokyo, Japan: Japan Publication Trading Co.

HSUEH-CHAO, MARCELLA HU, and PAUL NATION. 2000. Unknown vocabulary density and reading comprehension. *Reading in a Foreign Language* 13.403–430.

HUCKIN, THOMAS, and JAMES COADY. 1999. Incidental vocabulary acquisition in a second language. *Studies in Second Language Acquisition* 21.181–193.

IKEHARA, SATORU, MASAHIRO MIYAZAKI, SATOSHI SHIRAI, AKIO YOKOO, HIROMI NAKAIWA, KENTARO OGURA, YOSHIFUMI OOYAMA, and YOSHIHIKO HAYASHI. 1997. *Goi-Taikei – A Japanese Lexicon*. Tokyo, Japan: Iwanami Shoten.

ISAHARA, HITOSHI, FRANCIS BOND, KIYOTAKA UCHIMOTO, MASAO UTIYAMA, and KYOKO KANZAKI. 2008. Development of the Japanese wordnet. In *Proceedings of the 6th International Language Resources and Evaluation*, Marrakech, Morocco.

JAPAN FOUNDATION, 2009. *What is the JLPT: Contents of the Test*. Available at `http://www.jlpt.jp/e/about/content.html`. Accessed 7th June 2009.

JIN, ZHIHUI. 2008. Pinyomi-lite - improved Japanese-Chinese dictionary lookup using logograph transliteration. In *Proceedings of the 14th Annual Meeting of the Japan Society for Natural Language Processing*, Tokyo, Japan.

JOYCE, TERRY. 1999. Lexical access and the mental lexicon for two-kanji compound words: A priming paradigm study. In *Proceedings of the 7th International Conference on Conceptual Structures*, 1–12, Blacksburg, VA, USA.

——. 2002. Constituent-morpheme priming: Implications from the morphology of two-kanji compound words. *Japanese Psychological Research* 44.79–90.

——. 2005. Constructing a large-scale database of Japanese word associations. *Glottometrics* 10.82–98.

KNIGHT, KEVIN, and JONATHAN GRAEHL. 1998. Machine transliteration. *Computational Linguistics* 24.599–612.

KODA, KEIKO. 2007. Reading and language learning: Crosslinguistic constraints on second language reading development. *Language Learning* 57.1–44.

KONDRAK, GRZEGORZ. 2003. Identifying complex sound correspondences in bilingual wordlists. In *Proceedings of 4th International Conference on Computational Linguistics and Intelligent Text Processing*, 432–443, Mexico City, Mexico.

Krashen, Stephen. 1989. We acquire vocabulary and spelling by reading: Additional evidence for the input hypothesis. *Modern Language Journal* 73.440–464.

Landauer, T. K., and L. A. Streeter. 1973. Structural differences between common and rare words: Failure or equivalence assumptions for theories of word recognition. *Journal of Verbal Learning and Verbal Behavior* 12.119–131.

Landauer, Thomas K., and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review* 104.211–240.

Laufer, Batia. 1997. The lexical plight in second language reading; words you don't know, words you think you know, and words you can't guess. In *Second Language Vocabulary Acquisition*, ed. by James Coady and Thomas Huckin, Cambridge Applied Linguistics, 20–34. Cambridge, UK: Cambridge University Press.

——, and Zahava Goldstein. 2004. Testing vocabulary knowledge: Size, strength and computer adaptiveness. *Language Learning* 54.399–436.

——, and Tamar Levitzky-Aviad. 2006. Examining the effectiveness of 'bilingual dictionary plus' – a dictionary for production in a foreign language. *International Journal of Lexicography* 19.135–155.

——. 1991. *Similar Lexical Forms*. Tübingen, Germany: Gunter Narr Verlag Tübingen.

Leitner, S. 1972. *So lernt man lernen*. Freiburg, Germany: Herder.

Levelt, Willem J. M., Ardi Roelofs, and Antje S. Meyer. 1999. A theory of lexical access in speech production. *Behavioral and Brain Sciences* 1–75.

Lupker, Stephen J. 2005. Visual word recognition: Theories and findings. In *The Science of Reading: A Handbook*, ed. by Margaret J. Snowling and Charles Hulme, chapter 3. Carlton, VIC, Australia: Blackwell Publishing.

Marso, Ronald N. 1969. The influence of test difficulty upon study efforts and achievement. *American Educational Research Journal* 6.621–632.

McClelland, James L., and David E. Rumelhart. 1981. An interactive activation model of context effects in letter perception, Part 1: An account of basic findings. *Psychological Review* 88.375–407.

Meara, Paul. 2009. *Connected Words: Word associations and second language Vocabulary acquisition*. John Benjamins Publishing Company.

Miller, Georgia E. 1941. Vocabulary building through extensive reading. *The English Journal* 30.664–666.

MOERDIJK, FONS, CAROLE TIBERIUS, and JAN NIESTADT. 2008. Accessing the ANW dictionary. In *Proceedings of the 2008 Workshop on Cognitive Aspects of the Lexicon*, 18–24, Manchester, UK.

MOHSENI-FAR, MOHAMMAD. 2008. In search of the best technique for vocabulary acquisition. *Estonian Papers in Applied Linguistics (Eesti rakenduslingvistika uhingu aastaraamat)* 4.121–138.

MONDRIA, JAN-ARJEN. 2003. The effects of inferring, verifying and memorizing on the retention of L2 word meanings. *Studies in Second Language Acquisition* 25.473–499.

——. 2007. Myths about vocabulary acquisition. *Babylonia* 2.63–68.

——, and SIEBRICH MONDRIA-DE VRIES. 1994. Efficiently memorizing words with the help of word cards and hand computer: Theory and applications. *System* 22.47 – 57.

MURPHY, KEVIN R., and CHARLES O. DAVIDSHOFER. 1998. *Psychological Testing: Principles and Applications*. Upper Saddle River, NJ, USA: Prentice Hall, 4th edition edition.

NATION, I. S. P. 2001. *Learning Vocabulary in Another Language*. Cambridge, UK: Cambridge University Press.

NATION, P. 2006. *Vocabulary: Second Language*, 448–454. Oxford: Elsevier, 2nd ed edition.

NATIONAL VIRTUAL TRANSLATION CENTER, 2007. Language learning difficulty for English speakers. `http://www.nvtc.gov/lotw/months/november/learningExpectations.html`. Accessed 19th Jan 2008.

NISHIGUCHI, KOICHI. 1994. *Kanji in Context Workbook Volume 1: Book 1*. Tokyo, Japan: Japan Times.

NUTTALL, CHRISTINE. 1996. *Teaching Reading Skills in a Foreign Languages*. London, UK: Heinemann English Language Teaching, 2nd edition.

PAGEL, VINCENT, KEVIN LENZO, and ALAN W. BLACK. 1998. Letter to sound rules for accented lexicon compression. In *Proceedings of the 5th International Conference on Spoken Language Processsing*, 252–255, Sydney, Australia.

PELLOM, BRYAN, and KADRI HACIOGLU. 2001. *Sonic: The University of Colorado continuous speech recognizer*. Technical report, Center for Spoken Language Research, University of Colorado, Boulder, USA.

PERVOUCHINE, VLADIMIR, HAIZHOU LI, and BO LIN. 2009. Transliteration alignment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, 136–144, Suntec, Singapore.

PINKER, STEVEN. 1994. *The Language Instinct*. New York, USA: Harper Perennial.

PLOUX, SABINE, and HYUNGSUK JI. 2003. A Model for Matching Semantic Maps between Languages (French / English, English / French). *Computational Linguistics* 29.155–178.

PONSODA, VICENTE, JULIO OLEA, MARIA SOLEDAD RODRIGUEZ, and JAVIER REVUELTA. 1999. The effects of test difficulty manipulation in computerized adaptive testing and self-adapted testing. *Applied Measurement in Education* 12.167–184.

PRICHARD, CALEB. 2008. Evaluating L2 readers' vocabulary strategies and dictionary use. *Reading in a Foreign Language* 20.216–231.

PULIDO, DIANA, and DAVID Z. HAMBRICK. 2008. The virtuous circle: Modeling individual differences in L2 reading and vocabulary. *Reading in a Foreign Language* 20.164–190.

REIPS, ULF-DIETRICH. 2002. Standards for internet-based experimenting. *Experimental Psychology* 49.243–256.

RICH, ELAINE. 1983. Users are individuals: individualising user models. *International Journal of Man-Machine Studies* 18.199–214.

ROGET, PETER MARK, and ROBERT L. CHAPMAN. 1977. *Roget's international thesaurus*. New York, USA: Harper & Row.

ROSCH, ELEANOR, CAROLYN B. MERVIS, WAYNE D. GRAY, DAVID M. JOHNSON, and PENY BOYES-BRAEM. 1976. Basic Objects in Natural Categories. *Cognitive Psychology* 8.382–439.

ROSEN, ERIC. 2003. Systematic irregularity in Japanese rendaku: How the grammar mediates patterned lexical exceptions. *Canadian Journal of Linguistics* 48.1–37.

SAITO, HIRAFUMI, HISASHI MASUDA, and MASAHIRO KAWAKAMI. 1998. Form and sound similarity effects in kanji recognition. *Reading and Writing* 10.323–357.

SALTON, GERARD, and CHRISTOPHER BUCKLEY. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24.513–523.

Sato, Satoshi, Suguru Matsuyoshi, and Yoshsuke Kondoh. 2008. Automatic assessment of Japanese text readability based on a textbook corpus. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco.

Sears, Richard, 2008. *Chinese Etymology Home Page*. Available at `http://www.chineseetymology.org/`. Accessed 10th Dec 2008.

Shannon, Claude Edward, and Warren Weaver. 1949. *The Mathematical Theory of Communication*. Urbana, USA: University of Illinois Press.

Shute, Valerie J., and Joseph Psotka. 1994. *Intelligent Tutoring Systems: Past, Present and Future*. Technical report, Human Resources Directorate, Manpower and Personnel Research Division, Brooks Air Force Base, Texas.

So, Sofumi. 2008. Review: Remembering the Kanji 1. *The Modern Language Journal* 92.663–664.

Taft, Marcus. 1991. *Reading and the Mental Lexicon*. London, UK: Lawrence Erlbaum Associates.

——, Ying Liu, and Xiaoping Zhu. 1999a. Morphemic processing in reading Chinese. In *Reading Chinese Script: A Cognitive Analysis*, ed. by Jian Wang, Albrecht W. Inhoff, and Hsuan-Chih Chen, 91–113. Mahwah, USA: Lawrence Erlbaum Associates.

——, Xiaoping Zhu, and Danling Peng. 1999b. Positional specificity of radicals in Chinese character recognition. *Journal of Memory and Language* 40.498–519.

Taghva, Kazem, Julie Borsack, and Allen Condit. 1994. An expert system for automatically correcting OCR output. In *Proceedings of the IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, 270–278, San Jose, California, USA.

Tanaka-Ishii, Kumiko, and Julian Godon. 2006. Kansuke: A kanji look-up system based on a few stroke prototype. In *Proceedings of 21st International Conference on Computer Processing of Oriental Languages*, Sentosa, Singapore.

Toyoda, Etsuko, and Yoji Hashimoto. 2002. Improving a placement test battery: What can test analysis programs tell us. *ASAA e-journal of Asian Linguistics and Language Teaching* 2.

Tozcu, Anjel, and James Coady. 2004. Successful learning of frequent vocabulary through CALL also benefits reading comprehension and speed. *Computer Assisted Language Learning* 17.473–495.

TSUJIMURA, NATSUKO (ed.) 1999. *The Handbook of Japanese Linguistics*. Great Britain: Blackwell Publishers Ltd.

URQUHART, SANDY, and CYRIL WEIR. 1998. *Reading in a Second Language: Process, Product and Practice*. New York, USA: Longman.

VANCE, TIMOTHY J. 1987. *An Introduction to Japanese Phonology*. New York, USA: SUNY Press.

VERMEER, ANNE. 2001. Breadth and depth of vocabulary in relation to L1/L2 acquisition and frequency of input. *Applied Psycholinguistics* 22.217–234.

WAGNER, ROBERT A., and MICHAEL J. FISCHER. 1974. The string-to-string correction problem. *Journal of the ACM* 21.168–173.

WALTERS, JODEE. 2006. Methods of teaching inferring meaning from context. *RELC Journal* 37.176–190.

WALZ, JOEL. 1990. The dictionary as a secondary source in language learning. *The French Review* 64.79–94.

WANG, LIWEI, YAN ZHANG, and JUFU FENG. 2005. On the euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.1–6.

WILLS, SEBASTIAN A., and DAVID J. C. MACKAY. 2006. DASHER – an efficient writing system for brain-computer interfaces? *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14.244–246.

WINSTEAD, CHRIS. 2006. Electronic kanji dictionary based on "Dasher". In *Proceedings of the 2006 IEEE Mountain Workshop on Adaptive and Learning Systems*, 144–148, Logan, Utah, USA.

——, and ATSUKO NEELY. 2007. Kanjiru: a software interface for visual kanji search. In *Proceedings of the Forth International Conference On Computer Assisted Systems For Teaching & Learning Japanese*, Honolulu, USA.

YAMAMOTO, KEN, and YOSHIHIRO YAMAZAKI. 2009. Network of two-chinese-character compound words in Japanese languages. *Physica A: Statistical Mechanics and its Applications* 388.2555–2560.

YAMASHITA, HIROKO, and YUKIKO MARU. 2000. Compositional features of kanji for effective instruction. *Journal of the Association of Teachers of Japanese* 34.159–178.

YANG, TAI-NING, and SHENG-DE WANG. 2001. A rotation invariant printed Chinese character recognition systems. *Pattern Recogntion Letters* 22.85–95.

YEH, SU-LING, and JING-LING LI. 2002. Role of structure and component in judgments of visual similarity of Chinese characters. *Journal of Experimental Psychology: Human Perception and Performance* 28.933–947.

YENCKEN, LARS, ZHIHUI JIN, and KUMIKO TANAKA-ISHII. 2007. Pinyomi - dictionary lookup via orthographic associations. In *Proceedings of the 10th Conference for the Pacific Association of Computational Linguists*, Melbourne, Australia.

ZHANG, DENGSHENG, and GUOJUN LU. 2003. Evaluation of similarity measurement for image retrieval. In *Proceedings of the 2003 IEEE International Conference on Neural Networks and Signal Processing*, volume 2, 928–931.

ZHANG, JIANNA JIAN, HOWARD J. HAMILTON, and NICK J. CERCONE. 1999. Learning English grapheme segmentation using the iterated version space algorithms. In *Proceedings of the 11th International Symposium on Methodologies for Intelligent Systems*, Warsaw, Poland.

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 *Proceedings of the SemaNet 2002 Workshop on Building and Using Semantic Networks*, Taipei, Taiwan.

——, and STERGOS D. AFANTENOS. 2007. Let's get the student into the driver's seat. In *Proceedings of the 7th International Symposium on Natural Language Processing*, Chonburi, Thailand.

——, and SLAVEN BILAC. 2004. Word lookup on the basis of associations: from an idea to a roadmap. In *Proceedings of the 20th International Conference on Computational Linguistics*, 89–95, Geneva, Switzerland.

——, 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 *Proceedings of the 5th Workshop on Multilingual Lexical Databases*, Grenoble, France.

Index