Predicting and Compensating for Lexicon Access Errors

Lars Yencken
NICTA Victoria Research Lab
Dept. of Computer Science and Software Engineering
The University of Melbourne
lars@yencken.org

Timothy Baldwin
NICTA Victoria Research Lab
Dept. of Computer Science and Software Engineering
The University of Melbourne
tb@ldwin.net

ABSTRACT

Learning a foreign language is a long, error-prone process, and much of a learner's time is effectively spent studying vocabulary. Many errors occur because words are only partly known, and this makes their mental storage and retrieval problematic. This paper describes how an intelligent interface may take advantage of the access structure of the mental lexicon to help predict the types of mistakes that learners make, and thus compensate for them. We give two examples, firstly a dictionary interface which uses search-by-similarity to circumvent the tip-of-the-tongue problem, and secondly an adaptive test generator which leverages user errors to generate plausible multiple-choice distractors.

Author Keywords

dictionary search, character similarity, adaptive vocabulary testing

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

General Terms

Experimentation, Human Factors, Languages

INTRODUCTION

One of the biggest problems language learners face is acquiring the vast amount of vocabulary they need in order to function adequately in their chosen language. For example, the core oral vocabulary in English is considered to be some 3000-5000 word families [18], equivalent to that believed necessary for comfortable reading of general texts [12]. Furthermore, word knowledge is not binary, but multi-faceted [18] and graded [14]. In practical terms, this means that only partially knowing a word is the norm for learners, leading to many errors in word access and use.

Access difficulties typically result in tip-of-the-tongue states, where limited aspects of the word in question are known. The original study of this phenomenon found that, despite difficulties recalling a word fully, subjects could nonetheless access part of a word's sound and form, for example a few letters or perhaps a syllable [5]. Fortunately, the information they do have access to is sufficient for many forms of search.

Most existing systems target the semantic relationships between words in order to aid dictionary navigation, with examples including: ViVA [19], a visual dictionary for learners and aphasia sufferers; WordNet Explorer [7], a visualisation of WordNet's semantic relationships; and dictionaries based on semagrams, such as the ANW dictionary of contemporary standard Dutch [17]. However, knowledge about the word's visual form is often overlooked. In Chinese and Japanese, where characters are normally typed by pronunciation, a learner may simply have no means to type in an unknown word, and must resort to slower and more error-prone form-based dictionary indexes.

The dual problem facing learners is lack of access to high quality tests which might aid their vocabulary development. Tests with limited scope such as flashcards are always available, but expert-constructed tests with wide scope are only available in limited circumstances due to their expense. Recent work has attempted to automatically generate specific question types, such as cloze questions [9], but the scope of these attempts remains too narrow to reasonably approximate proficiency tests.

This paper describes how recent work on visual similarity modelling for Japanese and Chinese [30] provides the means to tackle both problems. In particular, it contributes two novel and different applications for this error modelling: SimSearch, a spatial similarity search for Japanese kanji which can correct lookup errors, and Kanji Tester, a system for automatically generating proficiency tests modelled after the Japanese Language Proficiency Test (JLPT),[1] designed instead to elicit errors. Together, these systems show how modelling learner errors allows us to improve the usability of assistive systems, and thus to help language learners achieve their proficiency goals.

BACKGROUND

This paper focuses on how mental lexicon access structure can be leveraged to improve the accessibility of interfaces supporting language learning. We use as our target language

IUI'11, February 13–16, 2011, Palo Alto, California, USA.

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

Japanese, and for this reason, this section provides a brief overview of the Japanese writing system, before describing the current state of these interfaces.

The Japanese writing system

The Japanese writing system consists of three main scripts, the morpho-syllabic *kanji* script and the syllabic *hiragana* and *katakana* scripts. The syllabic scripts have transparent pronunciation and use a limited number of symbols comparable to an alphabet, so do not present significant difficulties to learners. Kanji however are more complex, due to their hierarchical visual structure, their sheer number, and their pronunciation, which is contextual. For example 行 is pronounced *i* in 行く [*iku*] "to go", but *kō* in 旅行 [*ryokō*] "travel". Psycholinguistic studies of reading confirm that in Japanese, phonology is dominantly computed at the word-level [27]. For learners encountering an unknown word, this means they have less sub-word information available to guide pronunciation than in other languages, and thus make more mistakes. Since every-day-use kanji number in the thousands, acquiring these kanji makes up a large part of learning to read and write in Japanese, and is a significant component of vocabulary learning.

Whereas alphabets allow approximate spellings to be typed in, kanji are typed in by their pronunciation. This means a learner *cannot simply type* a new word containing unknown kanji into a computer, but must first look up the unknown kanji in a dictionary via a graphical index. The traditional method of graphical indexing relies heavily on stroke counts, however learners often make mistakes in their counting, because strokes overlap and can be counter-intuitive in composition.

MODELLING LEXICON ACCESS ERRORS

In order to correct for user errors, we must first predict what form they will take. This section provides an overview of how existing work in visual word recognition informs models of user errors.

The Japanese mental lexicon

Models of visual word recognition provide useful insights into the nature of errors we would expect to occur during the recognition process. There is a wide range of known effects on access latency and error prevalence, including *semantic priming*, *word frequency* and *word superiority* [15], which any successful model must predict.

An important family of recognition models which predicts these effects is based on McClelland's multilevel interactive-activation framework [16], for which variants have been proposed for Chinese [21] and Japanese [10]. These models are hierarchical, and suggest that in kanji recognition, activation flows from features to strokes, strokes to components, components to whole-kanji characters, and finally from characters to words.

As well as activation, there are also inhibition effects. For example, if the letter *i* is strongly activated, it will have an inhibiting effect on its visual neighbour *l*, preventing it from being mistakenly activated in the next higher level. This applies equally at the word level: if presented with the word *wool*, its neighbour *wood* will be inhibited. This inhibition is presumed to prevent the reader from being overwhelmed by competing candidates.

Crucially, we only expect inhibition to occur for known words. For example, we expect the unknown word *epple* to evoke its neighbour *apple*, since *epple* is not yet in our mental lexicon. This is the principle we use for kanji search by similarity: if a learner encounters the unknown kanji 動, but knows its neighbour 働 [*hatara(ku)*] "work", the neighbour will be evoked and can be used as a query. With an appropriate model of similarity, the query can lead the learner to their desired kanji 動 [*ugo(ku)*] "movement".

From strokes to characters

In order to determine what features of characters make them visually similar to one another, earlier work [30] investigated a number of distance (or similarity) metrics between Japanese kanji. These metrics included cosine similarity of radical vector representations, the L_1 norm over rendered images of kanji across various fonts, tree edit distance over hierarchical kanji representations, and finally edit distance over kanji stroke sequences. Each metric was evaluated over a number of data sets, which included the judgements of novices, learners, native speakers and experts.

The most surprising outcome of these experiments was that normalized stroke edit distance performed best in determining high-similarity pairs, equivalently to the more expensive tree edit distance. This was unexpected, since its calculation requires firstly turning a two-dimensional kanji into a linear list of strokes, which would seem to be discarding useful positional information. Since this metric had the highest agreement with human judgments, we use it within this paper as the basis of our visual search.

From characters to pronunciation

Once kanji are perceived, aspects of their pronunciation aid further retrieval. However, in Japanese, pronunciation is only fully determined at the word level. For example the kanji 高 is pronounced *taka* in 高い [*takai*] "tall" but *kō* in 高原 [*kōgeN*] "plateau". This leads to the common situation where a new word is encountered, and each kanji in the word is *partly known*, but the pronunciation of the new word remains unknown and must be guessed. In order to recover from the types of errors learners (and native speakers) normally make, we need a model of plausible (mis)pronunciation. This section provides a brief overview of the pronunciation model developed for the FOKS dictionary interface.

Suppose the user enters a reading r and we must determine what word w they are trying to find. The probability that they are looking for w is then

$$\Pr(w|r) \propto \Pr(r|w)\Pr(w)$$

where $\Pr(w)$ can be approximated by corpus frequency and $\Pr(r|w)$ is our (mis)pronunciation model. Suppose w is made up of kanji $w_1 \dots w_n$. Then we can make the approximation:

$$\Pr(r_1 \dots r_n | w_1 \dots w_n) \approx \prod_{i=1}^{n} \Pr(r_i | w_i)$$

Each w_i is a kanji, and by aligning dictionary pronunciations of words to their written form, we can develop direct frequency estimates of $\Pr(r_i | w_i)$, including common word formation effects which subtly change the pronunciation of readings in compounds. This now allows us to both recognise incorrect readings, and also to generate the types of incorrect readings which learners might guess.

From errors to applications

So far, we have described existing means of predicting what types of graphical and phonetic confusion errors learners are most likely to make. What remains is to embed them into useful applications and user interfaces so that learners themselves can actually benefit from this modelling.

To this end, we propose two distinct applications for these models, which are described in detail in the remainder of this paper. The first, SimSearch, is an open source dictionary interface aimed at circumventing the tip-of-the-tongue problem in Japanese, by allowing lookup of kanji by visual similarity. The second, Kanji Tester, is an adaptive testing system which automatically generates multiple-choice vocabulary questions with linguistically motivated distractors. Both build on the models described above.

SimSearch: VISUAL LEXICON NAVIGATION

Overview

The broad idea of visual lexicon navigation is simple: a user queries the system with an entry which looks visually similar to the desired target, and is presented with candidates in the visual neighbourhood. If the target is not amongst these candidates, they click on the candidate most similar to the target as a subsequent query. A user who cannot immediately spot their target can thus iterate towards it in a hill-climbing manner. To explain the utility of SimSearch, we must firstly provide additional background into dictionary systems.

Dictionary search

Traditional paper dictionaries allow access to a word's meaning based on its orthographic form. For languages with alphabetic writing systems, a word's form is more or less transparently related to its pronunciation, so they can equally be considered to be indexed by pronunciation. The advent of electronic dictionaries has made this type of lookup faster, but has also allowed multiple indexing methods to be used on the same dictionary.

In Chinese and Japanese, dictionary search has been dominated by the input problem, where words containing unknown kanji have unknown pronunciation. This has led to efforts in improving traditional lookup-by-form, including the SKIP [8] and Kansuke [24] lookup methods. In parallel, the FOKS (Forgiving Online Kanji Search) dictionary interface [2] was developed to make input by pronunciation more robust in Japanese, where the pronunciation of each kanji character is contextual to the word it is used in. This paper's models of mispronunciation are based on an augmented version of the FOKS error models [30].

Utility of SimSearch

There are two distinct cases where lexicon navigation by visual similarity has utility for the user, one language independent and one related to particular difficulties with the Japanese and Chinese orthographies.

In the first case, a learner may have only partial knowledge of the character (or word) in question, as in the tip-of-the-tongue problem. In such cases, a partial visual form is often recalled. For example, in English the first and last letters of a word are often known [5]. This is enough to let a user formulate a visually similar word as a query, and seeing other candidates in the visual neighbourhood may either allow them to identify their target, or otherwise help them to recall more details about the target.

In the second case, a learner may be visually certain of the character (e.g. it may be written in front of them), but unable to easily input the character. This is a common problem for learners of Japanese and Chinese, but likewise for native speakers of both languages, since computer input methods are usually based on pronunciation.

The SimSearch interface

Our open source implementation of visual search, SimSearch,[2] provides a Japanese-English dictionary interface for finding unknown kanji characters. Firstly, a search box is provided, with instructions to enter a kanji character which looks similar to the desired target. Once the user enters a query, they are presented with a visual layout of matching candidates, as shown in Figure 1.

Although there are many possible layout algorithms, we use a naive method in this paper: the top k candidates are arranged into three concentric fixed-size tiers around the query kanji, so that similarity to the query is loosely indicated by visual proximity. Within each tier, kanji are shuffled and spaced evenly. The second and third tiers are reduced in opacity, so that the user's focus is drawn to higher-similarity results instead.

If the user sees their target, they click on it, or else they click on the next closest kanji to the target. The selected candidate then becomes centred as the root of a new query. If the candidate is the target, the user simply clicks on it again to view its translation. The user may cancel the search at any time by clicking on the cross at the bottom of the search panel, and may also navigate backwards and forwards through their immediate search history using arrows provided at the base of the panel.

Query and update model

Since each candidate displayed in response to a query serves equally well as a new query itself, users are expected to perform a natural hill-climbing search towards their target. The

[2] http://similarity.gakusha.info/

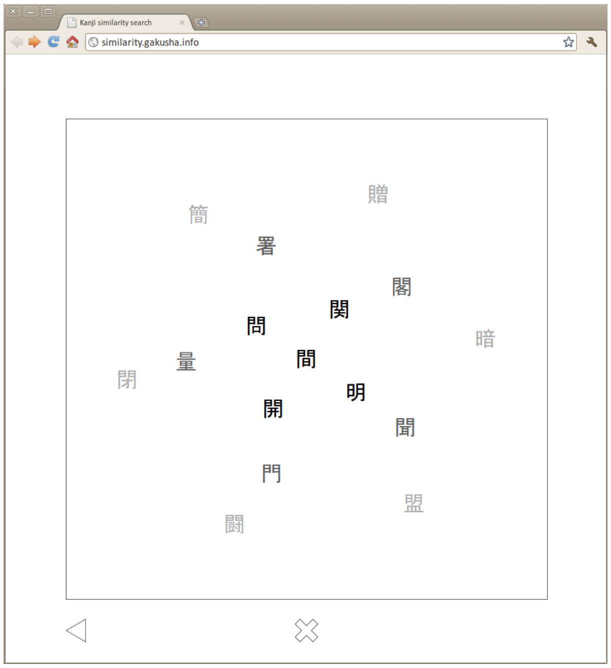


Figure 1. An example of visual search, rooted around the query kanji 間 [*aida*] "interval", with $k = 15$ neighbours displayed.

lack of distinction between query and result suggests a natural graph representation, and allows us to consider how we might best construct this graph to aid the users' search.

Consider each kanji as a vertex $s \in S$ in a directed graph, where edges $a \in A$ indicate confusable neighbours and are weighted by the likelihood of confusion. Successful queries are then paths through the graph, starting at the initial query point and ending at the kanji selected for translation. Each step is rooted at a query node and its candidate results are outgoing edges. The weights of these edges indicate their likelihood as candidates, and this is communicated to the user by visual proximity to the query kanji. However, the similarity models we use are known to be noisy; we need a way of gradually updating our similarity estimates to match actual human perception.

If we assume that at each step, the user only considers the candidates displayed to them when choosing their next action, then the search is a Markov Decision Process, and we can estimate the true long-run value of each query candidate using reinforcement learning techniques. We thus base our update model on Q-learning [26], an incremental update algorithm for solving Markov Decision Processes.[3]

Normally, learning algorithms focus on finding the optimal policy $\pi^* : S \to A$, which defines the best candidate to take at each state s. However, since SimSearch displays multiple candidates to the user, we instead are interested in the long-run value (i.e. across many user queries) of displaying a particular candidate. Our initial valuation of each candidate is given by:

$$Q_0(s,a) = \frac{\Pr(a)\phi(s,a)}{\sum_i \Pr(a_i)\phi(s)}$$

where ϕ calculates the similarity between two characters. In this paper we use normalized stroke edit distance as our metric, since it best matches human similarity judgements for high-similarity pairs [30].

Q-Learning provides the matching algorithm for updating this value based on our experience of user behaviour:

$$Q_{n+1}(s,a) = (1-\alpha_s)Q_n(s,a) + \alpha_s \left[r_a + \gamma \max_{a'} Q_n(s',a') \right]$$

where s' is the state action a leads to, r_a is the immediate reward, $\alpha_s \in [0,1]$ is the learning rate, and $\gamma \in [0,1]$ is a discount factor on future rewards from state s'.[4]

Since the reward r_a may be stochastic, we set:

$$r_a = \begin{cases} 1 & \text{if } a \text{ is the query target this session} \\ 0 & \text{otherwise} \end{cases}$$

This means each action has immediate expected reward matching its likelihood as a user's query target, i.e. $\mathbb{E}[R|a] \approx \Pr(a)$. The long run reward will converge to a mixture of its likelihood as a translation target combined with its likelihood to lead to one later.

Analysis

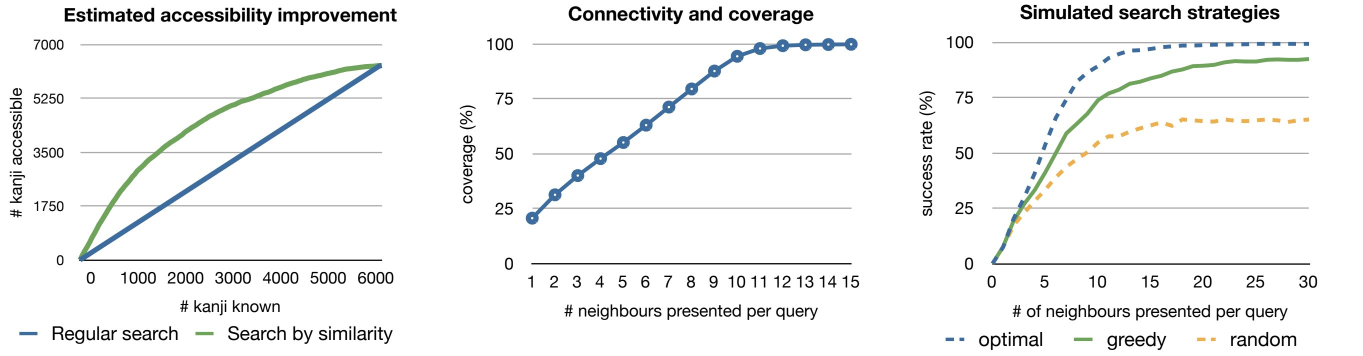
With such a system, there is a large number of potential issues one could consider, including: the similarity metric used; comparisons with alternative visual layouts; the method of achieving adaptivity; and speed in comparison with alternative interfaces. These are beyond the scope of this paper, and instead, we focus on two main points of analysis. Firstly, we use graph analysis to demonstrate the plausibility of visual search as a solution to the input problem for Japanese. Secondly, we use flashcard data as the basis for simulating query paths through the search graph.

On the plausibility of visual search

Firstly, we would like to know that visual search yields a plausible improvement in accessibility. This issue was discussed briefly in [30], and the theoretical argument is made as follows. Normally, a learner can only input kanji which they have studied, because only these kanji have known pronunciation. However, through similarity search, they can now also find unknown kanji within the visual neighbourhood of those that they know. Let us assume that learners study kanji in frequency order. Then if we have some realistic simulation of each kanji's neighbourhood, we can estimate the number of unknown but accessible kanji available to the learner at each stage.

[3] Other algorithms for solving Markov Decision Processes, such as Delayed Q-Learning [22], are known to converge faster, but do so via additional exploration of the solution space. In our case, this amounts to substantial sacrifice in early search utility, making such approaches inappropriate for our application.

[4] We used $\alpha(s) = \frac{1}{4+0.5u(s)}$ and $\gamma = 0.7$ in this work, where $u(s)$ is the number of times state s has already been updated.



(a) # of kanji reachable via SimSearch for varying levels of kanji knowledge, as compared to regular search.

(b) % of kanji reachable via SimSearch by at least one query, for varying numbers of candidates displayed.

(c) Success rate of simulated search for three different search strategies over 527 flashcard query–target pairs.

Figure 2. Evaluation of SimSearch

The key to an appropriate simulation is providing realistic neighbourhood sizes. This issue is independent and separate to the issue of how many candidates we display for each query; an individual user will either find two kanji to be similar or not, and the only useful candidates displayed will be those visually similar to the target. Local visual density near each kanji varies, in that some have a larger number of close neighbours than others. Furthermore, [30]'s distractor pool experiment suggests that the average neighbourhood size is low, between 1 and 3 neighbours. In this experiment we use a dynamic threshold, taking all neighbours within 0.95 of the first neighbour's similarity as being genuine, which yields an suitable average neighbourhood size ($\mu = 2.23$, $\sigma = 1.97$). The results of this simulation are shown in Figure 2(a).

On the number of candidates to display

In our visual layout, we limit the number of kanji displayed to a constant k. However, this search method does not guarantee that every kanji will be reachable by a similar neighbour; some may simply be visual outliers. Figure 2(b) indicates that although coverage is very limited with low k values, it increases linearly until saturation point at around $k = 11$, after which there is no gain in coverage with increasing numbers of neighbours displayed.

More broadly, we're interested in how successful realistic searches might be. In the absence of authentic user query and target pairs, we use the White Rabbit Flashcard data[5] as a plausible set of externally generated queries. Each flashcard in the set is based around a kanji, and contains one or two similar kanji for which it might be confused. We use the base kanji as a query, and consider each of its similar neighbours as targets. This yields 527 query–target pairs in total.

If the candidates returned from the query contain the target, we are done. However, if they do not, we simulate a strategy by which the user chooses the next candidate to continue the search. We naturally expect users to always choose the candidate most similar to their target character; we call this the *greedy* strategy. However, to the extent that humans agree on similarity judgements, our distance metric remains noisy. It is thus useful to consider strong bounds within which the search process might fall. As an upper bound, we include the *optimal* strategy which uses an oracle to determine the shortest path. For a lower bound, we assume *random*, undirected search on the user's behalf. For all strategies, we assume that the user loses interest after the fifth query, since in practice most searches seem to converge swiftly or not at all. These simulations lead to the success rates shown in Figure 2(c).

As the number of candidates increases, the graph becomes more connected. This should buoy success rates, provided the user is able to cope with the increasingly noisy presentation. Indeed, the optimal and greedy strategies monotonically improve with more candidates, whereas the two effects cancel one another out in the random case, which instead converges near 60% success. This might suggest that very large k values are desirable (e.g. $k = 30$ or even 100). However, pages containing more candidates will clearly incur more burden on the user, and may also cause them to miss their target amongst distractors, even if it is displayed.

Although modelling user burden is beyond the scope of this paper, psycholinguistic experiments yield some idea of its scope. [1] suggests an optimistic lower bound: a search task on Chinese characters (4, 8 or 12 items displayed) yielded a mean error rate of 7%, and a search rate of 30ms per item. In contrast, [28]'s study of similarity-based interference gives a pessimistic upper bound: subjects missed the target in a search task (24 items displayed) 33.6% of the time (from 3.6%) and had a reduced search rate of 57ms per item (from 36ms) when candidates shared both structure and radicals, though these figures were reduced for smaller displays.

Our candidates are designed to be similar to the query, and by extension some will be similar to the target kanji. An increased error rate is thus likely for this application. In an attempt to balance this factor with the desire for sufficient graph connectivity, we use $k = 15$ in our online system.

[5] `http://www.csse.unimeb.edu.au/~lljy/datasets/#whiterabbit`

Having discussed in detail how modelling lexicon access errors can lead to improved dictionary search, we now show how the same techniques can be used to develop randomized but authentic vocabulary tests.

KANJI TESTER: ADAPTIVE VOCABULARY TESTING

Overview

In our introduction, we suggested that the cost of test generation creates an artificial barrier limiting the ability of learners to self-test. Whilst SimSearch applied error modelling in a *corrective* manner, and thereby allowed intuitive search by form, this section describes how the Kanji Tester system instead uses them in a *generative* manner, creating randomized multiple-choice tests for learners. Before describing the system in detail, we motivate the need for automatic test generation.

The case for automatic test generation

Commonly used tests in second language learning range from simple flashcards to class tests, to accreditation of language competency. Whilst flashcards are widely available and can be used any time, tests with broader scope are far less available to learners, who must take them on an artificial schedule rather than when needed. This is largely due to two factors. Firstly, such tests require significant linguistic expertise to construct, making them expensive. Secondly, since each test is static it can only be used by each learner once, so the cost cannot be borne out over multiple testing sessions.

With the advent of Item Response Theory and Computer-Adaptive Testing, a workaround is presented: large item-banks can be constructed and sampled from for each test, thus reducing per-test costs [6]. As long as retests are few in number, a user is unlikely to encounter many test items more than once. Tests using these item-banks are cheaper to construct since each individual question may be re-used more often, however they lose validity if learners are allowed to re-test at will. For domains which permit it, generating tests automatically is a way around this fundamental problem.

A decade ago, automatic generation of human-like tests would have seemed inconceivable. Even today, the full range of questions used in modern proficiency tests such as TOEFL or JLPT is daunting in scope, and beyond the current state-of-the-art to generate. However, in recent times, many interesting classes of questions have been successfully generated. For example, [11] generates adaptive reading comprehension questions to accompany a passage of text. [23] and [9] generate cloze (or fill-in-the-gap) questions to test grammar and vocabulary. [4] generate vocabulary multiple-choice depth tests by using semantic relationships between words.

Although these systems address a wide variety of questions, they are nearly universally concerned with question generation in English. The remainder of this paper describes Kanji Tester, our attempt at fulfilling the need for automatic proficiency testing, modelled after the Japanese Language Proficiency Test. Creating questions is not in itself a problem; the main challenge is rather to generate distractors for each question which are difficult enough to be challenging for learners. The error rate induced by different question types will serve as our main metric for success.

Japanese Language Proficiency Test

The standard proficiency test for non-native Japanese learners is the Japanese Language Proficiency Test (JLPT), a family of tests pegged at several levels of proficiency and run by the Japan Foundation.[6] As of 2010, JLPT is offered biannually at levels N5 (easiest) to N1 (hardest). Level N1 is pegged roughly at the level of a native high-school graduate.

Since we are interested in automatic test generation for learners of Japanese, JLPT forms a natural long-term goal to aim for. However, generating questions of appropriate difficulty is a costly exercise typically undertaken by experts. Kanji Tester takes the approach of emulating only a subset of JLPT questions, those concerned with basic vocabulary knowledge. This is reasonable, since vocabulary is supportive of nearly all core language activities. For example, vocabulary size correlates better with success in reading than other measures, such as general reading or syntactic ability [13, 25].

JLPT has several weak points, but in general these only make it more suitable for emulating. For example, as an objective test it consists entirely of multiple-choice questions. This has the downside of only testing receptive knowledge, but the upside of simplifying scoring of answers, and of helping to focus the types of questions we generate. The JLPT is not a single monolithic test as is TOEFL, but instead limits learners somewhat arbitrarily to one of five levels to test against. This has the advantage of allowing us to focus on the more limited N4 and N5 levels instead of having to construct a full test of native-like Japanese proficiency.

Kanji Tester system

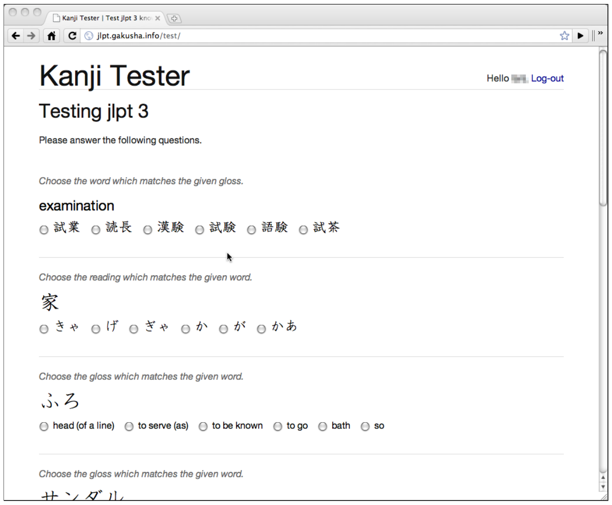
Although a full description of the Kanji Tester system is beyond the scope of this paper, this subsection provides a brief overview of the key points.

User perspective

When a user accesses Kanji Tester for the first time, they must sign up for an account. As part of sign-up, they choose a syllabus to study (one of the JLPT levels), specify their first language, and also any other languages they have studied. The user is then presented with a dashboard, and invited to take a test. They select a test length, from 10 to 50 questions, and click "Take a test". Kanji Tester then generates and displays a new test for them based on their syllabus and previous responses, as in Figure 3.

Each question in the test (a test *item*) is multiple-choice. Figure 4 compares an actual N4 question with a question generated using our method, in both cases requiring the user to select the correct pronunciation of a kanji word. In our example, the word is 自分 [*jibuN*] "self", but the user has chosen the incorrect *jipuN* as their answer. Once the the user has answered all questions and clicked "Check answers", their responses are scored and the incorrect responses highlighted.

[6] `http://www.jpf.go.jp/e/`



Kanji Tester | Test jlpt 3 kno

jlpt.gakusha.info/test/

Kanji Tester

Hello [redacted] Log-out

Testing jlpt 3

Please answer the following questions.

Choose the word which matches the given gloss.

examination

○ 試業 ○ 読長 ○ 漢験 ○ 試験 ○ 語験 ○ 試茶

Choose the reading which matches the given word.

家

○ きゃ ○ げ ○ ぎゃ ○ か ○ が ○ かあ

Choose the gloss which matches the given word.

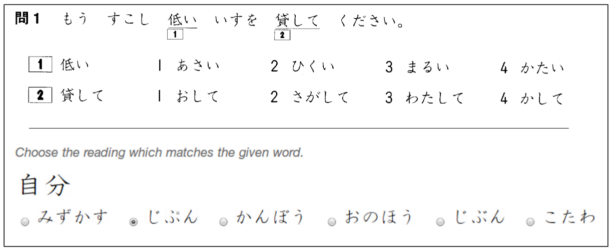
ふろ

○ head (of a line) ○ to serve (as) ○ to be known ○ to go ○ bath ○ so

Choose the gloss which matches the given word.

サンダル

Figure 3. An example test taken against the N4 syllabus. Each question is based on a single word or kanji randomly selected from the syllabus.



問１　もう　すこし　低い　いすを　貸して　ください。

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

Choose the reading which matches the given word.

自分

○ みずかす ● じぶん ○ かんぼう ○ おのほう ○ じぶん ○ こたわ

Figure 4. An N4 example question (above) contrasted with a question of the same type generated by Kanji Tester (below). Both questions ask the user to select the correct pronunciation for the given word form.

They can then click “Study mistakes”, and be taken to a revision page for the incorrect words, or return to the dashboard.

After several tests, the dashboard shows two graphs, one for words and one for kanji, each comparing the number of items seen over time, against the number which were correct when last tested. If learners are studying the mistakes they make or otherwise improving in proficiency, these graphs will converge over time; otherwise, if the learner’s knowledge stays roughly the same, they will diverge at a constant ratio. The dashboard also shows a user their score on the last test they took, and their long-run average for comparison.

System perspective

When a user first creates an account, they must choose a syllabus.[7] Upon doing so, a new user model is created for them. We limit our model to a grapheme confusion model for kanji recognition, and a phoneme confusion model for kanji pronunciation. Situating the models at the kanji level is useful since kanji are often the loci for misrecognition or misreading errors. We take after [20] in modelling each user individually, since we feel this makes the least assumptions about the types of mistake individual learners may make.

For each user u, their confusion models take the form of the distributions $P_u(reading|kanji)$ and $P_u(kanji'|kanji)$. These models focus on kanji within words, since kanji misreading or misrecognition errors are common amongst learners. Words without kanji avoid this whole class of errors, so for those we use a simpler form of question generation. In order to model word pronunciation at the kanji level, each syllabus’s word list needed to be grapheme–phoneme aligned; we used the unsupervised method of [29], manually correcting any alignment errors.

At this point we know little about the user, since they have not yet taken any tests, so we use as priors error models adapted from the FOKS dictionary interface [2, 30], combining many error types for reading confusion and using stroke edit distance to estimate visual confusion. After each test the user takes, these models will be updated with their responses, so as to adapt to their response patterns.

When the user initiates a new test, a number of words and kanji are chosen randomly as question seeds, according to their relative proportion in the user’s syllabus. Kanji Tester differentiates between question types based on whether distractors are based on form, reading or gloss. For each question, we randomly choose amongst applicable question types, generating a question of that type.

After the user has answered the question, we score the question and display to the user their results. At the same time, we update any error distributions used in question generation based on user responses. The only exceptions to this update process lie in questions with gloss distractors, where Kanji Tester currently lacks an intelligent means to choose distractors,[8] and in *control* questions.

To allow evaluation, the first test and every alternate test afterwards is set apart as a control test, where every question is generated in a simple rather than adaptive manner. Simple questions use the same pool of potential distractors as their adaptive counterparts, but ignore their estimated likelihood and instead sample *uniformly* from that pool. They are intended to serve as a strong benchmark for later comparison.

Update algorithm

When a question “successfully” provokes a user error, and it was generated using a confusion model, we want to update the original model to indicate that this error will be more likely in the future, thus increasing the chance we will elicit it again. Over many tests, this should increase the difficulty of the questions which are asked. We achieve this through our update algorithm, which we sketch out as follows.

Suppose the user is queried on a word w. For simplicity, we assume w is a kanji compound $w = k_1 \dots k_n$. The full distractor space O for the word is simply a product of kanji-level distractor spaces $O_1 \times \dots \times O_n$, but we only display a subset $D \subset O$ of the distractors to the user.

[7] Currently, JLPT levels N4 and N5 are available as syllabi.

[8] To generate distractor glosses, we simply choose glosses randomly from the JMDict dictionary (`http://www.csse.monash.edu.au/~jwb/jmdict.html`).

When the user chooses an incorrect answer $a \in D$, we assume it is because a seems more likely than all other distractors $D \setminus \{a\}$ by some margin ε. Our update rule attempts to enforce this ε margin in the posterior distribution for the distractors actually seen. This is expressed by the constraint:

$$\forall_{\{d:d\in D\setminus\{a\}\}} \Pr'(a|D) \geq \Pr(d|D) + \varepsilon \tag{1}$$

More specifically, we take the most likely distractor in $D \setminus \{a\}$, say $d_{\max}$, and define the posterior distribution as:

$$\begin{aligned} \Pr'(a|D) &= \max\{\Pr(a|D), \Pr(d_{\max}|D) + \varepsilon)\} \\ \Pr'(d_i|D) &= \alpha \Pr(d_i|D), \quad d_i \neq a \end{aligned} \tag{2}$$

where α is a normalising constant. When the ε margin already exists, the prior and posterior distributions are identical, but otherwise probability mass is moved from the other seen distractors to the one the user chose.[9]

In cases where this changes the word-level posterior probabilities, we must propagate the changes to the kanji-level model. For any distractor $d \in D$ with changed posterior probability, we define:

$$\Delta_d = \frac{\Pr'(d|D)}{\Pr(d|D)} \tag{3}$$

Since $d = d_1 \ldots d_n$, with $d_i \in D_i \subset O_i$, and our original word $w = k_1 \ldots k_n$, if we distribute this change equally between kanji-level distractors, then $\forall d_j \in O_j$:

$$\Pr'(d_j|k_j) = \begin{cases} (\Delta_d)^{\frac{1}{n}} \Pr(d_j|k_j) & \text{if } d_j \in D_j \\ \Pr(d_j|k_j) & \text{otherwise} \end{cases} \tag{4}$$

This final update rule is the basis for iterative updates to our per-user confusion models, and works equally well for both misrecognition and misreading models ($\Pr_u(kanji'|kanji)$ and $\Pr_u(reading|kanji)$).

Evaluation

The evaluation presented here is based on log data collected between November 2008 and April 2010.[10] Users were initially solicited through Google Adwords and a range of mailing lists in 2008, and then varied organically afterwards. In this section we examine who used the system during this period, how their test scores changed over the period of use, and the extent to which our adaptive tests were more difficult than their control counterparts.

User demographics

During this period, 366 users completed a minimum of one test, responding to 32945 questions in total (90.01 per user on average). In order to use the system, each user first had to enter their first and second language background. Although most users had English as their first language, we had users from 44 other language backgrounds, together accounting for nearly 50% of the user population. Note that this includes 23 people of Japanese first language background, who we exclude from further analysis in this paper.

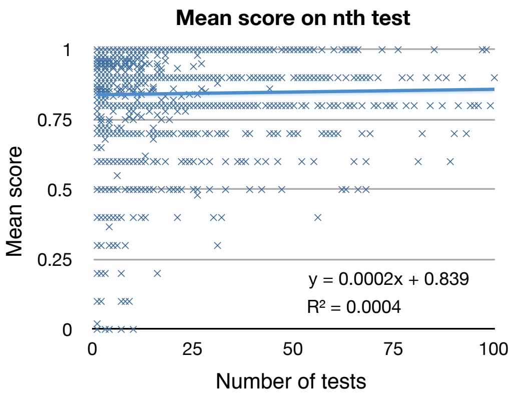


Figure 5. User scores on their *n*th test.

The average period of use is 13.2 days ($\sigma = 53.0$), indicating that many users returned to the system at least once over a period of a week or more. The average time between subsequent tests across all users was 2.71 days ($\sigma = 18.4$). The high standard deviations for both figures indicate a number of learners taking tests only a handful of times in quick succession and then not returning to the site, balanced out by learners who have returned periodically to retest themselves, even after long periods of time.

User performance and difficulty

Ideally, we would compare Kanji Tester performance against some known existing measure of proficiency. In the absence of such external data and a known user cohort, we are restricted to considering basic test difficulty for the self-selected users of the site, and to examining how their test responses changed over time.

The mean score across all tests was 86%, well above the required 60% pass mark for either JLPT level available to learners. Figure 5 shows the score for each user on their *n*th test. Note that a wide variety of scores exist on early tests, however after many retests the lower bound for user scores gradually increases. The figure also indicates no significant correlation between the number of tests and test score.

A better way to determine if users are actually improving is to consider words or kanji encountered multiple times, and to determine if the final time they tested on it was, on average, any more successful than the initial time. Figure 6, a histogram of mean pre–post differences for each user, shows the majority of the change is positive, i.e. the user improved on that word or kanji, with an average 6% improvement. However, since we test items randomly, most users rarely encounter the same item twice. This makes pre–post differences in accuracy sparse, and thus noisy, so the result is not statistically significant.

Question types and adaptivity

Table 1 gives the distribution of question types which were asked, split into their simple and adaptive variants. The gloss-based questions dominate, making up 58.2% of all questions.

[9] We use $\varepsilon = 0.2$ in this paper, with no attempt at tuning.

[10] This data is available for download at `http://www.csse.unimelb.edu.au/~lljy/datasets/`. Note that source code is also available upon request.

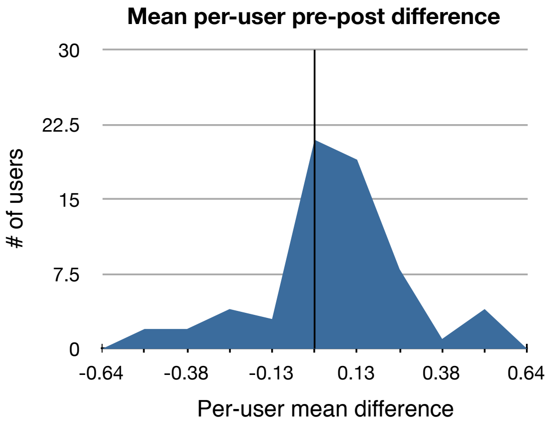


Figure 6. The per-user mean difference between initial-exposure accuracy and last-exposure accuracy, expressed as a histogram. ($\mu = 0.06, \sigma = 0.20$)

Type	Variant	# questions	
Reading	Simple	3631	9.3%
	Adaptive	4284	11.0%
Form	Simple	3802	9.7%
	Adaptive	4626	11.8%
Gloss	Simple	22734	58.2%
Total		39077	100.0%

Table 1. Question type, variant and number of questions answered.

This skew is due to the small number of kanji included in the early JLPT syllabi, where many words which are normally written with kanji are only learned using the syllabic scripts. This is an immediate blow to our attempts at adaptivity, since gloss-based questions are generated naïvely and do not adapt to user responses.

In order to determine whether our adaptive plugins generated more difficult questions than our control set, we can compare the error rates each plugin elicited. Figure 7 shows adaptive reading questions are more difficult than their simple counterparts (21.7% vs. 15.8% error rate), an effect statistically significant to the 99% level.[11] On the other hand, form based distractors are less effective in general, eliciting 10.5% error rate, and the adaptive variant is statistically indistinguishable from its control counterpart.

Discussion

All questions generated by the system provided 6 multiple-choice options to choose from; a naïve error rate for a random guess is thus 83.3%. Clearly none of our generated question types reach this level of error, and this could be indicative of a wide variety of effects, including: users making informed, knowledge-based guesses with better than chance odds; the possible existence of knowledge-independent answering strategies, such as guessing the centroid in a distractor set; and, most likely, many users already knowing most of the words or kanji being tested, and thus not having to guess very often.

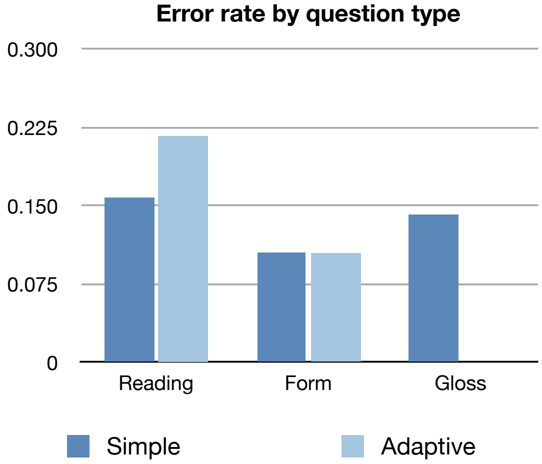


Figure 7. Error rates compared for simple and adaptive question types (higher is better).

Our form-based adaptive questions performed no better then their control counterparts, and this could be for several possible reasons. Firstly, the visual similarity metrics on which these models are based are known to be noisy, which limits their effectiveness. Secondly, they suffer from the *sparse neighbour problem*, where some kanji simply do not have many close visual neighbours in modern standard Japanese.

One solution to this problem could be to create a unified neighbourhood model for Chinese characters, including not only Japanese and Chinese characters, but also variants both archaic and modern. The resulting visual space would alleviate sparse-neighbourhood concerns due to its increased density, and would likewise allow stronger misrecognition questions to be constructed for learners of Chinese.

Based on Kanji Tester's usage, it clearly fulfills a need for quick testing and revision. However, its logs also indicate that some learners used it more like a drill, repeatedly testing themselves over short timeframes. Kanji Tester is less suited for this purpose, since it chooses items to test independently of previous tests. Drills which focus increasing vocabulary knowledge should instead repeatedly test users on their mistakes, and then retest later using a spaced repetition schedule, so as to maximize recall. This usage pattern suggests that the interface might be better divided into a study/drill part with these features, and an assessment part in the style of the current system.

Having mentioned data sparsity issues, post hoc analysis of user data might could be performed to determine the optimal method of grouping users so as to maximize useful error trends. [3] found significant differences in performance on a computer-adaptive test from learners of different first-language background, suggesting an appropriate direction for such work.

[11] As determined by a two-tailed Student's t-test.

CONCLUSION

In this paper, we have demonstrated how appropriate modelling of the mental lexicon allows assistive interfaces to better serve learners' needs. We applied existing error modelling to two novel interfaces: firstly, a visual search-by-similarity interface for Japanese kanji characters, which provides a plausible improvement to the accessibility of unknown characters; and secondly, an automatic test generation system, emulating the well-known Japanese Language Proficient Test. In addition to their careful modelling of user behaviour, both systems are designed to adapt to actual usage patterns, and to thereby increase their utility to learners as they are used.

Acknowledgements

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT centre of Excellence programme.

REFERENCES

1. G. A. Alvarez and P. Cavanagh. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological Science: a Journal of the American Psychological Society*, 15(2):106–11, Feb. 2004.
2. S. Bilac. Intelligent dictionary interface for learners of Japanese. Master's thesis, Tokyo Institute of Technology, Tokyo Institute of Technology, 2002.
3. A. Brown and N. Iwashita. Language background and item difficulty: the development of a computer-adaptive test of Japanese. *System*, 24(2):199–206, 1996.
4. J. C. Brown, G. A. Frishkoff, and M. Eskenazi. Automatic question generation for vocabulary assessment. In *HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 819–826, Vancouver, Canada, 2005.
5. R. Brown and D. McNeill. The "Tip of the Tongue" Phenomenon. *Journal of Verbal Learning and Verbal Behavior*, 5:325–337, 1966.
6. J. Bull and C. McKenna. *Blueprint for Computer-Assisted Assessment*. Routledge Falmer, London, UK, 2004.
7. C. Collins. WordNet Explorer: Applying Visualization Principles to Lexical Semantics. Technical report, 2006.
8. J. Halpern, editor. *The Kodansha Kanji Learner's Dictionary*. Kodansha International, Tokyo, Japan, 1999.
9. A. Hoshino, L. Huan, and H. Nakagawa. A Framework for Automatic Generation of Grammar and Vocabulary Questions. In *Proceedings of the WorldCALL 2008 Conference*, pages 179–182, Fukuoka, Japan, 2008.
10. T. Joyce. 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*, pages 1–12, Blacksburg, VA, USA, July 1999.
11. H. Kunichika, M. Urushima, T. Hirashima, and A. Takeuchi. Realizing Adaptive Questions and Answers for ICALL Systems. In *Proceeding of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pages 854–856, Amsterdam, Netherlands, 2005.
12. B. Laufer. How much lexis is necessary for reading comprehension? In P. Arnaud and H. Bejoint, editors, *Vocabulary and Applied Linguistics*, pages 126–132. Palgrave Macmillan, London, 1992.
13. B. Laufer. The lexical plight in second language reading; words you don't know, words you think you know, and words you can't guess. In J. Coady and T. Huckin, editors, *Second Language Vocabulary Acquisition*, Cambridge Applied Linguistics, pages 20–34. Cambridge University Press, Cambridge, UK, 1997.
14. B. Laufer and Z. Goldstein. Testing Vocabulary Knowledge: Size, Strength, and Computer Adaptiveness. *Language Learning*, 54(3):399–436, Sept. 2004.
15. S. J. Lupker. Visual word recognition: Theories and findings. In M. J. Snowling and C. Hulme, editors, *The Science of Reading: A Handbook*, chapter 3. Blackwell Publishing, Carlton, Australia, 2005.
16. J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception, Part 1: An account of basic findings. *Psychological Review*, 88:375–407, 1981.
17. F. Moerdijk, C. Tiberius, and J. Niestadt. Accessing the ANW dictionary. In *Proceedings of the 2008 Workshop on Cognitive Aspects of the Lexicon*, pages 18–24, Manchester, UK, 2008.
18. I. S. P. Nation. *Learning Vocabulary in Another Language*. Cambridge University Press, Cambridge, UK, 2001.
19. S. Nikolova, X. Ma, M. Tremaine, and P. Cook. Vocabulary Navigation Made Easier. In *Proceedings of the 2010 International Conference on Intelligent User Interfaces*, pages 361–364, Hong Kong, China, 2010.
20. E. Rich. Users are individuals: individualising user models. *International Journal of Man-Machine Studies*, 18:199–214, 1983.
21. H. Saito, H. Masuda, and M. Kawakami. Form and sound similarity effects in kanji recognition. *Reading and Writing*, 10(3 - 5):323–357, Oct. 1998.
22. A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 881–888, Pittsburgh, PA, USA, 2006.
23. E. Sumita, F. Sugaya, and S. Yamamoto. Measuring non-native speakers' proficiency of English by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, pages 61–68, Ann Arbor, USA, 2005.
24. K. Tanaka-Ishii and J. Godon. Kansuke: A kanji look-up system based on a few stroke prototype. In *Proceedings of 21st International Conference on Computer Processing of Oriental Languages*, Sentosa, Singapore, December 2006.
25. S. Urquhart and C. Weir. *Reading in a Second Language: Process, Product and Practice*. Longman, New York, USA, 1998.
26. C. J. C. H. Watkins and P. Dayan. Techincal Note: Q-Learning. *Machine Learning*, 8(3-4):279–292, May 1992.
27. T. N. Wydell, B. Butterworth, and K. Patterson. The inconsistency of consistency effects in reading: The case of Japanese kanji. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 21(5):1155–1168, 1995.
28. S.-L. Yeh and J.-L. Li. Role of structure and component in judgments of visual similarity of Chinese characters. *Journal of Experimental Psychology: Human Perception and Performance*, 28(4):933–947, 2002.
29. L. Yencken and T. Baldwin. Efficient grapheme-phoneme alignment for Japanese. In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 143–151, Sydney, Australia, 2005.
30. L. Yencken and T. Baldwin. Measuring and predicting orthographic associations: Modelling the similarity of Japanese kanji. In *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, UK, 2008.