How to Read Less and Know More: Approximate OCR for Thai

Doug Cooper
Southeast Asian Software Research Center
1617 Ratchaprarop Tower, 99 Soi Bunprarop, Ratchaprarop Road, Makasan, Bangkok, Thailand 10400
doug@nwg.nectec.or.th http://seasrc.th.net

Abstract

A large alphabet of similar letters and marks, wide and inconsistent variation in fonts and handwriting, and the absence of spaces between words all frustrate standard methods and applications for Thai-language OCR. We consider an alternative approach aimed at building information recognition and retrieval systems, rather than using OCR as a substitute for character-by-character data entry. Instead of trying to identify individual symbols, we define an ***approximation alphabet*** *of similar shapes and clusters, targeted to the predicted lower-bound accuracy of existing OCR. We test the effectiveness of approximation alphabets of 3, 7, 9, and 27 symbols for two tasks: discriminating between ambiguous input or queries (as from handwritten or pen-based input), and indexing scanned documents (as the basis of document-based IR systems).*

1. Introduction

Optical character recognition for Thai has been an active research area for many years. Success, however, has been difficult to achieve, and basic Thai OCR software is just beginning to appear on the market. Document retrieval systems for scanned legacy text have not even been attempted, and recognition of handwriting is not considered to be a realistic goal.

We believe that the common-sense target of Thai OCR — reading and identifying characters individually and accurately — is itself responsible for this slow progress. For legacy documents, such as typewriting, dot matrix, fax, newspaper, and similar text, it seems inevitable that the harder we try to achieve 100% accuracy, the less successful we will be at retrieving data.

Why the single-minded focus on reading letters? The appeal of character-by-character recognition derives partly from computing's historical development. For years, data storage and transfer were expensive in comparison to CPU cycles; this tended to concentrate interest in 'scan-and-discard' OCR systems. Only essential images were retained; turning text into its electronic equivalent, and not information retrieval per se, was the goal.

Characteristics of the Roman alphabet and European orthography helped make this practical. The alphabet has a relatively small set of distinctively designed letters, written on a single level. Add the fact that most text is segmented into individual words — making it amenable to effective methods of postprocessing and error correction — and it is not surprising that English and similar languages have enjoyed high-accuracy OCR.

This is not the case for Thai, which has a large alphabet of minimally differentiated symbols, written in *clusters* on four vertical levels, without space between words. Even given clean text, we take it for granted that humans cannot distinguish between many similar Thai letters and marks in isolation. In trying to out-do humans in reading letters, traditional OCR succeeds primarily in introducing errors that make indexing and finding words more difficult.

We take the position that for Southeast Asian writing systems like Thai (and including Lao, Burmese, and Khmer), accurate, letter-by-letter OCR is neither particularly likely, nor necessarily desirable. Instead of information reproduction, information *recognition and retrieval* should be our primary aims. To paraphrase [1], we want to know how much OCR-based IR can be done without the C or the R, and with as weak an O as possible.

This may sound like sour grapes; in effect, we are saying that because we can't do it well, we probably didn't want to do it in the first place. However, we claim that the very orthographic features that make Thai OCR so difficult have caused spelling and letter design to evolve in a manner that makes shape-based approximate recognition systems not just practical, but actually superior to standard OCR for IR and written or pen-based input.

Rather than trying to improve Thai OCR's upper bounds, we propose that predicting *lower* bounds for character-by-character OCR is possible — ฆ *and* บ or ท *and* ฑ may be indistinguishable, but ฆ *or* บ can always be told from ท *or* ฑ — and that this kind of difference will let us locate words. Instead of trying to discriminate between all characters, we settle for using *approximation alphabets* — OCR output alphabets that provide a many-to-one mapping between input clusters and output letters.

In this paper, we consider four approximation alphabets, reducing Thai's 70-odd symbols to between 3 and 27 letters, and test them as the basis of two distinct applications (see figure 1):

- disambiguating input (as from handwriting or pen-based input systems), and
- building IR systems for scanned documents (the approximation is used to index and retrieve original scanned images).

Test data include potential queries (eg. 45,648 personal names, 425 tax-related 'content' words, dictionary head and compound word lists, etc.) and text (the 1-megabyte Thai Tax Code, a 2-megabyte corpus). Note that because there are no existing Thai-language IR systems beyond rudimentary full text search/lexical match — no standard query or data sets, and certainly nothing based on OCR — we do not report on relative performance at this time. Instead, this paper tests the effectiveness of our approach in rendering certain IR problems tractable, and shows where and how the technique is best applied.

SIGIR 97 Philadelphia PA, USA

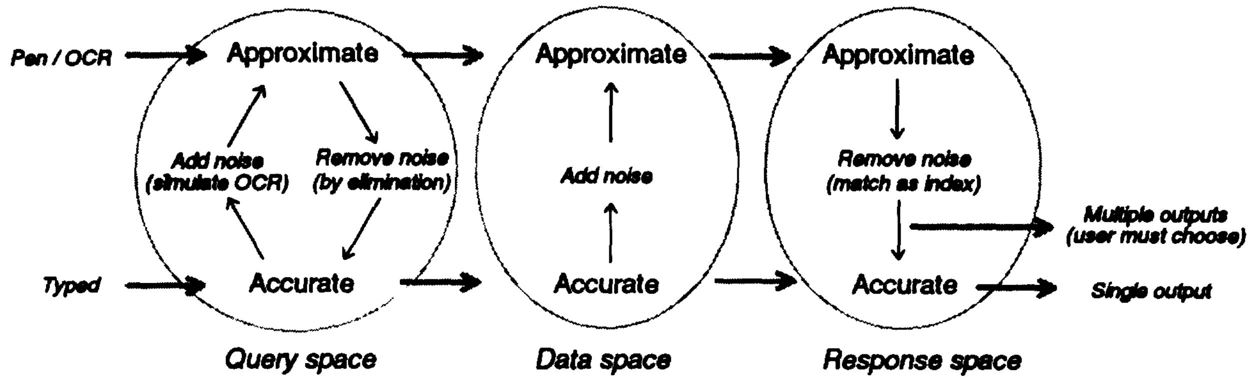


Figure 1 Both queries and data may be accurate or approximate. When approximate queries fall within a specific domain (eg. city names), it may be possible to disambiguate them in query space by elimination. Or, we may wish to add the same kind of noise to accurate data and hope for a unique (and correct) match. When queries are accurate but data are not, the process is reversed — we add noise to the query, then 'remove noise' from the response by using any match as an index.

2. Background

Early work on Thai OCR is described in [2,3]. [4,5] survey some of the problems that have persisted. Recent studies have focused almost exclusively on using neural networks to improve individual character recognition [6]. For example, [7] describes a system trained to recognize a specific set of fonts and print sizes, and provides detailed performance statistics.

Accuracy has not significantly improved over the years. Both of the two existing commercial systems (ThaiOCR 1.5 from Atrium Software, and ArnThai 1.0 from NECTEC/ThaiSoft) are sensitive to input text quality and font choice, and their best claimed performance is well below Roman-alphabet OCR.

We discussed the underlying reasons for this in [8]. In essence, the primary distinguishing characteristics of similar letterforms tend to disappear in typical font designs. Although subtler secondary features may help disambiguate letters, many letter pairs or triples are extremely difficult to distinguish from one another in isolation, or when accompanied by even minimal noise.

For instance, in these pairs — ข ฃ / ฎ ฏ — the tiny notches in the 'neck' of the second letter and the 'tail' of the fourth are too small to be distinguished by human or machine, native or otherwise. Noise, dropout, and poor font design take a toll as well: ป ('p') and บ่ ('b' plus a tone mark) converge rapidly.

Difficulty also occurs when alternative fonts appear. Below, we have the same pair of letters in three fonts; the circled letters should be as distinct as the first two, but are nearly identical:

In [9], we proposed that for an alphabet like Thai to have survived, the information content of individual letters and marks must be small, and the Hamming distance between words relatively large — in other words, if two letters look alike, the words they are in will probably look different. This is justified both by an analysis of the historical development of Thai orthography, and by computer investigation of the lexicon. We showed that this proposition survives the test; in general (typically 98% or better), a single word serves as the establishing context for minimally distinguished letters and marks.

Unfortunately, printed Thai is not segmented into individual words, and accurate segmentation even under ideal conditions is not an easy matter (see, for example, [10]). Many problems are similar to OCR for hand-written English; both in distinguishing words, and in a problem (vertical segmentation) equivalent to segmenting individual Roman letters (eg. [11, 12]).

Thus, the fact that errors persist even with revised algorithms and techniques appears to be an inherent consequence of Thai orthography, and high-quality Thai OCR still shares many of the problems of noisy English OCR. Methods used for correcting and retrieving information from noisy text are quite interesting [13, 14, 15], but the methods they rely on to improve performance — stemming, various approaches to spelling correction, etc. — cannot be readily applied to non-segmented Thai text.

In recent years, an alternative approach to the problem of low-quality letter-by-letter OCR has been to consider overall word shapes. Research has followed two basic paths: using image data from the documents themselves to search for specific words [1], and attempting to determine document content by simulating the shape of standard classification terms [16, 17].

Scrutinizing individual letter shapes has also been discussed. [18] looks at this in the context of easily recognizable aspects of handwritten data; primarily ascenders and descenders, and reports on distributions across a variety of lexicons, and [17] reduces the English alphabet to 7 shapes to do very fast scanning and categorization of large amounts of text. Probably the closest work to ours is reported in [19]; characters are coded by shape and disambiguated into specific letters where possible, then known letters are used as templates for 'recognizing' the remainder. Again, these techniques rely heavily on word-segmented data sets, and are not easily applicable to Thai text.

3. Specific Thai and Central SEA Issues

The writing systems of central Southeast Asia are all derived from the southern Indian Grantha script; based, in turn, on the ancient Indian Bhrami. Thus, even though the spoken languages of Thai/Lao, Burmese, and Khmer come from quite distinct language families, their writing systems present common problems for OCR. Most of these problems are represented in Thai:

- a large alphabet (forty-two consonants in common use, along with fifteen vowel symbols, six tone and other marks, and another half-dozen or so assorted letters and signs),
- letters and marks that are stacked in clusters, as in figure 2,
- lack of spaces between words.

Unlike the Roman alphabet, which is nearly unchanged in over a millennium, Thai has undergone frequent alterations since the core alphabet was borrowed from cursive Khmer in the late 13th century. Many letterforms have either converged to acquire similar shapes, which are differentiated only by the orientation of small features, or diverged, by the addition of tiny notches and tails, to create new letters for expressing the 'foreign' sounds of loanwords. This is the primary reason that many letters are difficult to differentiate in isolation, even in ordinary printing fonts.

Clustering and lack of word segmentation present problems for information retrieval as well. These include:

— a tendency to errors that involve missing characters.

For instance, a mangled cluster can create a gap of two characters in the output stream (as well as inserting a third, incorrect character). Allowing for such large gaps plus wildcards tends to make approximate searches blow up, especially because of . . .

— the absence of word boundaries, which make it easy to incorrectly match on adjacent strings.

These make the basic application of indexing large text databases also quite hard. Even with perfect data, segmentation is unreliable, especially in the presence of unknown words (like names and technical loanwords). Indexing systems for unsegmented text, in turn, are sensitive to spelling errors and missing letters, as above. In fact, one of the reasons that we find approximation so attractive is that it lets us use alternative methods (signature files, n-grams) to build indices.

At the same time, Thai grammar and orthography have a number of compensating characteristics. First, Thai is an isolating language, in which sentences consist of sequences of free morphemes or words. Higher-level semantic or grammatical nuances, such as tense, are shown by the insertion of additional words, rather than by the change of existing words. This does away with any need for stemming of any sort, and greatly simplifies the task of finding terms.

Second, Thai's very large alphabet, and the diverse national origins of its vocabulary, tend to result in a fairly large Hamming distance between words. As we discussed in [9], this is especially pronounced in the case of letters with similar appearance.

Third, Thai has a relatively small headword list of roughly 10,000 words, give or take a few thousand. This should not imply that Thai is not as expressive as any other language; rather, like many Asian languages, Thai relies on compound sequences, rather than neologisms, to express new concepts. This makes words longer, and easier to spot by overall appearance.

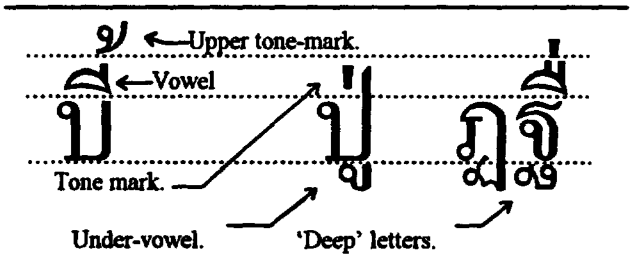


Figure 2 The Thai alphabet presents most characteristic problems of central Southeast Asian writing systems, including a large alphabet, stacked characters, letters and marks, and minimal differentation between letterforms. However, ligatures and context-dependent letter shapes — a particularly serious problem for Khmer and Burmese — are rare. The neatly placed tone marks shown here cannot be relied on, incidentally — many fonts freely shift marks between the top and second rows, and marks and vowels may overlap.

Fourth, even when modern loanwords are used, they almost invariably rely on peculiar spellings — both in choice and sequence of characters — that clearly stand out from 'native' Thai. Silent letters (marked with a special character) are frequently added for no other purpose than to designate the word as foreign.

Together, these characteristics make it likely that search strings are going to be relatively long, include a wide distribution of letters, and will consist of two or more uninflected words in fixed positions — ideal for working with a approximate index.

4. Approximation Alphabets

We define an approximation alphabet as a unique, several-to-one mapping between the letters (or in some cases, combinations of letters and/or marks) of a real alphabet, and the set of symbols we use for indexing and lookup. The salient features of an approximation alphabet are:

— within the limits of the approximation, OCR should be 100% correct, and

— simulation of the approximation must match the real thing.

These conditions guarantee that any IR system's recall — the percentage of relevant terms that are retrieved — will always be 100%. Naturally, there is a tradeoff, because precision, or percentage of hits that are relevant, will be lower than 100%. Thus, approximation may bring back irrelevant information, but it will never overlook useful data.

Note that the typical approach in commercial OCR systems — marking an unrecognized character with a ~ — is not sufficient. We must know in advance which characters or combinations will not be recognized exactly, and how they will be marked.

The simplest approximation alphabet we investigate has just three symbols (plus a space): one represents any main-row characters, while the others represent any sub- or superscripts. For example, each of these is 'recognized' as a single letter plus a sub- or superscript (even though in some cases, what appears to be a script is actually an integral part of the letter):

จุ ฐ   ฎ ภู   ล่ ส   ขี้ ป

An intermediate alphabet distinguishes gross features of ordinary full-size letters (but not sub/superscript details). For Thai, the orientation of concavity — the direction in which letters open and close — is a key indicator

กถภ   นมบ   วง   อธ

The most complex approximation alphabet is similar to the real alphabet, but consistently indistinguishable characters are merged. Each of these groups of distinct letters and clusters might be treated as a single approximation letter:

ข ช ซ   บ่ ป ป   ฎ ฏ   กี กี้ กิ่ ก้

For our tests, we defined four approximate alphabets:

Set 1 — zone (3 letters). The most basic system only sees letters, superscripts, and subscripts. There are four outcomes:

— ก น า . . . : recognized as generic letters.

— ดี ดี้ ฝ . . . : letter plus superscript.

— ดุ ฎ ฐ . . . : letter plus subscript.

— ดุ้ ปู . . . : letter plus superscript plus subscript.

Certain letters cross zone boundaries, and are implicitly assumed to include either superscripts (ป ฟ) or subscripts (ฎ ฏ).

Set 2 — zone and orientation (7 letters). We take easily recognized features into account as well:

- ก ศ ท . . . : open on bottom.
- น ม ข . . . : open on top.
- ณ ฌ ฒ . . . : straight sides, 3 verticals, wide.
- ง จ ธ ร ว ฉ อ ฮ ช ซ ย ะ ด ต . . . : all others.
- เ แ า ๆ . . . : exceptionally thin.
- ุ ู : subscripts.
- ่ ้ ๊ . . . : superscripts.

Again, certain letters are assumed to include either sub- or superscripts. เ and แ can are distinct and recognizable because the sequence เ+เ never occurs.

Set 3 — zone, orientation, and simple features (9 letters). Similar to set 2, this group detects easily spotted attributes in the 'all others' group.

- ก ศ ท . . . : open on bottom.
- น ม ข . . . : open on top.
- ณ ฌ ฒ . . . : straight sides, three verticals.
- ง จ ธ ร ว ฉ อ ฮ ะ . . . : curves, open left.
- ช ซ ย : curves, open or cleft top.
- ด ต : curves, open bottom.
- เ แ า ๆ . . . : exceptionally thin.
- ุ ู : subscripts.
- ่ ้ ๊ . . . : superscripts.

Set 4 — empirical errors (27 letters) Finally, set 4 starts with the regular alphabet, then merges potentially ambiguous letters and combinations, based on current Thai OCR software. The samples below show how grouping decisions are made; the actual groups are shown with a selection of fonts in Appendix 1.

- ช ซ / ท ฑ / ม ฆ : necks indistinguishable.
- ก ภ ถ : heads often broken or dropped.
- ณ ฌ : knots dropped or indistinguishable.
- ค ศ : notch indistinguishable.
- ด ค / ผ พ / ฝ ฟ : head orientation indistinguishable.
- อ ฮ / ด ต / ค ฅ : tail indistinguishable.
- ป่ ป : letter / letter plus tone mark.
- ผ ฝ พ ฟ ฬ : error-prone depending on font.
- ุ ู : subscripts indistinguishable.
- ่ ้ ๊ . . . : superscripts indistinguishable.

5. Test Data

Our tests involved three basic data sets: words, queries, and corpora. Words are the shortest meaningful units of Thai, as taken primarily from dictionary headword lists. Queries, in contrast, are both natural (names) and constructed (compound words, or pairs of words). The corpora are large text samples: a one-megabyte single-subject sample (the Thai tax code), and a two-megabyte collection of short selections. Figures in parentheses indicate the average number of characters per word or line, not counting blanks. All lists are of unique terms.

Full names A list of 45,648 (15.77) full Thai names, taken from the 1996 university entrance examination pass list (see [20]).

Last names Same source, 36,977 (9.32) last names.

Villages and provinces 13,465 (19.0) locations: 10,625 (11.44) village and 76 (7.62) province names [21].

'Standard' headwords Essentially the complete wordlist of 17,986 (5.41) terms described in the 'official' Ratchabandit dictionary, including all headwords plus all combined forms with potentially ambiguous pronunciation (usually words with historical roots in Pali/Sanskrit).

Haas headwords 5,941 (4.68) headwords [22].

Haas compounds 11,653 (8.33) subheads from the above.

Haas 'phrasewords' 541 (11.35) entries; all second-level subheads marked as nouns or verbs. These are essentially longer compound words.

Tax code About 1 megabyte (101.28, 10,224 lines). The full text of the 1995 Thai Revenue Code as taken from the HTML files provided on [23]. Lines were preserved as in the original file.

Tax code TOC compound and 'head' words The table of contents from [23] has 1,102 entries. We broke these into 425 (7.44) distinct content compounds (eg. 'income') and 361 (4.40) others (all single words, mostly function terms).

Tax code queries After removing all 'head' words from the tax code TOC, we generated all 712 (15.72) distinct two-content-word windows (eg. 'income' and 'foreign.').

Large text corpus About 2 megabytes (50,650 lines, 37.72 chars/line). Essentially a random sample from a variety of sources, with all non-Thai characters removed. The original sentence structure was preserved; however, all unambiguous breakpoints (numbers, punctuation, foreign letters) were considered to mark sentence breakpoints.

6. Methodology and Results

We are interested in two distinct questions. The first involves approximate queries and exact data (eg. a known query list of names or places), and the second involves exact queries and approximate data (eg. scanned text).

We present both raw results (tables 1-2) and graphical interpretation (graphs 1-3). Because of the shotgun approach taken in building query and data sets, we do not show recall/precision calculations; we feel they might be misleading given that we *a*) assume ideal coding, *b*) exhaustively test word lists, and *c*) count hits by words or sentences, rather than by pages or documents. Once again, we note that there are no existing Thai IR systems to serve as the basis for comparison, and that our main goal is to survey the applicability of our techniques.

6.1 Approximate query / exact data (Table 1, Graph 1) In the first case, we assume that the means we are using to obtain query terms is imprecise; eg. they are obtained through either traditional OCR or pen-based input. At the same time, we have a well-defined universe of possible queries, such as place names, personal names, or ordinary phrases. Our goal is to see if an approximate query can be disambiguated: correctly translated, by elimination, into its exact form. A simple text search or index lookup then completes the process.

For example, suppose that our list of potential queries consists of the following items, approximated as u (open at top) or n (open at bottom). In the example below, approximate input items 1 and 2 can be distinguished from the rest of the list, while items 3 and 4 have one correct and one incorrect match apiece. Note that word length comes into play as well.

1)	นม	un
2)	ถนน	nuu
3)	ทน	nu
4)	กน	nu

We assume that all list items are unique, and test our ability to disambiguate by:

— Producing a test version of the query list for each approximation alphabet.

— Approximating, in turn, each word on the original list. These serve as our query terms.

— Counting the number of matches for each approximate query.

Ideally, the approximate query will match just one 'approximated' data item. In practice, though, there are many applications in which letting the user choose from a few alternatives is reasonable. These range from machine OCR for automated mail routing (a human operator must confirm that all or part of an address has been read correctly), to using hand-held pen input devices for searching databases of names or specialized data.

We tested all four approximation alphabets against a range of potential query lists, including dictionary headwords and compounds, personal and place names, and 'constructed' queries (brief phrases taken from the Thai tax code).

We would anticipate that the best results would be found in lists that were short, and contained relatively long words, with a correspondingly high variation in word length. Indeed, the best performance came from such a list: names of the 76 Thai provinces. 69.7% of these could be uniquely identified on the basis of approximation alphabet 1, which only detected the presence of letters, subscripts and superscripts. Allowing two matches (one incorrect) increased this percentage to 80.2%, and permitting three (two wrong matches) raised the total to 81.5%.

In contrast, very long lists of relatively short words, such as dictionary headword lists, do not fare well. For example, alphabet 1 correctly identified only 3.9% of the list of 17,986 'standard' dictionary headwords. Even alphabet 4, which distinguished 27 different groups of letters, found distinct matches for just 74% of the full list. Because word length follows a more-or-less normal distribution, and because common words tend to be close to the average length (about five letters), this implies that approximation is not appropriate for dictionary lookup.

The best practical applications, even under poor conditions, are found when queries are longer than 10 characters. For example, we constructed 712 two-word queries, average length 15.72 characters, by extracting content words from the Tax Code's chapter headings. The crudest approximation alphabet let us correctly identify the query 76.1% of the time, while sets 2 and 3 were both 99.7% accurate, and set 4 reached 100%. Performance on full names (45,648 items, average 15.77 characters) were somewhat lower for alphabet 1, but were 99.9% accurate on alphabet sets 2, 3, and 4. Long compounds (541 items, average 11.35 characters) fared similarly: somewhat lower for alphabet 1, but 98.5% to 99.6% accurate for alphabet sets 2, 3, and 4.

Unexpectedly, the longest average query (village and province names, 13,465 items, 19.0 average characters) was slightly poorer: 95.5%, 97.4% and 99.3% for sets 2, 3, and 4. On investigation, this turned out to be the result of ambiguity between village names (average length 11.44 characters) *within* the large provinces, exacerbated by common substrings (equivalent to 'ville'). We suspect that a single extra approximate character — the zip code's last digit — might make a significant difference.

6.2 Exact query / approximate data (Table 2, Graphs 2, 3) In the second case, we assume that the data are approximate, having been obtained through OCR. Queries may be either exact or inexact; an exact query is intentionally approximated in the same manner as the data set.

Our test set is based on real data — the one-megabyte Thai Tax Code, and a two-megabyte corpus of randomly selected Thai texts. We used these data to simulate OCR according to approximation alphabet sets 2, 3, and 4, then treated these simulations as indexing the original data sentence-by-sentence. All unambiguous breakpoints were considered to indicate sentence boundaries.

Our methodology is similar to that described above:

— Produce test versions of the query lists and data sets for each approximation alphabet.

— Seek each approximate query in the approximate test data.

— Seek each appearance of the actual query in the actual test data. Two-word queries were required to match in order, in a single sentence, but with any number of characters in between.

Eleven of the twelve query sets (we excluded the standard dictionary headword list) were tested against the Tax Code, and five (the Haas phrases, province names, Tax Code compounds, Tax Code 2-word queries, and village/province name combinations) were tested against the two-megabyte text corpus.

Again, the approximated query should only match approximate test items that correctly 'index' the actual query to the actual text. As previously, there are situations in which a few false hits are acceptable, particularly if *a*) all correct hits are guaranteed to be found, and *b*) very large amounts of data are involved.

A good example of this is found in our search of the Thai Tax Code for the last names of entering University freshmen (which sometimes consist of ordinary words, and often involve references to money or wealth). Consider the results of test alphabet 4. Of 36,977 names, some 36,063 (97.5%) were *correctly* not located. Of the remainder, 31.3% were identified properly, an additional 21.9% had no more than one false hit, and 7.9% were incorrectly found no more than twice. Fewer than 1% of all the names returned more than two incorrect entries.

We noted similarly high precision in searching the text corpus for actual phrases. Using alphabet set 4, the Haas phrases (541 items) and Tax Code 2-word queries (712 items) matched exactly 165 (92.1%) and 135 (93.3%) of the time. Under the least favorable circumstances of alphabet 2 — Thai reduced to just seven letters — the search returned no more than two false hits 72.2% and 72.8% of the time, respectively.

7. Discussion and Future Work

The poor performance of traditional OCR for Thai has discouraged development of document management and IR systems, and diverted attention from less-precise forms of input. The results presented here show that an alternative approach, based on approximate rather than exact recognition, provides a practical basis for Thai-language IR. Because the approximation alphabets simulated are targeted at lower-bound OCR accuracy, we expect relatively robust performance even in the face of degraded input.

As we have seen, there are two distinct applications for approximation. In the first case, we obtain queries or data items through imprecise means, such as pen-based or handwritten input, and must decide exactly what the query or data item was. Although we are choosing from a restricted lexicon, that lexicon may be very large — performance on a 45K-plus list of names was 99.9% correct, even with a 7-letter alphabet.

Overall, our tests indicate that high performance can be expected for two-word inputs under any circumstances, but is more dependent on the approximation alphabet used for shorter terms. We feel that the test data are realistic, and point the way to practical applications — for appropriate tasks, pen-based or handwritten input for Thai may even leapfrog traditional OCR, rather than lagging at the usual respectful distance to the rear.

The second application involves using approximation to index scanned text. Queries are intentionally degraded to the same level as the text; all relevant entries will always be returned, but precision may be less than ideal.

The benefit of a longer approximation alphabet in rejecting false matches was clear across the board. While not all legacy data will meet the minimum guarantees of the 27-letter alphabet 4, we feel that its requirements are loose enough for most printed documents. For example, there is almost no Thai literature in usable electronic form; a database of page images that could be searched for example usage of idioms, elaborate expressions, and other multiple-word features would be of tremendous benefit to corpus-based lexicography, and is within reach.

For conventional commercial applications, such as finding personal or place names in legacy business documents, our results indicate that even the 7-letter set is serviceable. Typewritten or copied documents are probably within necessary bounds of accuracy, and preliminary experimentation with fax is promising.

Continued research is focused on the following:

- *Build test sets.* Having shown the validity of the approach, our inability to do performance testing on realistic data and query sets is of primary concern. Large amounts of scanned material are unavailable, and stardardized query sets do not exist; we invite all interested parties to join us in putting these resources together.
- *Test assumptions on lower-bound accuracy.* This work is based on certain judgments about our ability to approximate correctly all of the time. Their validity is intimately tied to the quality of input text; we would like to see at what point they begin to break down. By the same token, we have been extremely conservative in estimating present day OCR's discriminating ability, and would like to know how well we can do under relatively favorable conditions.
- *'Bootstrap' OCR and IR for SEA languages.* We are very interested in applying and extending these techniques in other unsegmented, non-Roman, multi-level writing systems like Lao, Khmer, and Burmese. 'Appropriate technology' for developing countries does not necessarily mean low-tech; there are many applications (eg. indexing the files gathered by the Cambodian Genocide Project) that would benefit enormously from even rudimentary IR systems.

8. Acknowledgments

The assistance of Namfon Boontua in preparing the word and query sets was invaluable, as were discussions with Dr. Pongskorn Saipetch of Atrium Software on the design of the approximation alphabets. Thanks also for the suggestions made by the anonymous reviewers of this paper.

9. References

[1] Church, K.W. et al. 'Fax: An Alternative to SGML.' In 'COLING '94,' Kyoto, Japan.

[2] Kimpan, C. et al. 'Thai Character Pattern Recognition Preliminary.' Conference L-05, Nihon University, Japan, 1980.

[3] Agui, T et al. 'A Recognition Method of Size Different Thai Characters' Country meeting conference, 81-1384, IECE, Japan, 1982.

[4] Hiranvanichakorn, Pipat and Boonsuwarn, Monlada. 'Recognition of Thai Characters.' In 'Proceedings of the Symposium on Natural Language Processing in Thailand,' Chulalongkorn University, 1993.

[5] Kimpan, Chom and Walairacht, Somsak. 'Thai Characters Recognition.' In 'Proceedings of the Symposium on Natural Language Processing in Thailand,' Chulalongkorn University, 1993.

[6] Tanprasert, C. and Koanantakook, T. 'Optical Character Reader for Thai Script Using Neural Networks.' In 'Southeast Asian Regional Computing Federation '96,' Bangkok, Thailand.

[7] Tanprasert, C. and Koanantakook, T. 'Thai OCR: A Neural Network Application.' In 'IEEE Region 10 Conference on Digital Signal Processing Applications,' Perth, Australia, 1996.

[8] Cooper, Doug. 'How Do Thais Tell Letters Apart? A Study of the Secondary Characteristics of Thai Letters.' In 'Pan-Asiatic Linguistics 1996 — 4th Int'l Symposium on Language and Linguistics, Mahidol University.'

[9] Cooper, Doug. 'Fuzzy Letters and Thai Optical Character Recognition.' In 'Symposium on Natural Language Processing '95,' Kasetsart University, Bangkok, Thailand.

[10] Cooper, Doug. 'Ambiguous (((Par(t)i)t)((ion))(s))in)) Thai Text.' In '11th Pacific Asia Conference on Language, Information, and Computation,' December 1996, Seoul, Korea.

[11] Powalk, R.K. et al. 'Multiple Word Segmentation with Interactive Look-Up for Cursive Script Recognition.' IDCAR '93, Tskuba Science City, Japan.

[12] Rose, T.G, et al. 'A Context-Based Approach to Text Recognition.' In '3rd International Symposium on Document Analysis and IR.' ISRI, UNLV, 1994.

[13] Croft, W.B. et al. 'An Evaluation of Information Retrieval Accuracy with Simulated OCR Output.' In '3rd Annual Symposium on Document Analysis and IR.' ISRI, UNLV, 1994.

[14] Taghva, K. et al. 'Results of Applying Probabilistic IR to OCR Text.' In '17th International ACM/SIG-IR.'

[15] Taghva, K. et al. 'Results and Implications of Noisy Data Projects.' In '3rd Int'l Symposium on Document Analysis and IR.' ISRI, UNLV, 1994.

[16] Ittner, D.J. et al. 'Text Categorization of Low Quality Images.' In '4th Int'l Symposium on Document Analysis and IR.' ISRI, UNLV, 1995.

[17] Nakayama, T. 'Content-Oriented Classification of Document Images.' In 'COLING '96,' Copenhagen, Denmark.

[18] Powalka, R.K. et al. 'Word Shape Analysis for a Hybrid Recognition System.' To appear Journal Of Pattern Recognition.

[19] Spitz, Lawrence A.. 'An OCR Based on Character Shape Codes and Lexical Information.' In 'Proceedings 3rd International Conference on Document Analysis and Recognition 1995,' Montreal, Canada.

[20] Cooper, Doug '45,656 Thai Names: Statistics and Implications of passlist.96.' In '1997 International Conference on Computer Processing of Oriental Languages,' Hong Kong.

[21] Data provided by ViewSiam Ltd. Bangkok (www.viewsiam.com).

[22] Haas, Mary. 'Thai-English Student's Dictionary.' Stanford University Press, 1964.

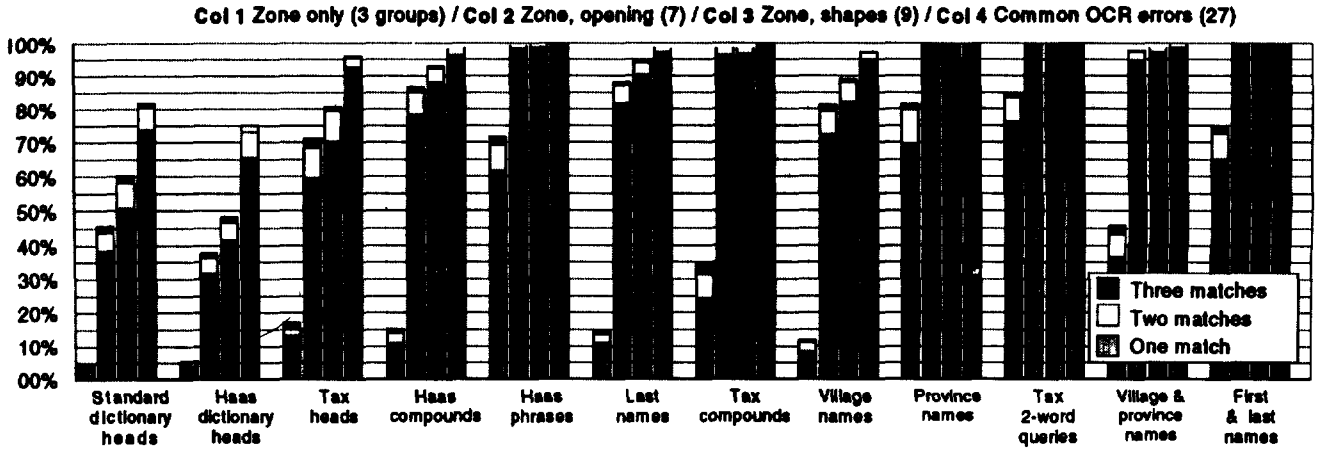
[23] 'Revenue Code Database,' CD-ROM. Published by LINKS Lab/-NECTEC, Bangkok, Thailand.

Query set	Alphabet	One match	Two matches	Three matches	Sum, 1-3 (items)
Standard dictionary headwords	set 1 (3 groups)	3.9%	0.9%	0.3%	5.1% (917)
Items: 17,986	set 2 (7 groups)	38.5%	5.4%	2.0%	45.9% (8,256)
Average length: 5.41	set 3 (9 groups)	51.7%	6.5%	2.0%	58.4% (11,187)
	set 4 (27 groups)	74.0%	6.4%	1.3%	81.7% (14,695)
Haas dictionary headwords	set 1	3.9%	1.1%	0.5%	5.5% (327)
Items: 5,941	set 2	31.5%	4.6%	1.7%	37.8% (2,246)
Average length: 4.68	set 3	41.5%	5.0%	1.9%	48.4% (2,875)
	set 4	65.5%	7.5%	2.1%	75.1% (4,462)
Tax code headwords	set 1	13.0%	1.9%	2.2%	17.1% (62)
Items: 361	set 2	60.1%	8.3%	2.7%	71.1% (257)
Average length: 4.4	set 3	70.3%	9.1%	1.1%	80.5% (291)
	set 4	92.7%	2.7%	0.5%	95.9% (346)
Haas dictionary compounds	set 1	11.6%	2.4%	1.1%	15.1% (1,760)
Items: 11,653	set 2	78.3%	6.6%	1.5%	86.4% (10,068)
Average length: 8.33	set 3	88.1%	4.2%	0.7%	93.0% (10,837)
	set 4	97.0%	1.3%	0%	98.3% (11,455)
Haas dictionary phrases	set 1	62.4%	6.8%	2.4%	71.6% (387)
Items: 541	set 2	98.5%	0.7%	0%	99.2% (536)
Average length: 11.35	set 3	98.8%	0.5%	0%	99.3% (537)
	set 4	99.6%	0.1%	0%	99.7% (539)
Last names	set 1	10.9%	2.6%	1.3%	14.8% (5,472)
Items: 36,977	set 2	81.7%	5.2%	1.1%	88.0% (32,540)
Average length: 9.32	set 3	90.8%	3.1%	0.5%	94.4% (34,906)
	set 4	97.3%	1.1%	0%	98.4% (36,385)
Tax code compounds	set 1	24.2%	7.2%	3.0%	34.4% (146)
Items: 425	set 2	96.7%	1.6%	0%	98.3% (418)
Average length: 7.44	set 3	97.1%	1.4%	0%	98.5% (419)
	set 4	100.0%	0%	0%	100.0% (425)
Village names	set 1	8.9%	2.1%	0.9%	11.9% (1,264)
Items: 10,625	set 2	72.8%	6.4%	2.1%	81.3% (8,638)
Average length: 11.44	set 3	82.4%	5.6%	1.1%	89.1% (9,467)
	set 4	95.0%	2.1%	0.2%	97.3% (10,338)
Province names	set 1	69.7%	10.5%	1.3%	81.5% (62)
Items: 76	set 2	100.0%	0%	0%	100.0% (76)
Average length: 7.62	set 3	100.0%	0%	0%	100.0% (76)
	set 4	100.0%	0%	0%	100.0% (76)
Tax code 2-word queries	set 1	76.1%	7.3%	1.5%	84.9% (604)
Items: 712	set 2	99.7%	0.1%	0%	99.8% (711)
Average length: 15.72	set 3	99.7%	0.1%	0%	99.8% (711)
	set 4	100.0%	0%	0%	100.0% (712)
Village and province names	set 1	36.5%	6.7%	2.6%	45.8% (6,302)
Items: 13,465	set 2	95.5%	1.9%	0.2%	97.6% (13,142)
Average length: 19.0	set 3	97.4%	1.1%	0%	98.5% (13,263)
	set 4	99.3%	0.3%	0%	99.6% (13,411)
First and last personal names	set 1	65.6%	6.9%	2.0%	74.5% (34,008)
Items: 45,648	set 2	99.9%	0%	0%	99.9% (45,602)
Average length: 15.77	set 3	99.9%	0%	0%	99.9% (45,602)
	set 4	99.9%	0%	0%	99.9% (45,602)

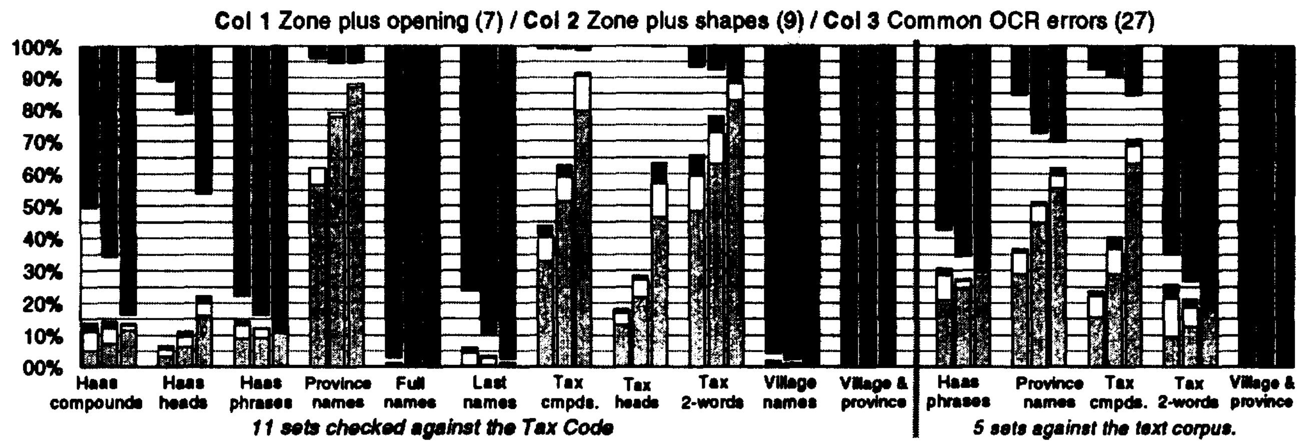
Table 1 Disambiguating approximate queries. We assume the complete query list exists in e-form, but that we must determine which query the user is making via some imprecise means (eg. pen-based input). A large figure in the *one match* column is best. However, for many practical applications, a small amount of overlap — incorrect matches — is not objectionable. The cutoff figure of two was chosen arbitrarily, but in general, performance is not dramatically increased by allowing a larger number of false matches. Query length is the best predictor of performance; fifteen-character queries were readily disambiguated using all but the three-letter approximation alphabet. Slight inconsistencies are due to rounding and spelling errors.

Queries against the 1-megabyte Thai Tax Code									
		Percentages against the full query set (graph 2)					*% against hits only (graph 3)*		
Query set	*Set*	*>2 false*	*Not found (items)*	*exact*	*1 false*	*2 false*	*exact*	*1 false*	*2 false*
Haas	2	35.5%	50.5% (5879)	5.1%	5.9%	3.0%	10.3%	11.8%	6.1%
compounds	3	19.8%	65.9% (7681)	7.2%	4.7%	2.4%	21.2%	13.6%	7.2%
Items: 11,653	4	2.6%	83.7% (9753)	11.2%	1.7%	0.8%	68.6%	10.3%	4.8%
Haas	2	82.1%	11.1% (658)	3.3%	2.2%	1.3%	3.8%	2.5%	1.5%
headwords	3	67.3%	21.4% (1270)	6.2%	3.2%	1.9%	7.9%	4.1%	2.5%
Items: 5,941	4	31.6%	46.2% (2745)	15.9%	4.0%	2.3%	29.6%	7.5%	4.3%
Haas	2	7.9%	77.8% (421)	8.9%	4.1%	1.3%	40.0%	18.3%	5.8%
phrases	3	3.9%	83.7% (453)	9.2%	2.8%	0.4%	56.8%	17.0%	2.3%
Items: 541	4	0.2%	88.4% (478)	10.5%	0.7%	0.2%	90.5%	6.3%	1.6%
Province	2	34.2%	3.9% (3)	56.6%	5.3%	0.0%	58.9%	5.5%	0.0%
names	3	15.8%	5.3% (4)	77.6%	1.3%	0.0%	81.9%	1.4%	0.0%
Items: 76	4	6.6%	5.3% (4)	86.8%	1.3%	0.0%	91.7%	1.4%	0.0%
Full	2	1.5%	97.2% (44370)	0.0%	0.9%	0.4%	0.0%	32.8%	13.6%
names	3	0.5%	99.5% (45421)	0.0%	0.2%	0.1%	0.0%	43.6%	13.7%
Items: 45,648	4	0.0%	100.0% (45638)	0.0%	0.0%	0.0%	0.0%	40.0%	10.0%
Last	2	17.6%	76.2% (28193)	0.3%	4.0%	1.9%	1.2%	16.8%	8.0%
names	3	7.1%	89.2% (32990)	0.5%	2.2%	1.0%	4.3%	20.3%	9.0%
Items: 36,977	4	1.0%	97.5% (36063)	0.8%	0.5%	0.2%	31.3%	21.9%	7.9%
Tax code	2	55.3%	0.9% (4)	32.7%	7.3%	3.8%	33.0%	7.4%	3.8%
compounds	3	36.2%	0.9% (4)	51.1%	7.8%	4.0%	51.5%	7.8%	4.0%
Items: 425	4	7.0%	1.4% (6)	79.3%	11.1%	1.2%	80.4%	11.2%	1.2%
Tax code	2	81.7%	0.0% (0)	13.0%	3.6%	1.7%	13.0%	3.6%	1.7%
headwords	3	71.5%	0.0% (0)	21.6%	5.5%	1.4%	21.6%	5.5%	1.4%
Items: 361	4	36.3%	0.3% (1)	46.3%	10.5%	6.6%	46.4%	10.6%	6.7%
Tax 2-word	2	27.5%	6.7% (48)	48.3%	11.0%	6.5%	51.8%	11.7%	6.9%
queries	3	14.5%	7.7% (55)	62.9%	9.6%	5.3%	68.2%	10.4%	5.8%
Items: 712	4	1.0%	9.8% (70)	82.6%	5.5%	1.1%	91.6%	6.1%	1.2%
Village	2	2.1%	95.8% (10184)	0.6%	0.8%	0.7%	15.2%	19.3%	17.9%
names	3	1.0%	97.7% (10376)	0.6%	0.4%	0.3%	27.3%	16.9%	14.9%
Items: 10,625	4	0.1%	99.1% (10530)	0.7%	0.0%	0.1%	76.8%	5.3%	9.5%
Village &	2	0.3%	99.6% (13414)	0.0%	0.1%	0.0%	2.0%	39.2%	9.8%
province	3	0.0%	99.9% (13451)	0.0%	0.1%	0.0%	7.1%	50.0%	7.1%
Items: 13,465	4	0.0%	100.0% (13463)	0.0%	0.0%	0.0%	50.0%	0.0%	0.0%
Queries against the 2-megabyte text corpus									
Haas	2	11.8%	57.5% (311)	20.7%	7.6%	2.4%	48.7%	17.8%	5.7%
phrases	3	6.9%	65.6% (355)	24.6%	2.0%	0.9%	71.5%	5.9%	2.7%
Items: 541	4	1.1%	69.5% (376)	28.1%	1.1%	0.2%	92.1%	3.6%	0.6%
Province	2	47.7%	15.8% (12)	28.9%	6.6%	1.3%	34.4%	7.8%	1.6%
names	3	21.1%	27.6% (21)	44.7%	5.3%	1.3%	61.8%	7.3%	1.8%
Items: 76	4	7.9%	30.3% (23)	55.3%	3.9%	2.6%	79.2%	5.7%	3.8%
Tax code	2	68.7%	7.5% (32)	15.5%	6.4%	1.9%	16.8%	6.9%	2.0%
compounds	3	50.0%	9.6% (41)	28.9%	7.5%	4.0%	32.0%	8.3%	4.4%
Items: 425	4	13.5%	15.8% (67)	63.1%	5.2%	2.4%	74.9%	6.1%	2.8%
Tax 2-word	2	9.5%	64.9% (462)	9.3%	11.8%	4.5%	26.4%	33.6%	12.8%
queries	3	5.5%	73.2% (521)	12.5%	5.9%	2.9%	46.6%	22.0%	11.0%
Items: 712	4	0.4%	81.0% (577)	17.7%	0.8%	0.1%	93.3%	4.4%	0.7%
Village &	2	0.2%	99.4% (13378)	0.0%	0.3%	0.1%	1.1%	44.8%	20.7%
province	3	0.1%	99.8% (13444)	0.0%	0.1%	0.0%	4.8%	47.6%	23.8%
Items: 13,465	4	0.0%	100.0% (13461)	0.0%	0.0%	0.0%	25.0%	50.0%	0.0%

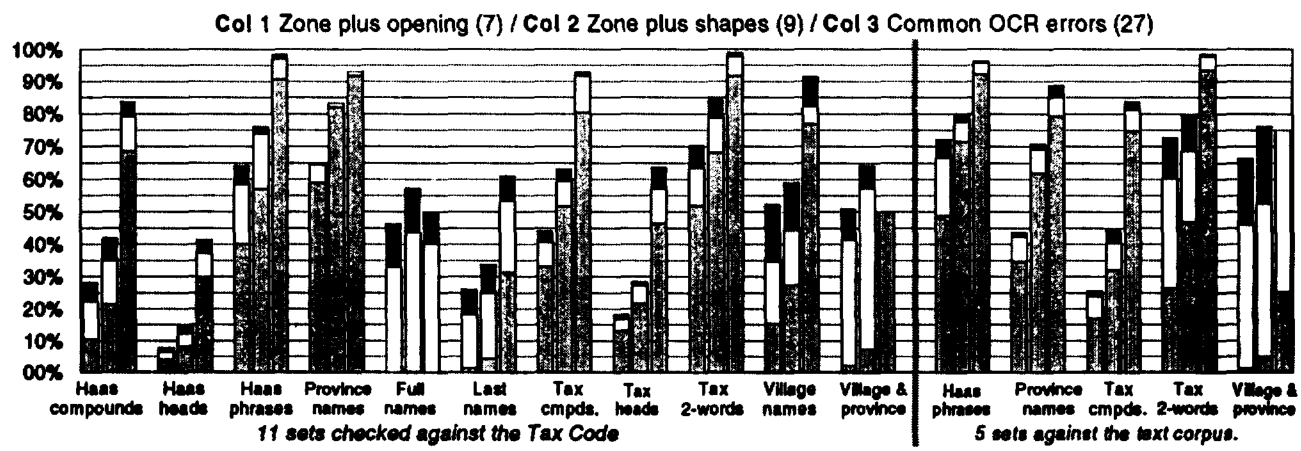
Table 2 Searching approximate data. We assume that the data have been scanned and approximately OCR'd, then intentionally approximate queries at the same level of detail. A smaller figure in the > *2 false* column is better; this number gives an indication of precision, and is equivalent to the mid-column gaps in graph 2. Once again, the cutoff figure of 2 false hits is arbitrary; we found that in many cases, one or two false responses were due to spelling errors in the text samples. Slight inconsistencies are due to rounding and spelling errors.



Graph 1 Disambiguating approximate queries. A taller gray portion mean that more words could be identified exactly on the basis of approximate information; shorter columns overall mean that more words were ambiguous. Note that within specific domains — province names, long queries, first and last names — even very crude approximations can be identified.



Graph 2 Searching approximate data (dark gray=0, light gray=1, white=2, black=3 items returned). Taller rising columns mean that more terms were *correctly* found, allowing 1, 2, or 3 items returned (ie. zero, 1, or 2 false hits, somtimes attributable to spelling errors); deeper falling bars (dark gray) mean that more words were *correctly* not found. The gap between represents the number of words found *incorrectly* and *frequently* — a large gap size indicates that many terms returned three or more false hits. A large *rising-column:gap* ratio indicates that of terms that had matches, relatively few had many false hits; this is shown more clearly in graph 3.



Graph 3 Searching approximate data, hits only (gray=1, white=2, black=3 items returned). Here, the non-hits are ignored. A taller gray area means that more terms were found *correctly*, with no false hits. Taller columns overall mean that of terms that returned something, more had two or fewer false hits. For example, graph 2 indicates that *full names* rarely returned matches against the Tax Code; graph 3 shows that while the hits were never correct, they returned just one or two false matches about half the time.

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู Angsana UPC 21

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู Cordia UPC 21

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู *JS Sirium 19*

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู *JS Toomtam 21*

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู SV Srimala 21

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู Kodchiang UPC 21

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู *SV Sakuntala 21*

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า (ำ)_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู **DS-Wisaka 21**

ชซข _ศคดต _ฆมน _ทฑ _ลส _อฮ _ห _ ย_ ง_ จ_ ร_ ว_ ฉ
กถภ(ฎฏ ฤฦๆ) _ ผพ(ฝฟฬ) _ ฒฌณ(ญ) _ บษ(ป) _ ธ(ฐ)
า [ำ]_ฯ _เ _แ_โ_ใ_ไ ๔ ฺ ั ๗ + ่ ิ ี ึ ื ุ ู **JS KOBORI ALLCAPS 18**

Appendix 1 A selection of Thai fonts, with letters grouped according to the approximation alphabet of set 4, and printed at approximately 150% of ordinary book size. Characters in parentheses belong to the preceding group, but are assumed to have an associated sub- or superscript character. The accuracy of the approximation groups varies slightly from font to font. We show two typical fonts from each of four groups — book, script/handwriting, modern/display, and decorative — plus a newspaper headline font.