Phonetic String Matching: Lessons from Information Retrieval

Justin Zobel
jz@cs.rmit.edu.au
Department of Computer Science
RMIT
GPO Box 2476V
Melbourne, Australia 3001

Philip Dart
philip@cs.mu.oz.au
Department of Computer Science
The University of Melbourne
Parkville
Melbourne, Australia 3052

Abstract

Phonetic matching is used in applications such as name retrieval, where the spelling of a name is used to identify other strings that are likely to be of similar pronunciation. In this paper we explain the parallels between information retrieval and phonetic matching, and describe our new phonetic matching techniques. Our experimental comparison with existing techniques such as Soundex and edit distances, which is based on recall and precision, demonstrates that the new techniques are superior. In addition, reasoning from the similarity of phonetic matching and information retrieval, we have applied combination of evidence to phonetic matching. Our experiments with combining demonstrate that it leads to substantial improvements in effectiveness.

1 Introduction

Phonetic matching is used to identify strings that may be of similar pronunciation, regardless of their actual spelling. A typical application is a "white pages" enquiry line, where a telephone operator is verbally given a name, guesses at the spelling (or is provided with a spelling, which may be incorrect), and uses the guess to query a database of names. The phonetic matching system must then find in the database those strings most likely to be of the same or similar pronunciation to that of the query. Since there is no reliable way of automatically determining the pronunciation of a string, such matching must be inexact.

There are two pragmatic issues that must be addressed in such a phonetic matching system. One is of speed—answers should be found reasonably quickly. We have shown elsewhere that, by indexing short substrings of the words in the databases, sets of matches can be identified in a small fraction of a second on current hardware [Zobel and Dart, 1995]. The other pragmatic issue is accuracy—as for information retrieval, some techniques are better than others at identifying matches. It is the issue of accuracy that we explore in this paper.

The parallels between information retrieval and phonetic matching mean that they can be measured by the same kinds of techniques. For example, it is, arguably, appropriate to compare phonetic matching techniques using recall and precision. In this paper we describe the results of a new comparative investigation of phonetic matching. We implemented several well-known techniques, such as Soundex and edit distances, and, based on earlier experiments [Zobel and Dart, 1995], have developed and explored new techniques. We then gathered data, queries, and relevance judgements, and used them to compare these matching techniques.[1]

The results show that our new phonetic matching techniques are indeed superior to the other techniques available. They also show, however, that some of the well-known difficulties with relevance are as evident in this domain as they are in information retrieval.

The parallels between information retrieval and phonetic matching also mean that methods for improving information retrieval performance may also apply to phonetic matching. In particular, we have experimented with combination of evidence, and have shown experimentally that it can lead to a marked improvement in performance—with best recall-precision improving from, for one set of judgements, 23.2% to 26.1%, a gain that is even more marked in the context of "baseline" performance (for a trivial algorithm) of 17.2%.

Phonetic matching and its similarities to information retrieval are discussed in Section 2. Techniques for phonetic matching are described in Section 3, and their performance analysed in Section 4. The results of combination of evidence are given in Section 5. Conclusions are given in Section 6.

2 Phonetic matching versus information retrieval

In information retrieval, ranking is the process of identifying which of a set of documents are most likely to be similar in content to a given query. Phonetic matching can, broadly, be described in analogous terms: it is the process of identifying which of a set of strings are most likely to be similar in sound to a given query string. In both cases the matching process is: fundamentally inexact, since human judgement is required to tell whether the process's guess is correct; likely

[1] The data, queries and judgements are publicly available from `ftp://goanna.cs.rmit.edu.au/pub/rmit/fnetik`, together with source code used for some of the experiments in this paper.

Code:	0	1	2	3	4	5	6
Letters:	a e i o u y h w	b p f v	c g j k q s x z	d t	l	m n	r

Figure 1: Soundex phonetic codes

1. Replace all but the first letter of the string by its phonetic code.
2. Eliminate any adjacent repetitions of codes.
3. Eliminate all occurrences of code 0 (that is, eliminate vowels).
4. Return the first four characters of the resulting string.

Figure 2: The Soundex algorithm

to involve ranking of the data set, rather than partitioning of the data set into matches and not-matches; and, since similarity is relative, unable in isolation to determine whether a query and potential answer are matches. In both cases it is difficult to give an accurate definition of relevance.

For example, when comparing the string `fret` to the strings

`clot`, `friend`, `grow`, `mouse`, and `rend`,

we could decide that `friend` is the best match, `rend` an acceptable match, and `mouse` a bad match. Against other data sets—also including, say, `fred` and `flet`—we might judge differently, and rank `rend` as a bad match.

However, the definition of phonetic matching as a search for strings "of similar sound" is too vague for practical purposes—such a definition might include rhymes, for example. More specifically, we consider phonetic matching to be the process of identifying strings that, after elimination of possible transmission or cognition errors, may sound the same. Transmission errors include, for example, sound-alike mistakes in data entry such as entering `surl` for the spoken name `searle`; mishearing of a spoken name on a imperfect transmission medium such as a telephone; or "chinese whispers" errors in which a name is repeated by a chain of people, some of whom do not communicate the name correctly. Cognition errors include, for example, mistaking a pronunciation for an expected word, such as hearing `america` for `emeritus`.

That is, phonetic matching is the processing of finding strings that, prior to possible changes that broadly preserve the sound, may have had the same pronunciation.

Given the similarity of phonetic matching and information retrieval, it follows that, at least in general terms, techniques for addressing these problems should be measured in the same way. We can therefore measure the performance of an algorithm for phonetic matching by assembling test data consisting of a set of strings, a set of queries, and, for each query, a set of relevance judgements. This test data can be used exactly as for experiments in information retrieval: we can determine recall-precision using an 11-point average, determine precision at various numbers of answers retrieved, test the reliability of our experiments with a Wilcoxon signed-rank test, and so on.

In the context of information retrieval, however, such methods of assessment of systems have known limitations [Wallis and Thom, 1996]. In particular, there can be significant disagreement between judges; however, it is argued that these disagreements do not affect the outcome of comparision of systems [Lesk and Salton, 1969]. Our experiences with collection and use of relevance judgements are described in Sections 4 and 5. Now we describe the phonetic matching techniques we evaluated.

3 Phonetic matching techniques

In this section we describe techniques for phonetic matching, including new phonetic matching techniques developed by us (Editex and phonometric methods); existing techniques designed for phonetic matching (Soundex and Phonix); techniques designed for approximate string matching that have properties that make them suitable for phonetic matching (q-grams, agrep, and edit distance methods); and a refinement of edit distance techniques (tapering).

Soundex is the best-known phonetic matching scheme. Developed by Odell and Russell, and patented in 1918 [Hall and Dowling, 1980], Soundex uses codes based on the sound of each letter to translate a string into a canonical form of at most four characters, preserving the first letter. The Soundex codes and algorithm are given in Figures 1 and 2. For example, `reynold` and `renauld` are both reduced to `r543`, but, more commonly, Soundex makes the error of transforming dissimilar-sounding strings such as `catherine` and `cotroneo` to the same code, and of transforming similar-sounding strings to different codes. There is no ranking of matches: strings are either similar or not similar.

Phonix is a Soundex variant [Gadd, 1988, 1990]. Letters are mapped to a set of codes using the same algorithm, but a slightly different set of codes is used, and prior to mapping about 160 letter-group transformations are used to standardise the string. For example, the sequence `tjV` (where `V` is any vowel) is mapped to `chV` if it occurs at the start of a string, and `x` is transformed to `ecs`. These transformations provide context for the phonetic coding and allow, for example, `c` and `s` to be distinguished. The Phonix codes are shown in Figure 3.

The truncation of Soundex and Phonix codes to four characters is useful if an exact index is required, but is less

Code:	0	1	2	3	4	5	6	7	8
Letters:	a e i o u y h w	b p	c g j k q	d t	l	m n	r	f v	s x z

Figure 3: Phonix phonetic codes

$$
\begin{aligned}
&edit(0,0) = 0\\
&edit(i,0) = i\\
&edit(0,j) = j\\
&edit(i,j) = min[&&edit(i-1,j)+1,\\
&&&edit(i,j-1)+1,\\
&&&edit(i-1,j-1)+r(s_i,t_j)]
\end{aligned}
$$

Figure 4: Recurrence relation for minimal edit distance

valuable if an approximate string matching technique such as an edit distance is available. In our experiments we consider a variant of Phonix, here called Phonix+, in which truncation is not applied and a minimal edit distance (described below) is used to compare the resulting strings.

Q-gram methods are string distance measures based on q-gram counts, where a q-gram of string s is any substring of s of some fixed length q. A simple such measure is to choose q and count the number of q-grams two strings have in common. However, simply counting q-grams does not allow for length differences; for example, `fred` has exactly as many q-grams in common with itself as it does with `frederick`. To address this problem, Ukkonen has proposed an q-gram distance [Ukkonen, 1992], which for strings without repeated q-grams (q-gram repeats are rare in names) can be defined as

$$|G_s| + |G_t| - 2|G_s \cap G_t|\,,$$

where G_s is the set of q-grams in string s. For example, according to this formula the distance between `rhodes` and `rod` is 5 for q of 2 or 3. In our experiments we have used this q-gram method with $q = 2$.

Such methods are not dissimilar to the similarity measures used in information retrieval, such as the cosine measure [Salton, 1989]. It might be supposed that such measures would be appropriate for phonetic matching with q-grams, but although the approaches are superficially similar, there is a crucial respect in which they differ: effective similarity measures factor out document length. For phonetic matching, this behaviour is undesirable. The cosine measure would, for example, regard a word as being as similar to any one of its q-grams as it is to itself.

Agrep is a utility that embodies a fast algorithm for identifying strings that contain a substring which is identical to a query but for at most k insertions, deletions, or replacements, where k is a predefined constant [Wu and Manber, 1992]. In this context, agrep rapidly finds the strings that are identical within the given tolerance.

Matches are not ranked. It would be straightforward to modify agrep to rank strings according to the error count; the result would be an algorithm that yielded the same ranking as the edit distance described below. Note that agrep was not designed for the task of phonetic matching, but rather for fast searching of large files.

Edit distances are measures of string similarity. A simple edit distance, which counts the minimal number of single-character insertions, deletions, and replacements needed to transform one string into another, could be used for phonetic matching since similar-sounding words are often spelled similarly. For two strings s and t of length m and n respectively, this edit distance can computed with the recurrence relation $edit(m, n)$ shown in Figure 4, in which the function $r(a, b)$ returns 0 if a and b are identical, and 1 otherwise [Hall and Dowling, 1980]. For example, the edit distance between `rhodes` and `rod` is 3. The edit distance can be computed in $\Theta(nm)$ time using dynamic programming.

We now describe our new phonetic matching techniques.

Editex is a phonetic distance measure that combines the properties of edit distances with the letter-grouping strategy used by Soundex and Phonix. We developed Editex after running experiments with Soundex, Phonix, and edit distances, and observing the matches found by the phonetic methods and not the string methods: although Soundex and Phonix are not very effective, they do find good matches that standard edit distances cannot. Soundex and Phonix require letter groups with distinct codes to determine a canonical representation for strings; it follows that these groups must partition the set of letters. Editex also groups letters that can result in similar pronunciations, but doesn't require that the groups be disjoint and can thus reflect the correspondences between letters and possible similar pronunciation more accurately.

Editex is defined by the edit distance recurrence relation of Figure 5 with a redefined function $r(a, b)$ and an additional function $d(a, b)$. For Editex, the function $r(a, b)$ returns 0 if a and b are identical, 1 if a and b are both occur in the same group, and 2 otherwise. The groups are listed in Figure 6. The function $d(a, b)$ is identical to $r(a, b)$—thus allowing pairs of the same letter to correspond to single occurrences of that letter—except that if a is `h` or `w` (letters that are often silent) and $a \neq b$ then $d(a, b)$ is 1. Note the similarity between the Editex and Phonix letter groupings;

$$
\begin{aligned}
&edit(0,0) = 0\\
&edit(i,0) = edit(i-1,0) + d(s_{i-1}, s_i)\\
&edit(0,j) = edit(0,j-1) + d(t_{j-1}, t_j)\\
&edit(i,j) = min[edit(i-1,j) + d(s_{i-1}, s_i),\\
&\qquad\qquad edit(i,j-1) + d(t_{j-1}, t_j),\\
&\qquad\qquad edit(i-1,j-1) + r(s_i, t_j)]
\end{aligned}
$$

Figure 5: Recurrence relation for Editex edit distance

0	1	2	3	4	5	6	7	8	9
aeiouy	bp	ckq	dt	lr	mn	gj	fpv	sxz	csz

Figure 6: Editex letter groups

but while Phonix groups the letter `h` and `w` with the vowels, Editex handles these as deletions; and Phonix does not group `c` and `s`.

Phonometric methods are matching techniques we have developed [Zobel and Dart, 1996] based on the study of phonetics [Calvert, 1992; Gimson and Cruttenden, 1994; Ladefoged, 1982]. Our algorithms for phonometric matching consist of two stages: first, the string of letters is converted into a string of phonemes by a string-to-pronunciation conversion algorithm [Carney, 1994]. There are several good algorithms for this purpose, but the more effective algorithms rely on context—the position of a word in a sentence, for example—and in phonetic matching no context is available. Moreover, the spelling and pronunciation of names is more variable than that of other words and thus the conversion process is inevitably approximate.

The second stage is comparison of strings of phonemes. We have developed an Editex-like algorithm in which the distance between phonemes varies. For example, it is possible to determine from tables of phonetic features that the phonemes `t` and `d` (which differ only in voice) are more similar than the phonemes `s` and `m` (which differ in several features).[2] Because of its complexity we do not reproduce here our table of distances between phonemes; a full description is available elsewhere [Zobel and Dart, 1996]. The distance between pronunciations as represented by strings of phonemes can be measured more precisely than the distance between strings of letters. Thus, given a reliable string-to-pronunciation algorithm, we would expect a phonometric method to give the best phonetic matching. We call the combination used in this paper, of Ainsworth's string-to-pronunciation algorithm [Ainsworth, 1973] and our phoneme-string edit distance, Ipadist.

Tapering is a refinement to the edit distance techniques based on a human-factors property: differences at the start of a pronunciation can be more significant than differences at the end. A tapered edit distance of particular interest is one in which the maximum penalty for replacement or deletion at start of string just exceeds twice the minimum penalty for replacement or deletion at end of string. Such an edit distance in effect breaks ties: two errors always attract a higher penalty than one, regardless of position; but strings with one error are ranked according to the position in which the error occurs. Our experiments included tapered versions of both the minimal edit distance and Editex.

[2] We have chosen these phonemes as examples because, in contrast to the many phonemes that are not represented by roman characters, the reader will understand what sound they correspond to.

4 Performance assessment

Comparison of techniques for phonetic matching requires, in the first instance, a data set. In our initial work we used confidential data sets available to us through commercial work [Zobel and Dart, 1995], but felt that the study of phonetic matching would be better served by use of data available in the public domain.

Over a period of about three days we extracted from Internet news the "From:" lines in each article, yielding about 70,000 distinct names. However this source of names, although plentiful, is extremely noisy. (Senders ranged from "Alien Space Monster" to "American Psychiatric Association Library" and names followed by long strings of degrees and telephone numbers.) We therefore hand-edited it into a standard format. The final file has just over 30,000 distinct surnames.

Queries were collected from the Melbourne White Pages telephone directory, by generating page numbers randomly and choosing the first surname in the second column on each page. In total we used 100 queries.

Our resources did not permit exhaustive relevance judgements, so we used a pooled method, in which each of 125 matching techniques (including the combined methods described in Section 5) returned 30 answers for each query, and the combined set of answers were pooled for judgement. As a confidence test we subsequently used each technique to find 40 answers for each query; across the set of queries we found only a few additional relevant names.

The most difficult problem was collection of the judgements themselves. Relevance judgements for each query were determined by a team in which one person read out the query and a potential match and the other listened to judge whether they were similar. The instructions directed the assessors to

> regard a name and query as a match if you think they may be a match, that is, a name and query

Method	Set of judgements		
	A	B	C
Editex	23.1 (17.8)	28.2 (4.3)	28.0 (6.9)
Ipadist	23.2 (16.4)	23.2 (3.8)	24.5 (6.1)
Tapered editex	21.6 (16.3)	26.0 (3.9)	22.0 (6.1)
Edit distance	20.5 (15.4)	24.0 (3.7)	24.6 (6.3)
Tapered edit	20.9 (14.5)	23.4 (3.9)	20.4 (5.9)
Q-gram	20.1 (16.5)	20.7 (3.5)	22.8 (6.2)
Baseline	17.2 (3.5)	18.2 (1.2)	18.4 (2.1)
"Best" agrep	12.1 (0.8)	20.3 (0.6)	14.9 (0.8)
Phonix+	12.0 (11.7)	7.2 (2.7)	9.0 (4.1)
Phonix	10.9 (6.5)	6.8 (1.7)	10.4 (3.2)
Soundex	10.0 (6.0)	7.2 (1.8)	9.0 (3.2)

Table 1: Phonetic matching techniques: percentage recall-precision (number relevant at 200)

`crews`:

Ipadist:	`crews`	`krewe`	`kreuser`	`crew`	`drews`	`clews`	`kruse`	`kroose`
Editex:	`crews`	`cress`	`clews`	`drews`	`crew`	`creps`	`kress`	`cross`

`farah`:

Ipadist:	`farah`	`fehr`	`fahr`	`fah`	`farace`	`faraz`	`fohr`	`farish`
Editex:	`farah`	`farrar`	`farrall`	`farra`	`faraz`	`faraj`	`vara`	`vaara`

Figure 7: Top-ranked answers for the names crews and farah

> are a match whenever you cannot be sure they are distinct. Thus you would not be likely to match "plank" with "puddle", but could well match "game" and "gain" or "wheel" and "weir".

The use of a reader and listener was designed to ensure that judgements were based on sound rather than spelling. But assessors move in mysterious ways. In some cases, for example, where two teams of assessors judged the same queries and both identified a good number of relevant words, only one or two of the words were in common. A particular source of such problems was a tendency for the reader to indicate to the listener, via intonation or body language, whether the name was a likely match—thus thwarting the experimental design.

We initiated two sets of judgements, set A on 25 queries and set B on 50 queries. Comparing the judgements, we considered set A to be reasonably reliable, and set B to be somewhat flawed for the reasons discussed above. We also had another set (set C) of judgements on 50 queries, conducted by a single assessor rather than a team, which we would expect to favour edit-distance approaches. Set A has 26.5 correct matches per query, set B has 5.4, and set C has 8.7.

We can now compare the various approaches to phonetic matching. Results are shown in Table 1, which is of 11-point recall-precision. (These results are in fact from a subset of the methods tested; the results for several of the less successful methods are omitted.) For many of the techniques tested, only a few distinct ranks are possible, and some techniques only return two ranks, match and not-match. In such circumstances standard methods for computing recall-precision can be unreliable, and alternatives have been proposed [Raghavan et al., 1989]. To give reliable results with the standard method for computing recall-precision, we randomly permuted, in each answer set, the answers of equal rank. The reported results are the averages of recall-precision figures computed for ten permutations. The reliability of the results is confirmed by the figures in brackets, which are the average number of correct matches in the top 200, in effect a test of the ability of the method to find all matches; deepening the search from 200 to 300 produces few further matches, indicating that the methods are, by this point, no longer able to match sensibly. (Inspection of the answer sets reveals that such low-ranked answers are being returned because they have, for example, two of six letters in common with the query—hardly a strong basis for assuming they are of similar sound. Moreover, the least effective methods such as Phonix and Soundex only return a small number of answers for most queries.) The "baseline" results are for a trivial phonetic matching method: find all strings with at most one character—an insertion, deletion, or replacement—different from the query. Methods below this baseline can be regarded as extremely poor.

Despite our uncertainty about the reliability of the relevance judgements, the results are fairly consistent. Soundex and Phonix are poor indeed, not only finding many wrong answers but not finding many right ones. Nor is Phonix+, in which the translated strings are not truncated, particularly better. At the other end of the scale Editex has done well, consistently outperforming the minimal edit distance and q-grams. This is a significant result: in our earlier work [Zobel

	(none)	Ipadist	Editex	Soundex	Phonix+	Phonix
(none)	—	23.2	23.1	10.0	12.0	10.9
Edit distance	20.5	24.7	24.3	21.0	23.3	22.2
Q-gram	20.1	24.6	24.8	19.3	26.1	21.2
Baseline	17.2	25.3	23.9	16.8	19.9	17.4

Table 2: Recall-precision for combination of evidence, using set A of relevance judgements (%)

and Dart, 1995], we had identified q-grams and the minimal edit distance as the best methods for phonetic matching.

The "'best' agrep" figures are for agrep in best-match mode, in which the strings with the minimum number of differences are returned as answers. A particular problem of this approach is the tiny number of correct answers returned—less than one per query—but we stress that agrep was not designed for this task. Also, in this mode agrep can easily return no correct matches at all: `hallis` might be the only string with one difference to `wallis`, but it is not a good match.

Some questions, however, remain open. The performance of tapering is disappointing, despite good indications from our initial experiments and strong positive results from work with users of a commercial system.

Nor has the Ipadist phonometric approach consistently performed well, and given the additional complexity of the phonometric method (compared to Editex) at this stage we would have to regard it as at best an interesting alternative. However, there remains scope for refining it, in particular by use of a more accurate algorithm for translating strings to phonemes, perhaps designed specifically for names.

Moreover, there are on reflection good reasons why Editex might perform as well as phonometric methods. Ipadist is the more precise algorithm—in that it uses phonetics rather than assuming that letters represent sounds—but is based on the assumption that its estimate of how the string should sound is correct. Editex transforms strings as it compares them, making rather crude assumptions about what characters can sound alike; but, in contrast to Ipadist, still gives some consideration to the original spelling.

This line of reasoning suggests possible new approaches to phonetic matching, such as, for example, performing phonetic translation during edit-distance comparison. This approach has the disadvantage, however, of being exponential rather than quadratic in string length, because there are usually several ways in which characters can be aggregated into phonemes. Such an algorithm is given by Veronis [1988]. We have also explored another, related approach, in which strings are transformed into all likely pronunciations rather than the single most likely pronunciation [Dart and Zobel, 1995b], but again the costs are unacceptable, and we have found no advantage in terms of effectiveness.

An interesting discovery is that even the most successful of the methods fetch rather different sets of answers, sometimes almost without overlap. For example, for the names `crews` and `farah` the top-ranked answers returned by Ipadist and Editex are shown in Figure 7.

The methods statistically perform about equally well on these queries. As for information retrieval, it seems, two methods can perform well without finding the same answers.

5 Combination of evidence

We argued above that phonetic matching has strong parallels with information retrieval. An aspect of the parallel is that, in both cases, matching techniques fetch a ranked list of matches in which each entry has a weight attached to it; this weight is the likelihood that the entry is a good match.

In the context of information retrieval, several studies [Belkin et al., 1993; Fox and Shaw, 1993; Lee, 1995; Wilkinson et al., 1995] have shown that combining the ranked lists produced by different retrieval mechanisms can improve performance. The intuition behind this process is that, if two different mechanisms (presumably interpreting the query in different ways) both regard a particular item as likely to be relevant, that is better evidence of relevance that the judgement of a single mechanism alone.

Similar logic can be applied to phonetic matching: if, for example, a string is judged to have both similar spelling and similar sound to a query, then the likelihood of relevance should be greater than is given by the evidence of spelling or sound alone. To test this theory, we ran experiments in which each of the spelling-comparison methods (baseline, q-grams, and edit distance) was combined with the methods with a phonetic basis (Soundex, Phonix+, Phonix, Editex, and Ipadist). These experiments consisted of using the respective methods to find the best matches, normalising the ranking weights so that the best match returned in each case scored 1.0, then adding the normalised weights given by the respective methods for each match.

Results are shown in Table 2. The "(none)" lines are the results of running the methods individually, as reported in Table 1. As can be seen, in every case but one the effect of combining evidence is to improve performance. In some cases—of combining the weaker methods—the improvement is quite substantial. In particular, the best performance of all is given by the combination of Phonix+ and the q-gram method, neither of which works particularly well alone.

We have observed similar behaviour with our other sets of relevance judgements—combination of evidence almost always improves performance—but with different combinations having best performance in each case. For set B, successful combinations were the q-gram method with Ipadist and the minimal edit distance with Editex, and q-grams combined with Phonix+ was poor. For set C, the most successful combination was the minimal edit distance with Ipadist. Thus combination is successful, but our certainty in precisely which combinations are best is tempered by conflicts between the relevance judgements.

More sophisticated techniques for combination could be used: weighting the ranks from the different techniques, combining more than two methods, and so on. We plan to explore the techniques for achieving better performance

based on these approaches, but are satisfied by our core result: that combination of evidence is successful in this context.

6 Conclusion

We have developed several new methods for phonetic matching, and have used measurement techniques developed for information retrieval to compare them. The results, for every set of relevance judgements, showed that two of our proposals—the Ipadist and Editex methods—do indeed lead to improved performance, whereas the third—tapering—was not successful. These results also confirmed our earlier, more preliminary work, again showing that Soundex and Phonix, the two best-known phonetic matching techniques, are particularly poor at finding matches.

We were also able to use the similarity with information retrieval to yield a new approach to phonetic matching: we showed that combination of evidence, which has been successfully applied to information retrieval, consistently improves performance.

We discovered, however, that different sets of relevance judgements yielded inconsistent results, even for queries on the same set of data. While it seems clear that Editex, Ipadist, and combination of evidence do improve performance, it is difficult to make specific recommendations about exactly which technique should be used—the inconsistencies between relevance judgements made it difficult to compare systems reliably. Nonetheless we were able to use the judgements to draw general conclusions about performance of phonetic matching techniques, showing that our new methods are substantially more effective than existing methods such as edit distances, and that combination of evidence is as valuable in this domain as it is in information retrieval.

Acknowledgements

We would like to thank our relevance assessors: Kevin Chan, Michael Coburn, Marcin Kaszkiel, Daniel Knapp, and Andrew Parker. We would also like to thank Ross Wilkinson and Hugh Williams. This work was supported by the Australian Research Council.

References

Ainsworth, W. [1973]. A system for converting English text into speech. *IEEE Transactions on Audio and Electroacoustics*, AU-21(3):288–290.

Belkin, N., Kantor, P., Cool, C., and Quatrain, R. [1993]. Combining evidence for information retrieval. In Harman, D., editor, *Proc. Text Retrieval Conference (TREC)*, pages 35–44, Washington. National Institute of Standards and Technology Special Publication 500-215.

Calvert, D. [1992]. *Descriptive Phonetics*. Thieme Medical Publishers, New York, revised second edition.

Carney, E. [1994]. *A Survey of English Spelling*. Routledge, London.

Dart, P. and Zobel, J. [1995a]. Effective phonetic string matching. Manuscript in submission.

Dart, P. and Zobel, J. [1995b]. Using a pronounciation dictionary for fnetik matching. Technical Report 95/28, Department of Computer Science, The University of Melbourne.

Fox, E. and Shaw, J. [1993]. Combination of multiple searches. In Harman, D., editor, *Proc. Text Retrieval Conference (TREC)*, pages 35–44, Washington. National Institute of Standards and Technology Special Publication 500-215.

Gadd, T. [1988]. 'Fisching fore werds': Phonetic retrieval of written text in information systems. *Program: automated library and information systems*, 22(3):222–237.

Gadd, T. [1990]. PHONIX: The algorithm. *Program: automated library and information systems*, 24(4):363–366.

Gimson, A. and Cruttenden, A. [1994]. *Gimson's Pronounciation of English*. Edward Arnold, London, fifth edition.

Hall, P. and Dowling, G. [1980]. Approximate string matching. *Computing Surveys*, 12(4):381–402.

Ladefoged, P. [1982]. *A Course in Phonetics*. Harcourt Brace Jovanovich, San Diego, second edition.

Lee, J. [1995]. Combining multiple evidence from different properties of weighting schemes. In Fox, E., Ingwersen, P., and Fidel, R., editors, *Proc. ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 180–188, Seattle, Washington.

Lesk, M. and Salton, G. [1969]. Relevance assessment and retrieval system evaluation. *Information Storage and Retrieval*, 4(4):343–359.

Raghavan, V., Jung, G., and Bollmann, P. [1989]. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems*, 7(3):205–229.

Salton, G. [1989]. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts.

Ukkonen, E. [1992]. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92:191–211.

Veronis, J. [1988]. Computerized correction of phonographic errors. *Computers and the Humanities*, 22:43–56.

Wallis, P. and Thom, J. [1996]. Relevance judgements for assessing recall. *Information Processing & Management*. (To appear).

Wilkinson, R., Zobel, J., and Sacks-Davis, R. [1995]. Similarity measures for short queries. In *Proc. Text Retrieval Conference (TREC)*. (Proceedings to appear).

Wu, S. and Manber, U. [1992]. Fast text searching allowing errors. *Communications of the ACM*, 35(10):83–91.

Zobel, J. and Dart, P. [1995]. Finding approximate matches in large lexicons. *Software—Practice and Experience*, 25(3):331–345.

Zobel, J. and Dart, P. [1996]. Fnetik: An integrated system for phonetic matching. Technical Report 96-6, Department of Computer Science, RMIT.