Improving Precision and Recall for Soundex Retrieval

David Holmes
NCR Corporation
david.holmes@ncr.com

M. Catherine McCabe
U.S. Government
mary.catherine.mccabe@home.com

Abstract

We present a phonetic algorithm that fuses existing techniques and introduces new features. This combination offers improved precision and recall.

1 Introduction

Names play a key role in information systems. They are frequently used as search criteria for information retrieval and identity matching systems. They are critical to applications based on names such as libraries (authors), police files (perpetrators, victims), immigration, customs, bookstores, businesses (vendors, customers).

In the field of tax compliance, identity matching helps find non-filers by locating individuals found in external source systems that are absent from tax systems. In Texas, solving this problem was estimated to be worth at least $43M over 18 months[NCR98].

Misspellings, nicknames, phonetic and cultural variations complicate name-based information retrieval. The challenge is to improve recall without lowering precision. The soundex algorithm developed by Russell was an early attempt at assigning a common phonetic code to similar sounding words and names. [Celko95] and [Pfeifer3] offer substantial enhancements to the original approach.

Traditional approaches to soundex retrieval share a common weakness. Assigning a single phonetic code to each name assumes that one algorithm provides the best fit for all situations. Alternatively, an algorithm could blend multiple techniques. A second common design flaw in soundex retrieval is the lack of a similarity metric for assessing the closeness two names to each other.

We demonstrate fusion for improving the precision and recall of name searches, by combining Russell, Celko and Pfeifer techniques with our own. The experiments described herein assign multiple phonetic codes to each name. Counting common phonetic codes and digrams, the experiments implement the Dice Co-Efficient to assign a similarity score between names. We use the Pfeifer corpus and relevance assessments to compare and contrast experimental results with traditional techniques.

2 Prior Work

Russell and O'Dell developed the soundex algorithm which provides an inexact search capability to information retrieval (IR) systems by equating variable length text to fixed length alphanumeric codes. Precision and recall are somewhat limited because the translation process only considers single letters, with the exception of repeating consonants.

2.1 N-grams

[Zamora81] and [Angell83] make a case for using trigrams to overcome some of the complexities of unstructured text. For instance, trigrams can be used to equate "Mississippi" with "Misisippi". N-grams are also valuable in other ways. Celko describes an algorithm that considers n-grams during translation.

[Damerau64] introduced a similarity measure counting differences between two words. Pfeifer, et al, combined Damerau with other similarity metrics to rank search results. All four phonetic algorithms overcome some of the spelling errors described by Damerau when generating codes. Table 1 quantifies four types of errors commonly found in text and in three of the examples, the difference is not actually an error, but a common variation of a name. The approach taken by Celko overcomes the specific insertion, omission and substitution variations found in Table 1. Every technique described in this paper fails on the specific transposition noted below.

Type of Error	Baseline Name	Deviation
Insertion	Fisher	Fischer
Omission	Johnston	Johnson
Substitution	Catherine	Katherine
Transposition	Hagler	Halger

Table 1. Error classifications.

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'02)
0-7695-1506-1/02 $17.00 © 2002 IEEE

Pfeifer, et al, clearly demonstrated improved precision by implementing an n-gram based similarity metric for various phonetic matching algorithms. Similarity scores were assigned to potential matches, based upon the number of n-grams shared by two names with common phonetic codes. The Pfeifer similarity coefficient (δ) for a conditional name's n-grams (α) and a result name's n-grams (β) is:

$$\delta = \frac{\alpha \cap \beta}{\alpha \cup \beta}$$

For example, "Cook" and "Cooke" would have a higher similarity score than "Cook" and "Cake", because they share more n-grams. Pfeifer, et al, obtained best results by using digrams with a leading and trailing blank. The leading and trailing blank add importance to matching initial and terminal letters.

Of course, those familiar with the Russell soundex algorithm know that "Johnson" and "Johnston" have different codes. N-gram substitution prior to translation improves recall by normalizing some phonetically similar combinations. For example, substituting PH with FF allow "Philip" and "Filipe" to share a common code.

2.2 N-gram Substitution

Celko describes how the usage of a letter alters its sound and his algorithm assigns variable codes to some letters depending upon their n-grams. For example, the algorithm substitutes "t" with "s" when it is found in the trigram "nst". We implement n-gram substitution similar to Celko, replacing digrams such as "ca" with "ka." and reducing substitution errors described by Damerau. We further refine this technique by adding positional dependence to the rules.

Prefix Substitution. Substitution can be limited to prefixes. For example, Celko replaces "Mac" with "Mcc".

Suffix Substitution. Some substitutions are limited to suffixes. The Celko algorithm drops terminal "t" when preceded by "n" or "ns". Pfeifer stems names after the last vowel.

Non-positional Substitution. Most n-gram substitution rules have no dependency upon the position of the n-gram within the name. For example, we always replace "ca" with "ka".

Silent Letters. Many conversions remove silent letters. By converting "sch" to "sss", many similar names are equated; "Bush" and "Busch", "Fisher" and "Fischer", "Schuler" and "Shuler".

2.3 Character Translation Rules

Translating names into alphanumeric codes is one way to equate similar names. The earliest of the algorithms described here, the Russell soundex, translates names into four byte alphanumeric codes. The others inherit some of their rules from the original. For example, rules in the Pfeifer and fuzzy algorithms, like Russell, convert "D" and "T" to "3". Several rules govern the translation process and are described below.

Consonant Removal. Except for the Celko approach, "H", "W" and "Y" are not translated. Their only significance is servings as a separator. The Celko approach removes "H" when it is not preceded by an "A".

Duplicate Consonant Removal. Repeating, consecutive consonants are removed by each algorithm. A single letter is translated. For example, only one "L" in "Miller" is translated by any of the approaches.

Vowel Removal. All non-leading vowels are removed by each algorithm. Leading vowels are retained by the Russell and fuzzy algorithms. Both of the others replace leading vowels with a constant; the Celko replacement rule replaces all leading vowels with an "A" and Phonix replaces with a "V".

Leading Characters. Retained, leading characters are always translated to an alpha character. For Celko, all other letters also translate to alpha codes, but the other approaches produce numeric codes for non-leading positions.

Ending Sounds. Phonix drops ending letters that follow the last vowel or Y. Our algorithm, as well as the Celko approach, drop some terminal letters, depending upon suffix n-grams.

Translation Length. The original soundex algorithm always produces a four byte code, zero padding unused trailing positions. The Celko approach pads trailing positions with spaces. Phonix can produce four or eight byte codes. Pfeifer indicated the best results for Phonix are obtained by using a four byte code. Our algorithm uses multiple code lengths. Typically, short codes provide high recall and low precision, while long codes produce high precision and low recall.

Multiple Codes. A unique aspect of our approach is that we produce multiple phonetic codes for every name. We use Russell, Celko and Fuzzy soundex codes of varying lengths to assign up to ten distinct codes per name.

2.4 Fusion

[McCabe99] showed improved precision by fusing the results from multiple retrieval strategies. Fusion appeared more effective when each method favored different relevant documents.

2.5 IR and the Relational Model

Information Retrieval within the relational framework is a well established concept [Grossman98]. and provides many of the features found in traditional inverted index IR systems, including stemming, phrases, proximity searches, similarity measures and n-grams. While relational IR may be slower than inverted indexes, [Lundquist99] demonstrated near linear scalability using Teradata and [McCabe99] showed rapid response times on a single node, Windows NT server running Teradata.

3 Fuzzy Soundex

The fuzzy soundex algorithm uses some elements from previous work with an emphasis on n-gram substitution. Table 2 summarizes our n-gram substitution rules.

N-grams	Prefix	Suffix	Any
CA			KA
CC,CK			KK
CE			SE
CH		KK	
CHL,CL			KL
CHR, CR			KR
CI			SI
CO			KO
CS,CZ,TS,TZ	SS		
CU			KU
CY			SY
DG			GG
GH			HH
GN	NN		
HR,WR	RR		
HW	WW		
KN, NG	NN		
MAC, MC			MK
NST			NSS
NT		TT	
PF, PH			FF
RT, RDT		RR	
SCH			SSS
TIO, TIA			SIO
TCH			CHH

Table 2. N-gram substitution.

Table 3 summarizes the translation of letters to numeric codes. Code translation occur after letter substitution.

	Russell	Celko	Pfeifer	Fuzzy
A,E,I,O,U		A**		
B,P	1	B,P	1	1
C	2	C	2	9
D,T	3	T	3	3
F,V	1	F,V	7	1
G,J,K,Q	2	G,J,C,G	2	7
H,W,Y		H,W,Y		
L	4	L	4	4
M,N	5	N	5	5
R	6	R	6	6
S,Z	2	S	8	9
X	2	X	8	7

Table 3. Letter translations.

3.1 Fusion of Search Techniques

Like Pfiefer, our approach fuses search techniques and scoring measures. We fuse Fuzzy, Celko and Russell and produce codes of multiple lengths, utilization as many as ten codes per name. Table 4 provides some examples.

Soundex Type	Kristen	Krissy	Christen
Fuzzy	K6935	K6900	K6935
Fuzzy	K693	K690	K693
Fuzzy	K69	K69	K69
Fuzzy	K6	K6	K6
Celko	KRST	KRSY	CRST
Celko	KRS	KRS	CRS
Celko	KR	KR	CR
Russell	K623	K620	C623
Russell	K62	K62	C62
Russell	K6	K6	C6

Table 4. Fuzzy soundex example.

We implement fusion based similarity by using common phonetic codes and digrams in the Dice Co-Efficient.

$$\delta = \frac{(2\chi)}{(\alpha + \beta)}$$

Where:
δ= Similarity score
χ= Common features
α=Features for name 1
β=Features for name 2

3.2 Code Shift

We introduce code shifting to reduce Damerau insertion and omission errors not fully addressed by the base search algorithms. By using multiple length codes, we adapt to errors found near the end of the name. To address errors near the beginning we create an additional code that removes the second position from the five byte fuzzy soundex. This technique catches many errors found near the beginning of the name without greatly increasing result set size.

4 Results

All tests used ANSI standard, unchanged SQL. The small corpus prevents meaningful run time performance testing, but the SQL requests are similar in complexity to the relational IR SQL proposed by Grossman, McCabe and Lundquist. Therefore, it is reasonable to estimate scaleable, rapid run time performance.

The surname corpus, named COMPLETE, presented by [Pfeifer3] is the source. The corpus contains 14,972 distinct surnames representing a number of cultures and data sources. Data sources include; AP newswire, ACM abstracts, Federal Register, Wall Street Journal, ZIFF-Davis Publishing, University of Dortmund phone book and a bibliographic database. The source includes ninety names used as queries, each one having a judged set of relevant, related names. There are 1,187 relevant variations for the 90 query names. The evaluation program used at the Text Retrieval Conferences [Harman96] scored responses in terms of precision and recall.

4.1 Soundex Only Results

Table 5 summarizes the test results for soundex retrieval without digrams. The widely used Russell algorithm has the worst precision. Despite missing many relevant names, the Celko algorithm has better precision because it avoids returning irrelevant names. Fuzzy and Fusion tests have the best recall and when combined with a similarity metric missing in Russell and Celko, the average precision is also better.

	Relevant Recalled	Avg Precision
Russell	658	0.3155
Celko	455	0.3398
Fuzzy	927	0.4574
Fusion	1010	0.5356

Table 5. Soundex results.

4.2 Soundex with Digram Results

Table 6 summarizes results for soundex retrieval with digrams. A suffix of -N indicates that a count of matching digrams was used for ranking results. The suffix -NS indicates the matching digram and soundex counts were added to produce a relevance score. Adding an ngram based similarity metric substantially improved the Russell and Celko algorithms. The integrated does even better and code shifting raises recall to 96% -- 1140 of 1187 relevant names are retrieved.

	Relevant Recalled	Avg Precision
Russell-N	658	0.5214
Celko-N	453	0.4315
Fuzzy-N	926	0.5518
Fusion-NS	1010	0.6951
Fusion, with Code Shift	1140	0.7071

Table 6. Digram results.

4.3 System Performance

The COMPLETE corpus is too small to project run time performance for a production system. [McCabe00] conducted performance testing for related work on information retrieval. The experiments performed keyword searches against a document collection of over 500,000 documents. The environment used the Teradata RDBMS, Windows NT and a four processor Pentium Pro 200 Mhz SMP. Single queries of comparable complexity to name searches described in this paper typically ran in less than 5 seconds.

5 Conclusions and Future Work

The experiments showed the benefit of integrating multiple phonetic algorithms by improving precision and recall. The well known Russell soundex retrieves only 658 of the 1,187 relevant names for the search criteria.

Our algorithm did somewhat better than other individual tests, but the best recall came from a fully integrated test. Using all of the phonetic algorithms, code shifts and digrams raised recall to 96%. Perhaps the integrated test faired well because it addresses multiple issues such as substitution, insertion and omission errors that are well documented problems.

Clearly, a similarity metric improves precision, even when a single code per name limits recall. Once again, the best results came with an integrated test.

6 Acknowledgements

Special thanks to David Grossman and Ophir Frieder for their time and valuable insight.

7 References

[NCR98] NCR. State of Texas Selects NCR Data Warehouse Solution to Help it Collect $43 Million in Additional Taxes, *News Release*, May 18, 1998.

[Celko95] J. Celko. Joe Celko's SQL For Smarties: Advanced SQL Programming. *Morgan Kaufmann Publishers, Inc*., 1995.

[Pfeifer3] U. Pfeifer, T. Poersch, N. Fuhr. Searching Proper Names in Databases.

[Zamora81] E. Zamora, J. Pollock, A. Zamora. The use of Trigram Analysis for Spelling Error Detection. *Information Processing and Management 17(6)*, pages 305-316, 1981.

[Angell83] R. Angell, G. Freund, P. Willet. Automatic Spelling Correction using a Trigram Similarity Measure. *Information Processing and Management 19(4)*, pages 255-261, 1983.

[Harman96] D. Harman. The Fourth Text Retrieval Conference (TREC-4). *NIST Special Publication 500-236*, 1996.

[Damerau64] F. Damerau. A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM 7*, pages 171-176, 1964.

[McCabe00] M. C. McCabe, D. Holmes, D. Grossman, O. Frieder. Parallel, Platform Independent Implementation of Information Retrieval Algorithms. *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA),* 2000.

[Grossman98] D. Grossman, Frieder O. Information Retrieval: Algorithms and Heuristics. *Klewer Academic Publishers*, 1998.

[McCabe99] M. C. McCabe, Chowdury A., Grossman D., Frieder O. A Unified Environment for Fusion of Information Retrieval Approaches. International Conference on Information Knowledge Management, 1999.

[Lundquist99] C. Lundquist, Frieder O., Holmes D., Grossman D. A Parallel Relational Database Management System Approach to Relevance Feedback in Information Retrieval. Journal for the Society of Information Sciences, 1999.