Soundex

From Wikipedia, the free encyclopedia

This article is about the phonetic algorithm. For the Rock n' Soul band, see the SoundEx.

Soundex is a phonetic algorithm for indexing names by sound, as pronounced in English. The goal is for homophones to be encoded to the same representation so that they can be matched despite minor differences in spelling.[1] The algorithm mainly encodes consonants; a vowel will not be encoded unless it is the first letter. Soundex is the most widely known of all phonetic algorithms (in part because it is a standard feature of popular database software such as DB2, PostgreSQL,[2] MySQL,[3] Ingres, MS SQL Server[4] and Oracle[5]) and is often used (incorrectly) as a synonym for "phonetic algorithm".[*citation needed*] Improvements to Soundex are the basis for many modern phonetic algorithms.[6]

History [edit]

Soundex was developed by Robert C. Russell and Margaret King[7] Odell and patented in 1918[8] and 1922.[9] A variation called **American Soundex** was used in the 1930s for a retrospective analysis of the US censuses from 1890 through 1920. The Soundex code came to prominence in the 1960s when it was the subject of several articles in the *Communications* and *Journal of the Association for Computing Machinery*, and especially when described in Donald Knuth's *The Art of Computer Programming*.[10]

The National Archives and Records Administration (NARA) maintains the current rule set for the official implementation of Soundex used by the U.S. Government.[1] These encoding rules are available from NARA, upon request, in the form of General Information Leaflet 55, "Using the Census Soundex".

American Soundex [edit]

The Soundex code for a name consists of a letter followed by three numerical digits: the letter is the first letter of the name, and the digits encode the remaining consonants. Consonants at a similar place of articulation share the same digit so, for example, the labial consonants B, F, P, and V are each encoded as the number 1.

The correct value can be found as follows:

1. Retain the first letter of the name and drop all other occurrences of a, e, i, o, u, y, h, w.
2. Replace consonants with digits as follows (after the first letter):
   - b, f, p, v → 1
   - c, g, j, k, q, s, x, z → 2
   - d, t → 3
   - l → 4
   - m, n → 5
   - r → 6
3. If two or more letters with the same number are adjacent in the original name (before step 1), only retain the first letter; also two letters with the same number separated by 'h' or 'w' are coded as a single number, whereas such letters separated by a vowel are coded twice. This rule also applies to the first letter.
4. Iterate the previous step until you have one letter and three numbers. If you have too few letters in your word that you can't assign three numbers, append with zeros until there are three numbers. If you have more than 3 letters, just retain the first 3 numbers.

Using this algorithm, both "Robert" and "Rupert" return the same string "R163" while "Rubin" yields "R150". "Ashcraft" and "Ashcroft" both yield "A261" and not "A226" (the chars 's' and 'c' in the name would receive a

Wikipedia The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
Español
Français
हिन्दी
עברית
Nederlands
Polski
Русский
Српски / srpski
ไทย
Türkçe
Українська
Tiếng Việt
中文
Edit links

single number of 2 and not 22 since an 'h' lies in between them). "Tymczak" yields "T522" not "T520" (the chars 'z' and 'k' in the name are coded as 2 twice since a vowel lies in between them). "Pfister" yields "P236" not "P123" (the first two letters have the same number and are coded once as 'P').

Variants [edit]

A similar algorithm called "Reverse Soundex" prefixes the last letter of the name instead of the first.

The NYSIIS algorithm was introduced by the New York State Identification and Intelligence System in 1970 as an improvement to the Soundex algorithm. NYSIIS handles some multi-character n-grams and maintains relative vowel positioning, whereas Soundex does not.

Daitch–Mokotoff Soundex (D–M Soundex) was developed in 1985 by genealogist Gary Mokotoff and later improved by genealogist Randy Daitch because of problems they encountered while trying to apply the Russell Soundex to Jews with Germanic or Slavic surnames (such as Moskowitz vs. Moskovitz or Levine vs. Lewin). D–M Soundex is sometimes referred to as "Jewish Soundex" or "Eastern European Soundex",[11] although the authors discourage the use of these nicknames. The D–M Soundex algorithm can return as many as 32 individual phonetic encodings for a single name. Results of D-M Soundex are returned in an all-numeric format between 100000 and 999999. This algorithm is much more complex than Russell Soundex.

As a response to deficiencies in the Soundex algorithm, Lawrence Philips developed the Metaphone algorithm in 1990 for the same purpose. Philips developed an improvement to Metaphone in 2000, which he called Double Metaphone. Double Metaphone includes a much larger encoding rule set than its predecessor, handles a subset of non-Latin characters, and returns a primary and a secondary encoding to account for different pronunciations of a single word in English. Philips created Metaphone 3 as a further revision in 2009 to provide a professional version that provides a much higher percentage of correct encodings for English words, non-English words familiar to Americans, and first and last names found in the United States. It also provides settings that allow more exact consonant and internal vowel matching to allow the programmer to focus the precision of matches more closely.

See also [edit]

- Phonetic algorithm
- Metaphone
- New York State Identification and Intelligence System
- Match Rating Approach

References [edit]

1. ^ [a] [b] "The Soundex Indexing System". National Archives and Records Administration. 2007-05-30. Retrieved 2010-12-24.
2. ^ "PostgreSQL: Documentation: 9.1: fuzzystrmatch". *postgresql.com*. Retrieved 2012-11-03.
3. ^ "MySQL :: MySQL 5.5 Reference Manual :: 12.5 String Functions - SOUNDEX". *dev.mysql.com*.
4. ^ "SOUNDEX (Transact-SQL)". *msdn.microsoft.com*. Retrieved 2012-11-03.
5. ^ "Soundex". *docs.oracle.com*. Retrieved 2012-11-03.
6. ^ "Phonetic Matching: A Better Soundex". Retrieved 2012-11-03.
7. ^ Odell, Margaret King (1956). "The profit in records management". *Systems* (New York) **20**: 20.
8. ^ US patent 1261167, R. C. Russell, "(untitled)", issued 1918-04-02 (Archived)
9. ^ US patent 1435663, R. C. Russell, "(untitled)", issued 1922-11-14 (Archived)
10. ^ Knuth, Donald E. (1973). *The Art of Computer Programming: Volume 3, Sorting and Searching*. Addison-Wesley. pp. 391–92. ISBN 978-0-201-03803-3. OCLC 39472999.
11. ^ Mokotoff, Gary (2007-09-08). "Soundexing and Genealogy". Retrieved 2008-01-27.

Categories: Phonetic algorithms