Paper 072-29

Matching Character Variables by Sound: A closer look at SOUNDEX function and Sounds-Like Operator (=*)

Zizhong Fan, Westat, Rockville, MD

Abstract

The Soundex phonetic filing system has been used for tracking genealogical data for many years. Many database systems, such as Microsoft® SQL Server, and Oracle®, have adopted the SOUNDEX as a query tool for retrieving character data based on their phonetic values. SAS® has included the SOUNDEX function and sounds-like operator (=*) since version 6.07 in both the DATA step and SQL procedure (PROC SQL). But phonetic matching is still relatively new to many SAS professionals. This paper will take a closer look at the function SOUNDEX and the sounds-like operator (=*). SOUNDEX function can be used to adjust the sensitivity level for matching character values based on their phonetic similarities, whereas the sounds-like operator provides exact phonetic value matching. Examples will be shown to demonstrate the power of these two phonetic matching tools in matching character values. Differences between SOUNDEX function and sounds-like operator (=*) will also be discussed in table lookup tasks. SAS product: Base SAS. Skill Level: All skill levels.

About SOUNDEX system

Speaking of SOUNDEX phonetic filing system, we need to trace its genealogy to genealogy. In the late nineteenth century, one of the things that frustrated genealogists was that it was very difficult to trace genealogical data by surnames on census data because of the inconsistencies of spelling over years. Smith could be Smyth, Johnson could be Johnsoon, and Fan could be Fann. There were lots of reasons for these kinds of inconsistencies. One of them could be that our ancestors knew nothing about English at all. They were immigrants from all over the world. Fortunately, in 1918, Robert C. Russell developed SOUNDEX system (US patent 1,261,167). The idea was to index words phonetically rather than alphabetically. From then on, SOUNDEX phonetic filling has been widely used in census and genealogical studies. The three misspelling examples listed above could be matched up by the SOUNDEX system. In other words, the Smith, Johnson and Fan families now can easily look up their ancestors' census data by matching up the sound of their surnames.

American Soundex Coding Rule

After modifications, the current American SOUNDEX code consists of a letter and three numbers, such as W252. The letter is always the first letter of the word. The numbers are assigned to the remaining letters of the string according to the SOUNDEX guide shown below. Zeroes are added at the end if necessary to produce a four-character code. Additional letters are disregarded. Examples:

- **Washington** is coded W252 (W, 2 for the S, 5 for the N, 2 for the G, remaining letters disregarded).
- **Lee** is coded L000 (L, 000 added).

Soundex Coding Guide

Number	Represents the Letters
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R

Disregard the letters A, E, I, O, U, H, W, and Y.

Additional Soundex Coding Rules

1. **Double Letters:** If the string has any double letters, they should be treated as one letter. For example:
   - **Gutierrez** is coded G362 (G, 3 for the T, 6 for the first R, second R ignored, 2 for the Z).

2. **Same Code Letters Side-by-Side:** If the string has different letters side-by-side that have the same number in the SOUNDEX coding guide, they should be treated as one letter. Examples:
   - **Pfister** is coded as P236 (P, F ignored, 2 for the S, 3 for the T, 6 for the R).
   - **Jackson** is coded as J250 (J, 2 for the C, K ignored, S ignored, 5 for the N, 0 added).

According to the rule, Smith/Smyth both have the same soundex code as S530; Johnson/Jahnsoon both are J525; and Fan/Fann are F500.

SOUNDEX function and Sounds-like operator in SAS

SAS system has included SOUNDEX function and sounds-like operator (=*) since version 6.07 in both DATA step and SQL procedure (PROC SQL). But the SOUNDEX code generated by SAS is different from the standard American SOUNDEX code. The differences are as follows.

1. SAS SOUNDEX function generates all the possible codes for the string. For example, **Washington** has a code of **W25235** instead of W252. Even if there are spaces in the string, SOUNDEX function will generate the SOUNDEX code for the entire string. So Hong Kong is coded as H52252. Notice here, 5 is for n, 2 is for g, the next 2 is coded because of the space between the g and K, in this case, the Side-by-Side rule does not apply.
2. No additional zeros will be added. **Jackson** is coded as **J25** instead of J250. There are no supplemental zeros being added to make it a four-character code.

Sounds-like operator (=*) can be considered as the equal sign (=) between two SOUNDEX codes. For example, City1 =* City2 is the same as SOUNDEX (City1) = SOUNDEX (City2). It has to be pointed out that when sounds-like operator is used in DATA step, it can only be used in the WHERE statement.

Example

Here is an example simulated from the National Household Travel Survey (NHTS) project. The data set Travel has three variables: TripID, Departure (Departure City), and Arrival (Arrival City). Some of the trips are apparently wrong because the Departure City is the same as the Arrival City (trip 2 through trip 6). So our task here is to identify the wrong trips. But some of the city names are somehow misspelled: trip 3 through trip 6. What a lousy typist!

Travel

TripID	Departure	Arrival
1	Paris	Washington
2	London	London
3	London	Londonn
4	Tokyo	Tokio
5	Philadelphia	Filadelfia
6	Beijing	Peking

If we use the DATA step below, we will only be able to identify trip 2 ('London' = 'London').

```
data SameCity;
  set Travel;
  where Departure = Arrival;
run;
```

SOUNDEX function and sounds-like operator come in handy in situations like this. Let's take a closer look by laying out the SOUNDEX code for the city names

SOUNDEX codes for Departure and Arrival

Trip ID	Departure	Arrival	SOUNDEX (Departure)	SOUNDEX (Arrival)
1	Paris	Washington	P62	W25235
2	London	London	L535	L535
3	London	Londonn	L535	L535
4	Tokyo	Tokio	T2	T2
5	Philadelphia	Filadelfia	P4341	F4341
6	Beijing	Peking	B252	P252

So let's first try the sounds-like operator =*.

```
data SameCity;
  set Travel;
  where Departure =* Arrival;
run;
```

We will get two more trips: trip 3 ('L535') and trip 4 ('T2'). Here, Departure =* Arrival is equivalent to SOUNDEX (Departure) = SOUNDEX (Arrival).

What about trip 5 and trip 6? They have the same characteristic, which is the first letter is different, but the following digit codes are the same. So a combination of SOUNDEX and SUBSTR will fix this problem. This time, let's use the SQL procedure.

```
proc sql;
  create table SameCity as
  select * from Travel
  where SUBSTR (SOUNDEX (Departure), 2) = SUBSTR (SOUNDEX (Arrival), 2);
quit;
```

Now we get them all including trip 5 and 6. This shows we can adjust the sensitivity level by using the SOUNDEX with SUBSTR to determine which positions and how many positions we want to compare, in other words, how fuzzy you allow the match to be.

Conclusion

SOUNDEX function and sounds-like operator =* are very useful when fuzzy matching of character values is needed. They match character strings based on their phonetic values. The sensitivity level of the matching can be adjusted by the combination of SOUNDEX and SUBSTR functions.

Reference
SAS Guide to the SQL procedure: Usage and Reference, Version 6, First Edition; SAS Institute, Cary, NC, USA

The Soundex Indexing System: February 19, 2000 http://www.archive.gov.research_room/genealogy/census/soundex.html

2001 National Household Travel Survey (NHTS); US Department of Transportation

Contact Information
Zizhong Fan
Westat
1650 Research Boulevard
Rockville, MD 20850
(240) 314 -2486
E-mail: JamesFan@westat.com