SAS® GLOBAL FORUM 2018

USERS PROGRAM

Phonetic Search Helps To Fight Risks

Yaobin CHEN, Walt HUI, HSBC

April 8 – 11 | Denver, CO

#SASGF

RESTRICTED

Phonetic Search Helps To Fight Risks

Yaobin CHEN, Walt HUI, HSBC

ABSTRACT

Phonetic Search is a solution we built using the SAS Technology. It can be used across various areas to detect or identify information that is similar or identical e.g. names with exact pronunciation but spelt differently (Elisabeth and Elizabeth). In our day to day work, we may not receive exact or complete information which makes searching based on exact spelling a challenge. To overcome this, we developed a search capability based on **'sounds like'** and **'spelling distance'** to improve search accuracy and search efficiency.

Basically, Phonetic Search makes use of two algorithms available in SAS. The first one is a customized **Soundex** algorithm and the other one is a fuzzy match using the **Levenshtein Distance** algorithm associated with scores to improve the accuracy. It enables searching with higher efficiency in less time because the search complexity has been encapsulated in an abstraction layer of the program. As a result, users are able to get more accurate results in a shorter time.

WHY APPLY PHONETIC SEARCH

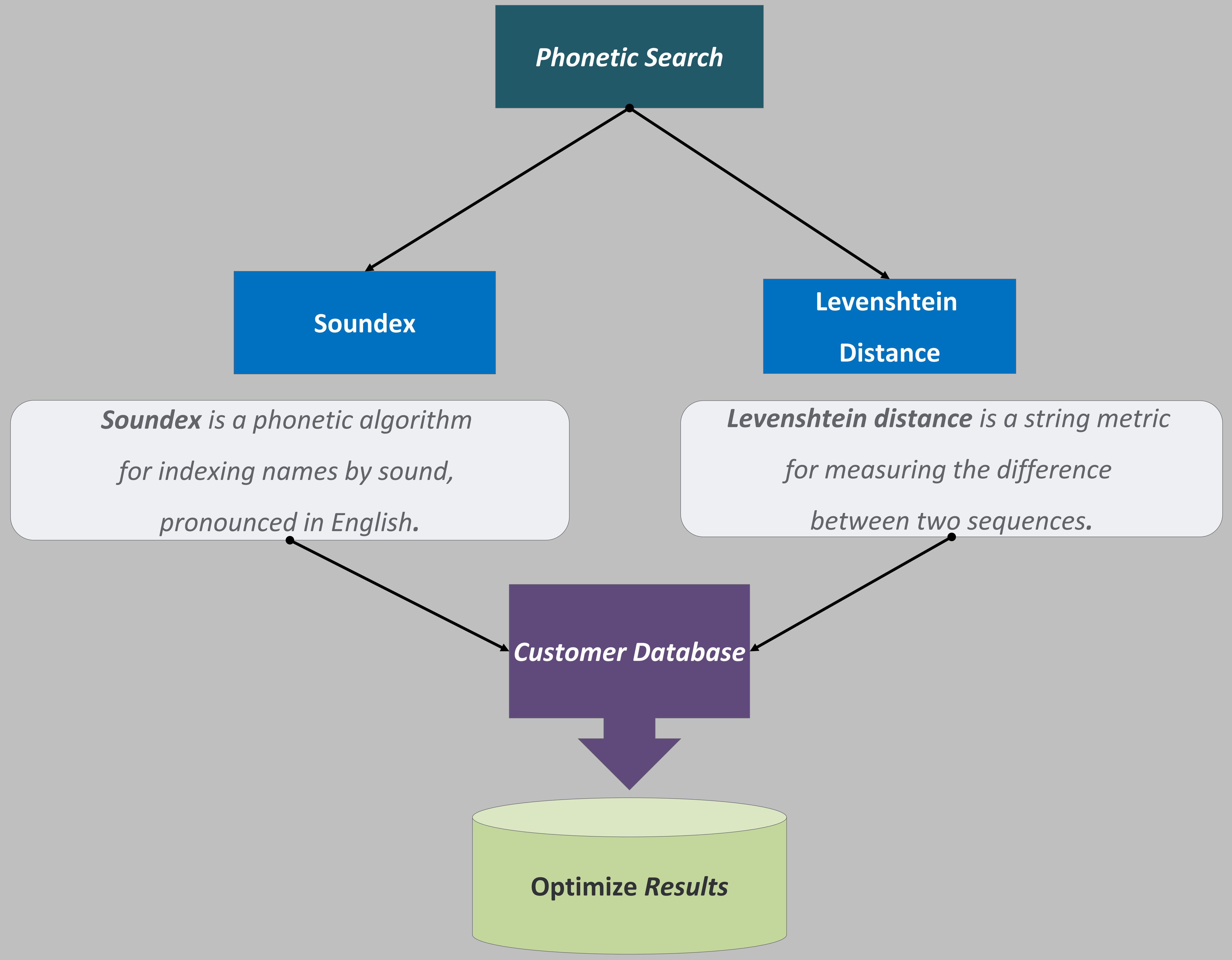
- **User misspelling**
- Avoids losing any observed records in case of misspelling by users.

- **Data issues**
- Overcomes limitation that search results might not be extracted in case inputting users makes other spelling forms or any typo during data entry.

- **Searching accuracy & efficiency**
- Improves accuracy and search hit rates.

- **Reduces cost**
- Manual efforts are reduced and takes less time for results to be available.

METHODS



Phonetic Search

Soundex

Levenshtein Distance

***Soundex** is a phonetic algorithm for indexing names by sound, pronounced in English.*

***Levenshtein distance** is a string metric for measuring the difference between two sequences.*

Customer Database

Optimize *Results*

Phonetic Search Helps To Fight Risks

Yaobin CHEN, Walt HUI, HSBC

EXAMPLES

Names	Encodes
Tracie Chen	T62 C5
Tracy Chan	T62 C5
Tracey Chen	T62 C5
Jason Macdonald	J25 M235
Jaison McDonnell	J25 M235
Jayson McDonald	J25 M235
Elizabeth Smyth	E421 S53
Elisabeth Smith	E421 S53
Elizabeth Smythe	E421 S53

$$\textbf{Similarity} = 1 - \frac{\textit{edit distance}}{\textit{max.length}(\textit{word1,word2})}$$

CONCLUSIONS

Phonetic Search is a user friendly tool. It increases work efficiency and accuracy. Specifically, the two algorithms are really helpful and powerful. Throughout the whole research, it was found that both algorithms improved the accuracy of the outcome but in a different way. Therefore, this program applies both a phonetic algorithm and Levenshtein distance for helping users correct typed input. The following shows the pros and cons of each algorithm.

Soundex Algorithm:

Phonetic comparison

- Works well with standard English words.
- General soundex rules have limitation to apply for all languages.
- This method is fast, but accuracy may be lower if we want to see most likely match.

Levenshtein Distance:

Spelling comparison

- Better for catching typos such as repeated letters, transposed letters, or hitting the wrong key.
- Allows customization of spelling distances/scores, more flexible to enhance the searching ability by adjusting the criteria or threshold.
- Accuracy is higher, but low efficiency in program execution.

REFERENCES

1. Fan, Zizhong. 2004. Matching Character Variables by Sound: A closer look at SOUNDEX function and SoundsLike Operator . Rockville, MD: Westat.
2. Roesch, Amanda. "Matching data using sounds-like operators and SAS® compare functions." In *SAS, ed. SAS Global Forum*, vol. 2012, pp. 1-11. 2012.
3. Navarro, Gonzalo. "A guided tour to approximate string matching." *ACM computing surveys (CSUR)* 33, no. 1 (2001): 31-88.
4. https://en.wikipedia.org/wiki/Levenshtein_distance

Phonetic Search Helps To Fight Risks

Appendix - How the Soundex algorithm works ?

Soundex Algorithm:

1. Capitalize all letters in the word and drop all punctuation marks.
2. Retain the first letter of the word.
3. Convert each letter to a code like below:

A	E	I	O	U	H	W	Y	0
B	F	P	V					1
C	G	J	K	Q	S	X	Z	2
D	T							3
L								4
M	N							5
R								6

4. Remove all pairs of digits which occur beside each other from the string.
5. Remove all zeroes from the string that results from step 4.

Enhancements / Customization of the Algorithm:

Several adjustments have been made to improve the original Soundex algorithm to make the results more robust such as transform the word GH to H, KN to N, TCH to CH, CAL to Ke etc.

Names	Encodes
Tracie Chen	T62 C5
Tracy Chan	T62 C5
Tracey Chen	T62 C5
Jason Macdonald	J25 M235
Jaison McDonnell	J25 M235
Jayson McDonald	J25 M235
Elizabeth Smyth	E421 S53
Elisabeth Smith	E421 S53
Elizabeth Smythe	E421 S53

Phonetic Search Helps To Fight Risks

Appendix - How the Levenshtein algorithm works ?

Levenshtein Algorithm:

Minimum number of edit steps required to change one word into the other.

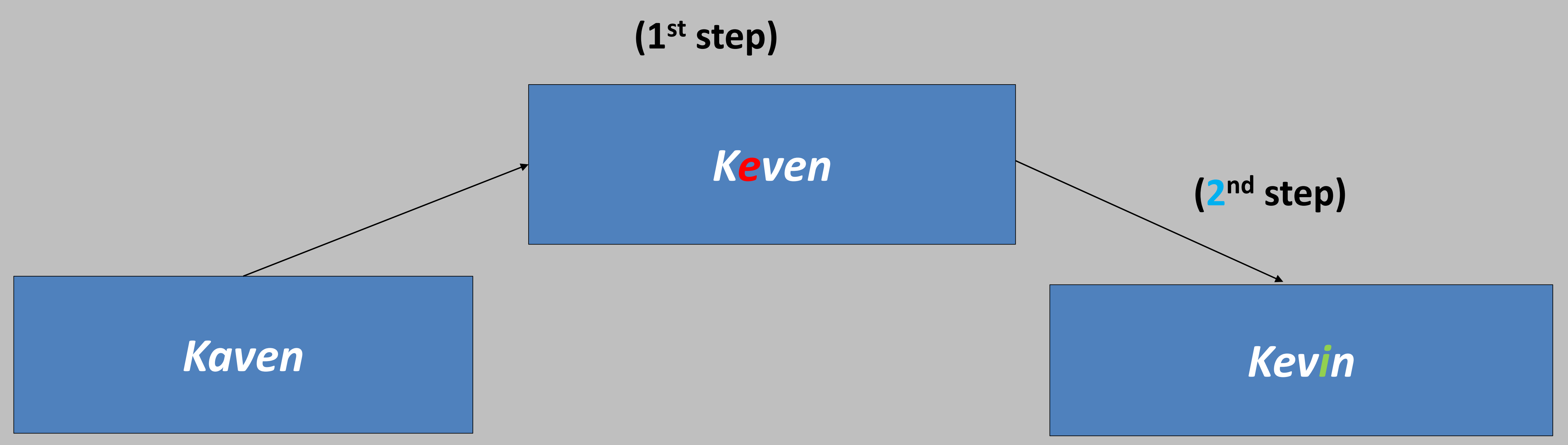
E.g.. insertions, deletions or substitutions.

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if} \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

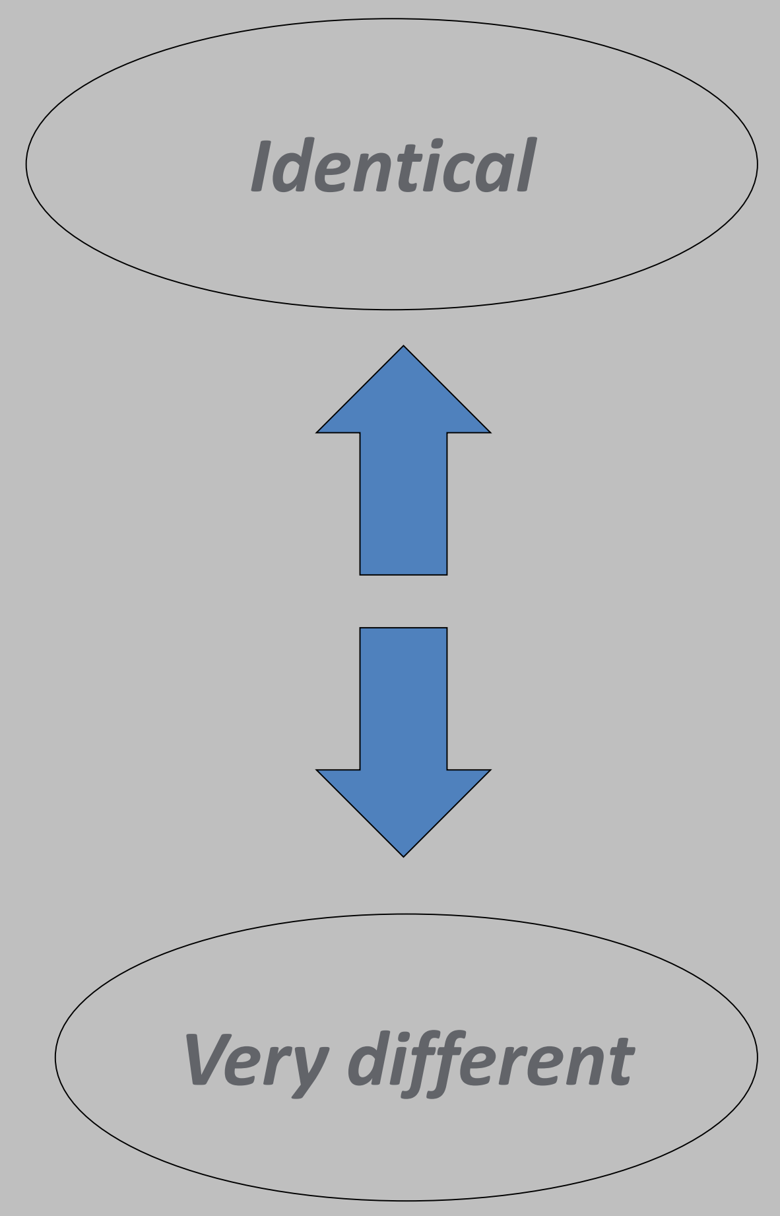
	Edit Distance Score					
		k	a	v	e	n
	0	1	2	3	4	5
k	1	0	1	2	3	4
e	2	1	1	2	2	3
v	3	2	2	1	2	3
i	4	3	3	2	2	3
n	5	4	4	3	3	2(steps)

Example:

- **Example, the Levenshtein distance between "Kaven" and "Kevin" is 2:**



Edit Distance Score
No Change
Delete one of a double letter
Double a letter
Swap two consecutive letters
Delete a letter from the end
Add a letter to the end
Delete a letter from the middle
Insert a letter in the middle
Replace a letter in the middle
Delete the first letter
Insert a letter at the beginning
Replace the first letter



SAS® GLOBAL FORUM 2018

April 8 – 11 | Denver, CO
Colorado Convention Center

#SASGF