IndicSOUNDEX Algorithm for Text Matching

Christopher DiPersio
dipersio@amazon.com
Amazon Alexa AI
Cambridge, Massachusetts, United States

ABSTRACT

Information retrieval in multilingual systems poses a challenging problem; this challenge is exacerbated when the component languages do not share the same script and writing system. These differences make indexing names across scripts incredibly difficult or even impossible, and more languages that are part of the system make the problem worse and searches less reliable. This paper describes a new, SOUNDEX-based approach, called IndicSOUNDEX, that attempts to alleviate such problems for an information retrieval system that includes Hindi, Marathi, Telugu, Tamil, Malayalam, Punjabi, Bengali, Kannada, Gujarati, and English by collapsing spelling differences in phonetically related words. We examine IndicSOUNDEX's strengths in handling word pairs written in two different Indic scripts, one written in Indic script and the other in Latin script, and from different phonemic representations.

1 INTRODUCTION

A key problem in information retrieval is how to neutralize potential spelling differences or errors for a given name or word in order to return an appropriate result. For monolingual systems, a SOUNDEX or SOUNDEX-like algorithm can reduce those differences by converting the word to a SOUNDEX representation shared by different spellings of that word. However, this same problem becomes much more complicated when the information retrieval system must handle multiple languages and multiple scripts, as that same SOUNDEX approach must now accommodate many more sounds and text representation differences. One such locale where this problem is particularly present is India where individuals usually speak two or more languages in their daily lives. To address this problem, we created a truly multilingual and multi-script SOUNDEX implementation designed to accommodate Hindi, Marathi, Telugu, Tamil, Bengali, Punjabi, Kannada, Gujarati, and Malayalam as well as English, and their nine individual scripts. We call this implementation IndicSOUNDEX.

2 RELATED WORK

Most work on SOUNDEX algorithms has focused primarily on monolingual solutions, with one of the best known implementations with a multilingual component being Beider-Morse Phonetic Matching [1]. However, even this implementation is not truly multilingual since it only identifies the language in which a given word is written to choose which pronunciation rules to apply. Other attempts at multilingual SOUNDEX algorithms, particularly for Indic languages, were mostly small in scope covering two Indic languages with or without English as an included third language. Chaware and Rao [2] tested custom SOUNDEX and Q-gram implementations against their own matching technique called Indic-Phonetic on Hindi and Marathi word pairs. Their custom SOUNDEX implementation only matched 34% of Hindi pairs and 14% of Marathi pairs. Both Q-gram and Indic-Phonetic performed better matching 68% and 76% of Hindi pairs and 46% and 38% of Marathi pairs, respectively. However, their test sets were purely monolingual, meaning their word pairs were made of either two Hindi words or two Marathi words. Shah and Singh [5] describe an actual multilingual SOUNDEX implementation for Hindi, Gujarati, and English. They tested their algorithm on 100 names written in the three languages and achieved perfect accuracy; however, this seems to be, in part, attributable to a matching threshold; their system will declare a pair as a match even if the resulting SOUNDEX representations for both words differ in up to two characters. In contrast to this previous work, IndicSOUNDEX is built to be completely multilingual, supporting conversion of words in the aforementioned nine Indic languages and English all at once.

3 ALGORITHM

Similar to other SOUNDEX implementations, the IndicSOUNDEX algorithm follows a number of steps to convert a token from its original form into a SOUNDEX representation. However, unlike other implementations, IndicSOUNDEX uses a specialized character-to-character mapping system for each of the eight Indic scripts used by the nine Indic languages with separate treatment for English which is discussed in section 3.3. The first step of this process is to convert each character to an intermediate representation in the form of a partial Unicode point derived from that character's relative position in its script's Unicode block. This is very similar to International Components for Unicode's (ICU) method for transliterating text from Indic scripts using a form they call Inter-Indic [3]. Unicode blocks for Indic scripts share the same dimensions (eight rows by sixteen columns) and relative positioning of similar sounding characters. For example, the Devanagari character "क" (at U+0915), Telugu character "క" (at U+0C15), Gujarati character "ક" (at U+0A95), and Tamil character "க" (at U+0B95) (all representing /k/) appear in the second row and sixth column of their respective blocks. From the four Unicode points for each character, we can derive two identical partial points using the final two characters in each point: "15" for Devanagari and Telugu, "95" for Gujarati and Tamil. Performing this same operation for all characters across all eight Indic scripts, we derive two groups of matching partial Unicode points: an "upper" and "lower" partial block, shown in table 1.

Following this scheme, we can use partial points to easily map all characters that represent the same sound across the eight scripts by converting a given character to a string representation of its hexadecimal Unicode point and using the final two characters to look up the corresponding converted representation. This reduces

Table 1: ”Upper” and ”lower” Indic script Unicode blocks

Block Row	Upper (U+_ _1x → U+_ _7x)	Lower (U+_ _8x → U+_ _Fx)
9	Devanagari	Bengali
A	Gurmukhi (Punjabi)	Gujarati
B		Tamil
C	Telugu	Kannada
D	Malayalam	

Table 2: ISO-15919 conversion comparison

Latin	Hindi	Kannada	Malayalam	Tamil
suresh	सुरेश (surē**ś**)	ಸುರೇಶ್ (sur**ś**)	സുരേഷ് (surē**ṣ**)	
thamarai	थमाराई (**th**mārāī)			தாமரை (**t**āmrai)

Table 3: Modified ISO-15919 conversion comparison

Latin	Hindi	Kannada	Malayalam	Tamil
suresh	सुरेश (surē**s**)	ಸುರೇಶ್ (sur**s**)	സുരേഷ് (sure**s**)	
thamarai	थमाराई (**t**maraī)			தாமரை (**t**amrai)

the number of character-to-character mappings from one for each of the eight scripts to one for each of the two partial blocks.

3.1 Character Mappings

IndicSOUNDEX converts a character from its above described intermediate form into one of two final forms, depending on its position in the original token:

(1) Modified ISO-15919 – A customized version of the Romanization scheme for transliterating text written in Indic scripts to Latin script; used for the first character in a token
(2) IndicSOUNDEX Codes – A set of fifteen groups of characters denoting to which alphanumeric code a given partial Unicode point should be converted; used for all remaining characters in a token

3.1.1 Modified ISO-15919. Like American SOUNDEX, IndicSOUNDEX does not convert the first letter of a given token to a SOUNDEX code since word-initial sounds tend to be phonologically prominent and rarely confusable (with some exceptions). However, unlike American SOUNDEX, IndicSOUNDEX cannot leave the first letter unchanged as it needs to accommodate multiple scripts, so it converts the first letter to a neutral form to facilitate cross-language matching. We used ISO-15919 as base for this form because it creates a consistent, near one-to-one mapping from Indic to Latin characters. However, unaltered ISO-15919 retains too many distinctions on certain phonological features that differ in use between the languages, such as aspiration. The ISO-15919 standard usually realizes these differences as diacritics on a given character or as the second character in a pair of characters, shown in table 2. A simple removal of the diacritics or second character in a pair is enough to strike a balance between specificity and reduction, shown in table 3.

3.1.2 IndicSOUNDEX codes. The American SOUNDEX mapping is very reductive, grouping sets of sounds based on their representation in orthography, which in English is fairly variable. Some of these sets, particularly the second of the six in American SOUNDEX which includes “c”, “g”, “j”, “k”, “q”, “s”, “x”, and “z”, are quite large. Such reductionism is dangerous when mixing many languages together due to the dramatic increase in vocabulary and false positive rate, which can be as high as 95% for American SOUNDEX [1]. Instead, we based IndicSOUNDEX encoding on linguistically-motivated distinctive features representing the manner of articulation for a given sound, within the context of the nine Indic languages. For English, we used the best approximation, which in some cases meant mapping a single English graph onto two IndicSOUNDEX codes, e.g. “x” → “4s”. Plosive, affricate, lateral, and retroflex consonants keep most of their distinctions based on place of articulation and voicing, but forms differing in other phonological features (aspirated vs. non-aspirated, etc.) are merged into a single code. Fricatives keep all distinctions. All nasals are merged into a single code due to wildly varying usage and representation. Vowels are dropped during conversion and so do not have a code. These criteria result in fifteen individual SOUNDEX codes that retain higher levels of granularity than the base American SOUNDEX encoding.

3.2 Algorithm steps

IndicSOUNDEX converts a token following the process described in algorithm 1, in conjunction with the mapping schemes described above. Table 4 shows sample output for a set of example words converted using the IndicSOUNDEX algorithm.

Algorithm 1 Convert a word to an IndicSOUNDEX representation

```
$soundexConversion \leftarrow$ ""
for $i$ in $range(len(word))$ do
    $char \leftarrow word[i]$
    if $i == 0$ then
        $soundexChar \leftarrow convertToModISO(char)$
    else
        $soundexChar \leftarrow convertToSOUNDEX(char)$
    end if
    $soundexConversion \leftarrow soundexConversion + soundexChar$
end for
$soundexConversion \leftarrow removeHAfterConsonant(soundexConversion)$
$soundexConversion \leftarrow removeNonWordInitialVowels(soundexConversion)$
$soundexConversion \leftarrow reduceConsecutiveIdenticalCodesToOneOccurrence$
```

Table 4: IndicSOUNDEX conversion example

Script	Indic Original	ISO-15919	IndicSOUNDEX
Devanagari (Hindi)	इन्दिके	indikē	i034
Devanagari (Marathi)	इतिहासाचा	itihāsācā	i8hs2
Telugu	పడినది	pḍindi	p303
Gurmukhi (Punjabi)	ਸੋਹਣ	sōhṇ	sh0

Table 5: Cross-Indic name accuracy test

Language Pair	Accuracy	Language Pair	Accuracy
Bengali/Gujarati	93.62% (44/47)	Hindi/Punjabi	88.10% (37/42)
Bengali/Hindi	92.81% (129/139)	Hindi/Tamil	60.13% (95/158)
Bengali/Kannada	94.00% (47/50)	Hindi/Telugu	100% (167/167)
Bengali/Malayalam	76.09% (35/46)	Kannada/Malayalam	83.67% (82/98)
Bengali/Marathi	94.62% (88/93)	Kannada/Marathi	100% (128/128)
Bengali/Punjabi	87.50% (28/32)	Kannada/Punjabi	83.33% (20/24)
Bengali/Tamil	45.83% (22/48)	Kannada/Tamil	57.81% (74/128)
Bengali/Telugu	93.75% (45/48)	Kannada/Telugu	100% (133/133)
Gujarati/Hindi	100% (102/102)	Malayalam/Marathi	81.52% (75/92)
Gujarati/Kannada	100% (48/48)	Malayalam/Punjabi	66.67% (16/24)
Gujarati/Malayalam	70.27% (26/37)	Malayalam/Tamil	43.93% (47/107)
Gujarati/Marathi	100% (99/99)	Malayalam/Telugu	81.25% (78/96)
Gujarati/Punjabi	83.33% (25/30)	Marathi/Punjabi	87.50% (35/40)
Gujarati/Tamil	63.41% (26/41)	Marathi/Tamil	60.68% (71/117)
Gujarati/Telugu	100% (47/47)	Marathi/Telugu	100% (122/122)
Hindi/Kannada	99.38% (159/160)	Punjabi/Tamil	58.91% (76/129)
Hindi/Malayalam	83.93% (94/112)	Punjabi/Telugu	58.33% (14/24)
Hindi/Marathi	99.70% (332/33)	Tamil/Telugu	58.91% (76/129)

3.3 Latin conversion

In addition to the eight Indic scripts, IndicSOUNDEX is also capable of converting a token written in Latin to the same phonetic-based representation. This serves two purposes:

(1) Matching the same word in a single Indic language where one version is written in the original Indic script and the other in Latin script (Latin script to represent Indic languages is commonly used on the internet and is completely unstandardized leading to multiple spellings for the same word)
(2) Matching an English word to its loanword form in an Indic language

This conversion largely follows the same steps described in section 3.2; however, due to the fact that the Unicode block containing the English alphabet subset of Latin characters is structured differently, both in terms of dimensions and character positions, IndicSOUNDEX uses a separate mapping of partial Unicode points for English. Additionally, because the core of IndicSOUNDEX centers primarily on the eight Indic scripts, there are some resulting weakness when converting native English words. For example, the

Table 6: Tamil error summary

Latin	Tamil	Paired Language	ISO-15919 Sound Difference (Tamil vs. Paired Language)
gopal	k9l (கோபால்)	g9l (গোপাল) (Bengali)	k vs. g
dinesh	t0s (தினேஷ்)	d0s (દિનેશ) (Gujarati)	t vs. d
rahul	r4l (ராகுல்)	rhl (ರಾಹುಲ್) (Kannada)	k vs. h
shiva	cv (சிவா)	sv (ശിവ) (Malayalam)	c vs. ś
bharath	pl8 (பரத்)	bl8 (ഭരത്) (Malayalam	p vs. bh

Table 7: Hindi, Marathi, Punjabi, and Telugu Indic/Latin accuracy test

Language	Total Pairs	Matching Pairs Produced by IndicSOUNDEX	Accuracy
Hindi	404	397	93.81%
Marathi	162	150	92.59%
Punjabi	200	176	88.00%
Telugu	197	186	94.42%

Table 8: Indic/Latin name accuracy test

Language	Total Pairs	Matching Pairs Produced by IndicSOUNDEX	Accuracy
Bengali	176	160	90.91%
Gujarati	102	102	100%
Hindi	9,280	8,954	96.49%
Kannada	176	175	99.43%
Malayalam	135	108	80.00%
Marathi	355	355	100%
Punjabi	44	39	88.64%
Tamil	199	117	58.79%
Telugu	188	188	100%
Overall	10,655	10,198	95.71%

Table 9: Accuracy of IndicSOUNDEX on text tokens produced by ASR, binned by count of text representations to a single phonemic representation

Count of Text Representations per Phonemic Representation	Total Text Representation Pairs in Bin	Matching Pairs Produced by IndicSOUNDEX	Accuracy
2	5,198	4,560	87.73%
3	4,509	3,818	84.68%
4	3,138	2,565	81.74%
5	1,960	1,581	80.66%
6	1,350	946	70.07%
7	1,071	764	71.34%
8	420	314	74.76%
9	540	364	67.41%
Overall	18,186	14,914	82.00%

removal of “h” following a consonant code is to reduce differences in aspiration representation in the Indic scripts. However, this also reduces English “th”, “sh”, and “ch”, which are double-graph representations of single sounds (two fricatives and an affricate), to “t”, “s”, and “c”, respectively.

Table 10: Phonemic representation accuracy test binned by string length, one edit distance substitution difference

Phonemic Representation String Length	Total Pairs	Matching Pairs Produced by IndicSOUNDEX	Accuracy
3	17,621	6,392	36.27%
4	23,255	10,313	44.35%
5	7,226	3,817	52.82%
6	2,301	1,354	58.84%
7	558	398	71.33%
8	267	234	87.64%
9	95	84	88.42%
10	40	37	92.50%
11	24	17	79.17%
12	12	12	100%
14	2	2	100%

4 EXPERIMENTS AND RESULTS

To test the IndicSOUNDEX algorithm, we took pairs of phonetically similar tokens, converted them to a SOUNDEX representation, and checked if the results are an exact match. We calculated accuracy, defined as a percentage of the number of token pairs that had matching IndicSOUNDEX representations over the total number of token pairs.

We evaluated on three test sets, each designed to evaluate a different aspect of IndicSOUNDEX:

(1) Pairs of possible of words that are both words that have the same meaning (for example, names and synonyms) and homophones across Indic languages: the assumption is that such pairs would represent the same word in two languages, rather than different words that happen to be homophones: e.g. "अमित" (Hindi), "അമിത്" (Malayalam) - both "amit" in Latin script
(2) Pairs of the same word with one written in the Indic script and the other in Latin script in the same Indic language: e.g. "amit" (Latin script), "அமித்" (Tamil script)
(3) Pairs of words with identical or similar phonemic representations but potentially different meanings: e.g. "गली" ("glī") and "गाली" ("gālī"), "जियो" ("jiyō") and "जोर" ("jōr")

4.1 Pairs of possible synonyms and homophones across languages

On a small-scale test, IndicSOUNDEX achieved 64.29% accuracy (9 matching pairs out of 14 total pairs) for Hindi and Marathi, 42.55% (20 out of 47) on Hindi and Punjabi, and 45.83% (11 out of 24) on Hindi and Telugu. Accuracy was lower than expected, but upon examination of the results, we determined it was due to morphological differences for which IndicSOUNDEX was not designed to account. For example, the Hindi word "निर्माताओं" ("nirmātāōṁ") ends in an "ōṁ" suffix (the plural marker) whereas a Punjabi counter part "ਨਿਰਮਾਤਾ" ("nirmātā") has a different ending. To see results on data unaffected by language-specific morphological changes, we reran the same test using uninflected people names sourced from behindthename.com and Parlikar [4], where each token in the pair was the same name written in each target script; these results are shown in table 5.

Different language pairs yielded different results with pairs that included Tamil being the weakest. This seems to be, in part, due to only a handful of sounds that differ in usage between Tamil and the other languages and so cause problems that are unique to Tamil paired with any of them. Table 6 shows some examples. However, overall accuracy across all pairs was high at 82.00%.

4.2 Same word written Indic script and Latin script

Table 7 shows accuracy on pairs of words in Hindi, Marathi, Punjabi, and Telugu, where the same word is represented by its native script and by Latin script. In some cases, an Indic word had more than one Latin representation because there does not exist a standardized way to spell Indic words using Latin script. Where that was the case, we created an extra pair for that Indic script word and its additional Latin representation.

We reran the same test using the name data from section 4.1 but with pairs made from the Latin and Indic script representations of the same name, shown in table 8; results were promising suggesting that IndicSOUNDEX is much better at creating a third representation when the same token is in an Indic script and Latin script. Tamil, again, was the overall weakest at 58.79% accuracy, but overall accuracy across all languages was otherwise extremely high: 95.71%.

4.3 Pairs of words with identical and similar phonemic representations

This test set is based on the output of an ASR model, which consists of a phonemic representation of each spoken word and its associated set of possible written text representations. Depending on a variety of factors, an ASR model may associate a number of different text representations with a single phonemic hypothesis, so we created sets of text representations linked by the same phonemic representation. We then ran the accuracy test binning results by the number of text tokens per phonemic representation, shown in table 9.

Results of this test showed higher accuracy for pairs of words from phonemic representations with fewer text representations, and lower accuracy from those with more text representations. Our next step was to test pairs of words derived from slightly different phonemic representations. This would simulate IndicSOUNDEX's ability to address cases where ASR either heard the word incorrectly or a user slightly misspoke. Using an edit distance algorithm, we identified phonemic representations that differed by a single substitution and recreated the token pairs. In total, 22,662 pairs out of 51,401 matched (44.09% accuracy), a dramatic decrease. To analyze the possible root causes, we binned our results by phonemic representation string length. Table 10 shows that word pairs derived from longer phonemic representations have higher accuracy rates, suggesting the longer a phonemic representation is, the better chance its words will have matching IndicSOUNDEX representations. This is both logical and desired; in shorter strings, a single sound substitution is significantly more meaningful than in longer strings; one substitution in a string of 3 characters is one third of the string, whereas one substitution in a string of 10 characters is only one tenth of the string. Also, the longer two phonemic representations are where they differ in only one substitution, the more likely it will be that their text representations are genuinely phonetically related.

5 CONCLUSIONS

We designed IndicSOUNDEX to handle spelling and script differences for content indexing in a hypothetical multilingual information retrieval system for the Indian market. In our tests, we showed our SOUNDEX implementation succeeds at its three goals.

(1) High accuracy at matching two strings if they are the same word in the same language but in two different scripts
(2) High accuracy at matching two strings if they are the same word in two different languages (although these results can be affected by extensive morphological inflection)
(3) High accuracy at matching different text representations of the same phonemic representation, and good accuracy at matching words that have slightly different phonemic representations

This last point means that IndicSOUNDEX can match related words across slightly different pronunciations, something simply using a phonemic representation or phonetic transcription cannot achieve.

REFERENCES

[1] A. Beider and S. P. Morse. 2010. *Phonetic matching: a better SOUNDEX.* Association of Professional Genealogists Quarterly.
[2] S. Chaware and S. Rao. 2012. *Analysis of phonetic matching approaches for Indic languages.* International Journal of Advanced Research in Computer and Communication Engineering.
[3] A. Liu. 2000. *[ICU4J-discussion] Inter-Indic transliterators.* ICU. https://sourceforge.net/p/icu/mailman/message/11239480/.
[4] A. Parlikar. 2011. *corpus_indian_names.* https://bitbucket.org/happyalu/corpus_indian_names/src/master/.
[5] R. Shah and D. K. Singh. 2014. *Improvement of SOUNDEX algorithm for Indian languages based on phonetic matching.* International Journal of Computer Science, Engineering and Applications.