An Automatic Approximate Matching Technique Based on Phonetic Encoding for Odia Query

Rakesh Chandra Balabantaray, Bibhuprasad Sahoo, Sanjaya Kumar Lenka, Deepak Kumar Sahoo, Monalisa Swain

CLIA Lab, IIIT Bhubaneswar, Gothapatna, PO: Malipada, Bhubaneswar-751003, India.

Abstract

In search engine query optimization plays the major role in order to give relevant result. The user query mostly contains name entities. Not only names but so many words are frequently used as search criteria for information retrieval and identity matching systems in Odia. The names have normally several variations. This variations and errors in names make the exact string matching problematic. If all the variations are approximately matched, then the result can be more relevant. In this paper we put forward an automatic approximate matching technique by which all the variations having similar phonetic code of the query word can be searched and gives the best result. Our algorithm is based on the phonemic encoding of the given query words which can give more relevant result of the desired search.

Keywords: Soundex, Query Optimization, Information extraction, search Engine, Index.

1. Introduction

A web search engine is designed to search for information on the World Wide Web and FTP servers. The search results are generally presented in a list of results often referred as search engine results pages. When a user enters a query into a search engine (typically by using keywords), the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes few parts of the text. Most search engines support the use of the Boolean operators AND, OR and NOT to further specify the search query. Boolean operators are for literal searches that allow the user to refine and extend the terms of the search. The engine looks for the words or phrases exactly as entered. Most search engines employ methods to rank the results to provide the "best" results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another. Search Result / best-matching web pages as well as fast information retrieval depend on nature of query (Optimized Query) and correctly spelled words (soundEx).

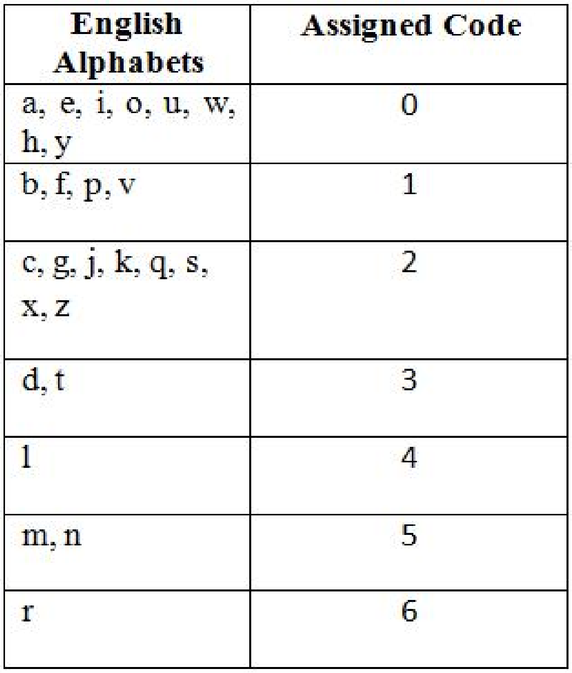
Names play a key role in information systems. Generally when a user gives a query to a search engine, most of the query consists of name entities. Not only names but so many words are frequently used as search criteria for information retrieval and identity matching systems in Odia. There are several variations of nominal form when we type a name in the query. These variations and error in names make the exact string matching problematic. In order to get the best result of the searching query, the query should be legitimate. In this paper we introduce an automatic approximate matching technique based on phonemic features of phoneme of Odia language.

2. Related Works

Many algorithms have been developed to measure the similarities between words both written and spoken; however, most of those researches are based mainly on Soundex for phonetic matching and the Levenshtein distance also called edit distance for string matching techniques. Soundex, the best-known algorithm was developed by Russell and O'Dell (1918) as an early effort to assign a common phonetic code to similar sounding words in the Latin alphabets [8] [10]. This algorithm converts each name to a four-character code that is based on the six places of articulation (plosive, fricative, affricate, glide, liquid, and nasal). It retains the first letter of the name, and drops all the other vowels (i.e. a, e, i, o, u, w, h, y) in the word. If an output of Soundex code is less than four characters, it adds zeros to complete the length. If an output is longer than four, it discards in coding. Soundex phonetic codes for English are given in Table 1.

Soundex algorithm for English is not multilingual. It is language dependent especially based on English pronunciation. The assigning codes and phonemic categories of other languages cannot be directly fitted into original Soundex algorithm developed for English. Therefore, Soundex is adopted by other languages according to their specific phonemic characteristics.

Table-1. Soundex phonetic codes for English



English Alphabets	Assigned Code
a, e, i, o, u, w, h, 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

On the other hand, Levenshtein distance is primarily an algorithm used to investigate a channel model considering the problem of constructing optimal codes capable of correcting deletions, insertions, and reversals [2]. The distance calculates the least number of edit operations that are necessary to modify one string to obtain another string. The cost is normally set to one unit for each of the operations. However, the Levenshtein algorithm does only edit operations between two strings and it does not directly provide knowledge base to identify phonetic similarity among the languages that appeared in different phonemes. But many researches have been conducted by assigning different cost for operations to integrate the knowledge base concept to Levenshtein algorithm [3].

Several methods are used in practice to complete the linkage process. In all nominal data studies, methods must exist to overcome the problems of name variations. Several researchers [1, 4, 5, 6] have proposed composite and hybrid (based on alternative types of variation such a spelling or phonetics) methods to overcome name variation and most hybrid methods are language specific with highly evolved software for parsing and linking the names, times, and spatial variables used in matching[14]. Most of the research works have done using the following algorithms.

- Guth algorithm. This type name is based on the approach due to Guth [7]. The method is left to right sequence driven, and is essentially alphabetic but is independent of language and ethnic issues. It is straightforward to code, is portable, and gives reliable results. It is, however, weak when comparing short names.
- Levenshtein algorithm. These are strictly alphabetic techniques based on edit distance metrics first fully described by Levenshtein [3]. Edit distance is defined for strings of arbitrary length and counts differences between strings in terms of the number of character insertions and deletions needed to convert one into the other, the minimum edit distance is then the similarity.
- Soundex algorithm. The method implemented here is due to Odel and Russell [8]. Soundex is a commonly used technique and has been modified for languages other than English [6].
- Metaphone algorithm. This type name is taken from Binstock and Rex [9] although many variants exist. The method implemented assumes English phonetics but works equally well for forenames and surnames.
- NYSIIS algorithm is an alphabetic algorithms which is easy to implement and which yields canonical index code similar to Soundex. However, NYSIIS differs from Soundex in that it retains information about the position of vowels in the encoded word by converting all vowels to the letter A. The NYSIIS method returns a purely alphabetic code. NYSIIS has been modified and used successfully for an extensive series of record linkage studies and also in the pre-processing step of a generalised, iterative, record linkage system [11].
- Phonex algorithm is a combination of the two methods; Soundex and Metaphone. The method was proved to give a good overall performance when applied to names in the English language [10].
- ISG algorithm. These are hybrid techniques combining alphabetic and phonetic approaches. The similarity comparison is based on the Guth method. The method implemented is due to Bouchard and Pouyez [1]. Bouchard [5] explains that the approach seeks to overcome phonetic variations between names.
- LIG algorithms (e.g. LIG1, LIG2, and LIG3) are hybrid algorithms which combine phonetic and spelling based approaches using similarity measure as probability which described by Snae [11]. The algorithms are a combination of three name matching methods: Levenshtein, Index of Similarity Group (called ISG), and Guth. The LIG algorithms have the best performance in term of producing most accurate true matches, overcoming name variations and increasing the hit rate. They have proved to be more accurate than other methods in the literature which provide phonetic tuning to address multi-cultural names without depending on the language [11].

3. The Features of Odia Language

Odia (Before it is called Oriya) as a mother tongue of more than 32 million people is the official language of Odisha (Officially changed from Orissa to Odisha from November 2011) which is located in eastern part of India and tenth largest language as scheduled in the constitution part- VIII of India. It is also second or third language of many speakers living also in Odisha and in other states of India. Odia people are also in some of the prominent countries of the World e.g. USA, UK, Canada, UAE, Srilanka, Singapore, Malaysia, Burma, Indonesia. There are 45 million Odia speaking people living globally. So we are motivated towards Odia language and working for this language.

Genetically, this language belongs to Indo-Aryan language family which is a branch of Indo-Eranian language which is also a branch of Indo- European family of languages. The proto language from which Odia language is developed is Purbamagadhi Apabhramas. The immediate close languages of Odia are Bengali and Assamese.

This language has many variations regarding the place and language competency of native speakers. On the basis of place, the language has four dialect variations. Odia language in coastal area is called standard Odia. Sambalpuri Odia is spoken in the district of Sambalpur, Phulbani, Kalahandi and Balangari. Dakhinachaliya Odia is spoken in south part of Odisha. Utaranchaliya Odia is spoken in districts of Balasore and Bhadrak. Puri boli as it is a variation of Odia is spoken in district of Puri and in Konark, Chilika and in Puri temple.

The Odia language appears in between ninth to tenth century from Purbamagadhi Apabhrams which is a branch of languages of Indo-Aryan. The Indo-Aryan family of languages in India goes through three stages of development. These are described below.

Old Indo –Aryan (1500BC to 600 BC)
Middle Indo-Aryan (600BC to 1000 AD)
New Indo- Aryan (after 1000 AD)

Sanskrit in Veda provides specimens of Indo-Aryan speech of old period. The earliest document of the linguistic history of Indo-Aryan is the Rig-Veda.

Pali the inscriptional and the literary Prakrits and Apabhrams are specimens of the middle Indo-Aryan period. This compromises of three successive stages of development

1. The earliest stage dated 600B.C to 200 A.D. the language was called Pali during this period. It was used in the literature of budhism.
2. The second stage of it is dated from A.D 200 to A.D 600; standard literary Prakrit represents the stage of development. This language found mainly in Dharm and religious writings of the Jains.
3. The third stage of it is dated from AD 600 to AD 1000. This stage is called Apabhrams. Chryapad is believed to be text of this language. This first literary evidence of Odia is Charyapada. But before Charyapada which is written in tenth century the evidence of Odia language is found in various types of inscription written in seventh century. For instance, Odia word kumbhara (potter) is found in copper plate inscriptions written by Madhaba Barma of khordha, a district in south-eastern part of Orissa. In A.D.991, a copper plate inscription called manjusaa was found which was written by Annanta Barma, where bhitaru, (from within) and pandara (fifteen) Odia words are found. In 1051 a stone inscription called urjam is written in Odia language. [13]

From above evidences it is assumed that, Odia language is originated from ninth century and its evidence is found in various copper and stone inscriptions. In the tenth century this language developed considerably. The literary form of tenth century Odia is seen in charyapadas[13]. In fourteenth century Odia language is completely developed and it is furnished in Mahakabi Saarala Das' Sarala mahabhaarata.

4. Our Approach

The data which are analyzed in this study collected from Odia language and the hypothesis which formulates coding system is based on phonemic pattern of Odia language. All the segmental phonemes are grouped according to their segmental features .The grouping of phonemes is given in the Table-2.

Here, the vowel sounds including two diphthongs and semi vowels are coded '0'. Semivowels are included in vowel group because they cannot stand independently. All the plosive sounds are coded as '1' plosive means, it is a manner of articulation of complete close and suddenly open. There are 18 sounds of plosive in Odia.

Affricate sounds are coded as (2); these sounds are uttered in a manner of complete closer and gradually open . Fricative sounds are coded as (3) it is a manner of creating narrow space of utterance of sound. The trill sound is coded as (4) but ର (ra) and ଋ (ru) can be uttered as trill sound. So in Odia orthography it is two different scripts but it is a single phoneme. Flap sound is coded as 5 and lateral sound is coded as 6 in this study. All the nasal sounds are coded as 7 in this program.

Table-2. Soundex phonemic codes for Odia

Manner of articulation	Odia Phonetic Units	Assigned Code
Vowel and semivowels (ସ୍ବର ଏବଂ ଅର୍ଦ୍ଧସ୍ବର)	ୟ ,ଅ, ଆ, ଇ, ଈ, ଉ, ଊ, ଏ, ଐ, ଓ, ଔ	0
Plosive (ସ୍ପର୍ଶଧ୍ୱନୀ)	କ, ଖ, ଗ, ଘ, ପ, ଫ, ବ, ଭ, ଟ, ଠ, ଡ, ଢ, ଡ଼, ଢ଼, ତ, ଥ, ଦ, ଧ	1
Affricate (ସ୍ପର୍ଶସଂଘର୍ଷି)	ଚ,ଛ,ଜ,ଝ,ଯ	2
Fricative (ସଂଘର୍ଷି)	ସ, ଷ, ଶ, ହ	3
Trill (ତାଡ଼ିତ)	ର, ର, ର	4
Flapped	ଳ	5
Lateral (ଲାଳିତ)	ଲ	6
Nasal (ନାସିକ)	ଙ ,ଞ, ଣ, ନ, ମ ,ଂ, ଁ	7

Odia has other four marginal phonemes these are anuswar, vilar nasal sound, chandrabindu, vowel length.

Chandrabindu is a symbol of nasal sound it can occur with any vowel phoneme. So, it is coded as 7. Similarly, vilar nasal sound is also coded as 7.

There are some exceptional case of these marginal phoneme where coding may not provide positive answer. One important case is when nasal chandrabindu become phoneme it may violate the rules of coding system. But the words based on this case are very few. For example:

ଗା (gaa/sing) , ଗାଁ(gan/village)

After assigning the code to the Odia character, the different variation of an Odia word can be coded into same phonetic code by using the assigned code. Here we have taken the maximum length of the final code as 4. The first character remains same as it is, and the rests are coded using the above code. After that all the zeros are removed and the two conjugative same codes reduced to one like 33 is reduced to 3. After that if the length of the code is less than 4, in order to maintain the length required number of trailing zeros are added to the code. For example:

The variations and the corresponding soundex code for “Bhubaneswar”:

ଭୁବନେଶ୍ୱର --- > ଭ173
ଭୂବନେଶ୍ୱର --- > ଭ173
ଭୁବନେସ୍ୱର --- > ଭ173
ଭୋବନେଶ୍ୱର --- > ଭ173

The variations and the corresponding soundex code for “Sanjay”:

ସଞ୍ଜୟ --- > ସ720
ସଂଜୟ --- > ସ720
ସଙ୍ଜୟ --- > ସ720

The variations and the corresponding soundex code for “Engineering”:

ଇଞ୍ଜିନିଅରିଙ୍ଗ --- > ଇ727
ଇଞ୍ଜିନିଅରିଙ୍ଗ୍ --- > ଇ727
ଇଞ୍ଜିନିୟରିଂ --- > ଇ727
ଇଞ୍ଜିନିୟରିଙ୍ଗ --- > ଇ727

When a user gives a query, the search engine search for all the variations of the given query word or name having the same phonetic code. By this type of approximate matching technique the search engine can give more relevant result.

5. Result and Conclusion

Generally user commits mistake during typing name entities as query in the search engine. But the search engine should be able to give result for all variations of the misspelled name entities for user satisfaction. In this paper, we have proposed an approximate matching technique through which all the variation of the user query word basically name entities can be searched and give the best result. Our approach is basically based on phonemic encoding.

We have collected all the variations of several words from different website and tested our system with these words. We have got an accuracy of 92% by testing on 500 words each with 4 to 5 variations.

6. Acknowledgment

We are very much thankful to The Department of Information Technology (DIT), Ministry of Communication and Information & Technology (MCIT), Govt. of India for supporting this research work.

7. References

[1] G. Bouchard and C. Pouyez, "Name Variations and Computerised Record Linkage," Historical Methods, vol. 13, no. 2, 1980, pp. 119-125.

[2] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals", Journal of Soviet Physics Doklady, vol. 10, no. 8, 1966, pp.707–709.

[3] Prof. dr Kees Mandemakers, "Linking system for historical family reconstruction", International Institute of Social History, pp.5, 2007.

[4] I.P. Fellegi and A. B. Sumter, "A Theory for Record Linkage," Journal of the American Statistical Association, vol. 64, 1969, pp. 1183-1210.

[5] G. Bouchard, "The processing of ambiguous links in computerised family reconstruction," Historical Methods, vol. 19, no. 1, 1986, pp. 9-19.

[6] D. De Brou and M. Olsen, "The Guth Algorithm and the Nominal Record Linkage of ulti-Ethnic Populations," Historical Methods, vol. 19, no. 1, 1986, pp. 20-24.

[7] G. J. A. Guth, "Surname Spellings and Computerised Record Linkage," Historical ethods. Newsletter, vol. 10, no. 1, 1976, pp. 10-19.

[8] K. M. Odell and R. C. Russell, Soundex phonetic comparison system [cf. U.S. Patents 1261167 (1918), 1435663 (1922)].

[9] A. Binstock and J. Rex, Practical Algorithms for programmers. Addison-Wesley, Reading, Mass., pp. 158-160, 1995.

[10] A. J. Lait and B. Randell, "An Assessment of Name Matching Algorithm," Society of Indexers Genealogical Group, Newsletter Contents, SIGGNL issues 17, 1998.

[11] C. Snae and B. M. Diaz, "An Interface for Mining Genealogical Nominal Data Using the Concept of linkage and a Hybrid Name Matching Algorithm," Journal of 3D-Forum Society, vol. 16, no. 1, 2002, pp. 142-147.

[12] Peter Christen, "A Comparison of Personal Name Matching: Techniques and Practical Issues" Proceeding ICDMW '06 Proceedings of the Sixth IEEE International Conference on Data Mining – Workshops, 2006.

[13] Dash. G.N. 1997. "History of Oriya Language". Comprehensive History and Culture of Orissa ,1.

[14] Chakkrit Snae, "A comparision and Analysis of Name Matching Algorithm" World Academy of Science and Technology 25 2007, pp. 252-257.

Dr. Rakesh Chandra Balabantaray is currently working as Assistant Professor in the Department of Computer Science & Engineering and Principal Investigator of CLIA-II Project at IIIT, BHUBANESWAR, Odisha, India. He did his Masters in Computer Science in the year 2001 and Ph.D. in Computer Science in the year 2008 from Utkal University, Odisha, India. He was born in the year 1978. He has more than thirty publications in various reputed journals and conferences. His major area of research is Artificial Intelligence, Natural Language Processing & Information Retrieval.

Bibhuprasad Sahoo is a Research Project Fellow in CLIA-II Lab at IIIT- Bhubaneswar, Odisha, India working in the project entitled "Cross Lingual Information Retrieval (CLIA-II)". He has earned the ME degree in Computer Science and Engineering with Specialization in Knowledge Engineering (KE) from Utkal University, Odisha, India in 2011 and MSc in Computer Science from Ravenshaw Autonomous College, Cuttack, Odisha, India in 2005. He was working as a Junior Project Fellow at RC-ILTR, Utkal University in the period 2007-2009. His major area of research is Natural Language Processing & Information retrieval.

Dr. Sanjaya Kumar Lenka working as Project Fellow in CLIA Lab at IIIT-Bhubaneswar gains degrees of MA and PhD in Linguistics from the Department of Linguistics at Banaras Hindu University and publishes and presents more than 10 research papers in international journals and conferences on the subject of Morphosyntax, Syntax, Morphology, Phonology and Computational Linguistics and currently, concentrates on WordNet of Odia and study of Lexical Semantic of Odia Language. He is awarded with gold medal for standing first in MA in Linguistics.(Sanjaya.lenka@gmail.com)

Deepak Kumar Sahoo earned his Master of Technology (MTech) Degree from International Institute of Information Technology, Bhubaneswar (IIIT-BH) in the year 2009. Worked as faculty in Dept. Of computer Science & Engg at Indic Institute of Design & Research a degree engineering collge. Currently working as a Research project fellow at International Institute of Information Technology, Bhubaneswar (IIIT-BH). Current Research interest Natural Language processing and Information Retrieval.

Monalisa Swain earned her Master of Computer Application (MCA) degree from Biju Patnaik University of Technology In the year 2011. Currently, working as a Research Project Assistant at International Institute of Information Technology, Bhubaneswar (IIIT-BH). Current Research interest is Natural Language Processing and Information Retrieval.