Konkani Integer Phonetic Transcription System

Swapnil Fadte[1], *Edna Vaz Fernandes*[2],[3], *Hanumant Redkar*[1], *Teja Kundaikar*[1], *Ramdas Karmali*[1], *Jyoti D. Pawar*[1]

[1]Discipline of Computer Science and Technology, Goa Business School, Goa University, India
[2]Department of Konkani, Govt. College of Arts Science and Commerce Quepem, Goa India
[3]University of Mumbai, India

swapnil.fadte@unigoa.ac.in, edna.vaz22@gmail.com, hanumantredkar@unigoa.ac.in,
teja.kundaikar@unigoa.ac.in, rnk@unigoa.ac.in, jdp@unigoa.ac.in

Abstract

This paper describes an ongoing work on the Phonetic Dictionary for Konkani language. In this work, we have build a resource that would phonetically transcribe Konkani Integers and generate their written form in the Devanagari script. The algorithm developed in this work takes an integer as an input and generates its written form in the Devanagari script, along with its phonetic transcription in the International Phonetic Alphabet (IPA). Our algorithm is a rule-based system in which phonetic transcriptions of numerals are created using rules from the available literature and for some cases we have proposed new forms for the numerals. The algorithm has been made robust enough to automatically give a written form of any Konkani numeral in the Devanagari script, along with its equivalent IPA transcription. This work is the first step towards providing an open-source phonetic dictionary for Konkani language. We have tried to keep the phonetic transcriptions as much as closer to their natural pronunciations. This is done for the purpose of capturing the general tendency of the language. So, for example, while the number ’8’ आठ [aʈ^{h}] is written with an aspirated retroflex consonant ठ [ʈ^{h}], the final consonant [ʈ^{h}] is heard without aspiration in the actual speech. This loss of aspiration at word final position generally happens across all the consonants of the language, in the Konkani varieties spoken in Goa.

Index Terms: Konkani, Konkani Speech data, phonetic dictionary, integers, Devanagari, integer dictionary

1. Introduction

India is a multilingual country having various languages and dialects. Konkani is the official language of the state of Goa (India). It belongs to the Indo-Aryan language family. The constitution of India - the document that lays down the framework which demarcates fundamental political code, structure, procedures, powers, and duties of government institutions and sets out fundamental rights, directive principles, and the duties of citizens, in its Eighth Schedule (”list”) (Articles 344(1) and 351), recognizes 22 Regional languages as official languages [1]. Speakers of Scheduled Languages enjoy some advantages over the speakers of the non-scheduled languages. For example, the members of the parliament are allowed to speak and present their views in their language if it is one of the Scheduled languages. The Seventy-First Amendment to the Constitution on 20thAugust 1992, added Konkani to the list of Scheduled languages. The official script for Konkani is Devanagari. Numerals are usually written in the Devanagari script. However, they are also written as per the Arabic writing system for ease of communication.

2. Motivation

When we read any text written in a certain script, we encounter characters, numerals and punctuation marks. Anyone who understands a certain script will be able to read any language written in that script. However, when it comes to the reading (pronunciation) of numerals, one needs to follow certain rules. For example, if the integer 1234 is to be read as some value of currency, it needs to be read as ”One Thousand Two Hundred Thirty Four” or ”Twelve Hundred Thirty Four”. Similar rules for integers will also be applicable to the pronunciation of Konkani integers. There are a few machine translation systems available for Konkani[2, 3], but they do not provide text transcriptions of Konkani Numerals. Also, testing of Konkani integers in the text representation on Google translator showed a good amount of errors. Following are some representative examples where we can see a combination of text with numerals. Such instances of data in a text corpus could pose a big challenge for any system that aims to transcribe data accurately.

1. "ता. १ फेब्रुवारी २०२२" [ta. ek februvari don həd͡zar baviːs] (’dtd. 1st February 2022’). The above character and numeral combination refers to a specific date and a year. While १ ’one’ in १ फेब्रुवारी ’1st February’ might be easily transcribed as *ek (februvari)* by any system, the numeral 2022 which is a year, has to be read and transcribed as दोन हजार बावीस [don həd͡zar baviːs] ”two thousand twenty-two” and not as दोन शुन्य दोन दोन [don ʃunj^{ə} don don] ”two zero two two”. Thus, it is important that the system correctly identifies the context in which a numeral appears so that it can generate a proper transcription of the same.
2. "सकाळीं 08:00 ते 10:00 वरांमेरेन" [səkaɭĩ aːʈ^{h} tɛ d^{h}a vərãmeren] ’(from) morning 8 to 10 a.m.’ (Lit. morning 8 to 10 hours till). This phrase specifies a certain time of the day. The system needs to acknowledge this context of time and generate a string that reads the numerals as hours (and minutes in some other temporal context).
3. "गोंयचें क्षेत्रफळ 3701 चौखण किलोमिटर आसा.″ [gõjt͡ʃɛ̃ kʃetrə’fəɭ tiːn həd͡zar satʃɛ ek t͡soʊk^{h}əɳ kilomiʈər asa] (’The (total) area of Goa is 3701 sq. kms.’). The numeral in the above sentence specifies the area of the region of Goa. The accepted way of reading the numeral in this sentence is considering the entire string of numbers as one unit, i.e., as तीन हजार सातशे एक [tiːn həd͡zar satʃɛ ek] (Lit. ”Three thousand Seven Hundred One”) and not as individual numbers- तीन सात शुन्य एक [tiːn sat ʃunj^{ə} ek] ’Three Seven Zero One’.
4. "माशेलाचो पिन कोड 403 107." [maʃɛlat͡sɔ pin’koɖ t͡ʃar ʃunj^{ə} tiːn ek ʃunj^{ə} saːt] (’The pin code of Marcela is 403 107’). Postal Index Number (PIN or simply PIN Code) refers to the

six-digit number used by India Post in its postal code system. More commonly, the numbers indicating such a code are read by spelling out the numerals as discrete units. The Pin code in the above example needs to be read and transcribed as चार शुन्य तीन एक शुन्य सात [t͡ʃar ʃunjᵊ tiːn ek ʃunjᵊ saːt] ’Four Zero Three One Zero Seven’.

5. "ताचो फोन नंबर 9850 403 107" [tat͡ʃɔ fon nəmbər n̪əʋ aːt̪ʰ pãːt͡s ʃunjᵊ t͡ʃaːr ʃunjᵊ tin eːk ʃunjᵊ saːt / tat͡ʃɔ fon nəmbər nain et̪ faiʋ d͡ziro foːr d͡ziro tʰri vən d͡ziro sɛvən](’His phone number is 9850 403 107’). Phone numbers can be read differently by different speakers. However, reading the numbers as discrete units would be a good way to spell out the long number string.

From the above examples, it is clear that the numerals in any given sentence do not lend themselves to same type of output in the spoken and consequently in the written form. While the numerals in example 1 and example 3 are read as (a date and) a year and area respectively, the numerals in example 2 do not undergo much change in the way they are read/pronounced (and hence written) except for the need to write them in the Devanagari script. Similarly, while the numerals in example 4 are read as individual items, there could be other ways some speakers might want to read (pronounce) these. The same holds for the numerals mentioned in example 5 which allow different combinations for pronunciations. Given this background, an accurate transcription of the numerals that adheres to the speaking and writing rules of the language is of great relevance for developing a good transcription System.

3. Scope of the work

Through our work, we have made an effort to develop an automatic system for Konkani language that gives the phonetic as well as Devanagari transcription of a given integer. This is the first kind of work that aims to automatically transcribe Konkani numerals appearing in different contexts into the officially recognised Devanagari script along with the pronunciation of the numerals (given in IPA). As of now, our system only handles the transcriptions of numerals in a positional number system.

4. The proposed system

We are presenting here an automatic phonetic transcription system for Konkani Integers. The system takes an integer as an input and generates its representation in word (the written form) along with its phonetic transcription. Figure 1 diagrammatically presents the system developed by us.

5. Methodology

This section describes the implementation of the system and transcription rules that were used in the Algorithm design.

5.1. Implementation details

The implementation consists of two parts. In the first part, the input is processed and standardised. In the second part, the standardised input is transcribed.

5.1.1. Script conversion

This is the first and the simplest component of the system which takes the input either in Devanagari or Roman digits and converts it to Roman numerals. This is achieved by converting the input value to a new string representation by one-to-one mapping of the digits.

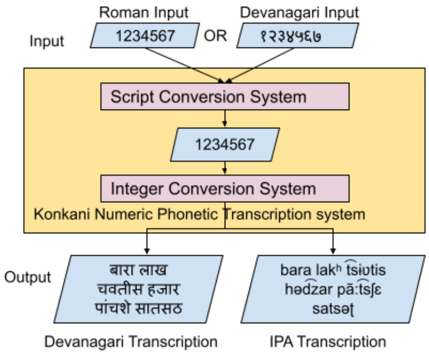


Figure 1: *Integer Transcription System Diagram.*

5.1.2. Integer transcription

This is the main component of the system. It converts the integer into its spoken form (its pronunciation in Konkani) by applying conversion rules and using a dictionary database for the numbers and their corresponding pronunciation. These rules and mappings are presented in a tabular form and explained in detail in section 5.2 below. The algorithm for the transcription is created using these rules. It takes the integer and checks its length. Depending on the length of integer, it then calls for two sub algorithms: Left and Right transcription. Both Left and Right transcription works recursively and completes the transcription and results are merged and returned as final transcription of integers. Although it is rare to get an integer with a large number (length) in practice, ours algorithm can handle an integer of any length.

```
**Data:** $integer$
**Result:** $transcription\_text$
$y \leftarrow$ " ";
$X \leftarrow input$;
$N \leftarrow len(X)$;
**if** $N \geq 12$ **then**
    $RIGHT \leftarrow assign\ last\ 11\ didgits$;
    $LEFT \leftarrow X/10^{11}$ ;    /* removing last 11
      digits from X */
    $y \leftarrow left\_transcription(LEFT) +$
      $+postion\_mapping(12) +$
      $right\_transcription(RIGHT)$
**else**
    $y \leftarrow right\_transcription(X)$;
**end**
```

Algorithm 1: Integer transcription.

5.2. Identification of transcription rules

First, we have identified word transcription for integers. Devanagari transcription provided in Table 1 is from [4]. However, phonetic transcription for integers is not available. This is proposed and provided here for the first time. Rules for transcription for the integers one to hundred are directly mapped and are provided in Table 1. For example, the integer 63 will be mapped

Sr No	Devanagari	Roman	IPA	Devanagari Transcription	Sr No	Devanagari	Roman	IPA	Devanagari Transcription
1	१	1	[eːk]	एक	**51**	५१	51	[ɛkavən]	एकावन
2	२	2	[doːn]	दोन	**52**	५२	52	[bavən]	बावन
3	३	3	[tiːn]	तीन	**53**	५३	53	[treppən]	त्रेप्पन
4	४	4	[t͡ʃaːr]	चार	**54**	५४	54	[t͡ʃoʊpən]	चौपन
5	५	5	[pãːt͡s]	पांच	**55**	५५	55	[pə̃nt͡savən]	पंचावन
6	६	6	[sɨː]	स	**56**	५६	56	[t͡ʃʰappən]	छाप्पन
7	७	7	[saːt]	सात	**57**	५७	57	[səttavən]	सत्तावन
8	८	8	[aːʈ]	आठ	**58**	५८	58	[əʈʈavən]	अड्डावन
9	९	9	[ɳə̃ʋ]	णव	**59**	५९	59	[ekuɳsaʈ]	एकुणसाठ
10	१०	10	[dʰaː]	धा	**60**	६०	60	[saːʈʰ]	साठ
11	११	11	[ikra]	इकरा	**61**	६१	61	[eksəʈ]	एकसठ
12	१२	12	[bara]	बारा	**62**	६२	62	[bãsəʈ]	बांसठ
13	१३	13	[tɛra]	तेरा	**63**	६३	63	[trẽəʈ]	त्रेंसठ
14	१४	14	[t͡ʃɨʋda]	चवदा	**64**	६४	64	[t͡sɨʋsəʈ]	चवसठ
15	१५	15	[pəndra]	पंदरा	**65**	६५	65	[pãsəʈ]	पांसठ
16	१६	16	[sɔɭa]	सोळा	**66**	६६	66	[sãsəʈ]	सांसठ
17	१७	17	[sɨtra]	सतरा	**67**	६७	67	[satsəʈ]	सातसठ
18	१८	18	[ɨʈra]	अठरा	**68**	६८	68	[aʈsəʈ]	आठसठ
19	१९	19	[ekuɳis]	एकुणीस	**69**	६९	69	[ekuɳsɨttɨr]	एकुणसत्तर
20	२०	20	[viːs]	वीस	**70**	७०	70	[sɨttɨ]	सत्तर
21	२१	21	[ekvis]	एकवीस	**71**	७१	71	[ɛkjattər]	एक्यात्तर
22	२२	22	[bavis]	बावीस	**72**	७२	72	[bjattər]	ब्यात्तर
23	२३	23	[tevis]	तेवीस	**73**	७३	73	[trjattər]	त्र्यात्तर
24	२४	24	[t͡ʃovis]	चोवीस	**74**	७४	74	[t͡səʋdjattər]	चवद्यात्तर
25	२५	25	[pə̃ɲt͡ʃvis]	पंचवीस	**75**	७५	75	[pəɲt͡ʃattər]	पंच्यात्तर
26	२६	26	[sɨvvis]	सव्वीस	**76**	७६	76	[ʃattər]	शात्तर
27	२७	27	[səʈavis]	सत्तावीस	**77**	७७	77	[səttjattər]	सत्त्यात्तर
28	२८	28	[əʈʈavis]	अड्डावीस	**78**	७८	78	[əʈʈʰjattər]	अठ्ठ्यात्तर
29	२९	29	[ekuɳis]	एकुणतीस	**79**	७९	79	[ekuɳə̃jʃĩ]	एकुणअंयशीं
30	३०	30	[tiːs]	तीस	**80**	८०	80	[ə̃jʃĩ]	अंयशीं
31	३१	31	[ektis]	एकतीस	**81**	८१	81	[ɛkjãjʃĩ]	एक्यांयशीं
32	३२	32	[bəttis]	बत्तीस	**82**	८२	82	[bjãjʃĩ]	ब्यांयशीं
33	३३	33	[tɛttis]	तेत्तीस	**83**	८३	83	[trãjʃĩ]	त्र्यांयशीं
34	३४	34	[t͡sɨʋtis]	चवतीस	**84**	८४	84	[t͡səʋdjajʃĩ]	चवद्यांयशीं
35	३५	35	[pəstis]	पस्तीस	**85**	८५	85	[pə̃ɲt͡ʃãjʃĩ]	पंच्यांयशीं
36	३६	36	[t͡ʃʰəttis]	छत्तीस	**86**	८६	86	[ʃãjʃĩ]	श्यांयशीं
37	३७	37	[sattis]	सात्तीस	**87**	८७	87	[səttjãjʃĩ]	सत्त्यांयशीं
38	३८	38	[aʈʈis]	आड्डीस	**88**	८८	88	[əʈʈʰjãjʃĩ]	अठ्ठ्यांयशीं
39	३९	39	[ekuɳt͡ʃaɭis]	एकुणचाळीस	**89**	८९	89	[ekuɳɳəʋʋəd]	एकुणणव्वद
40	४०	40	[t͡ʃaɭis]	चाळीस	**90**	९०	90	[ɳəʋʋəd]	णव्वद
41	४१	41	[eket͡ʃaɭis]	एकेचाळीस	**91**	९१	91	[ɛkjaɳɳə̃ʋ]	एक्याण्णव
42	४२	42	[bet͡ʃaɭis]	बेचाळीस	**92**	९२	92	[bjaɳɳə̃ʋ]	ब्याण्णव
43	४३	43	[tret͡ʃaɭis]	त्रेचाळीस	**93**	९३	93	[trjaɳɳə̃ʋ]	त्र्याण्णव
44	४४	44	[t͡səvet͡ʃaɭis]	चवेचाळीस	**94**	९४	94	[t͡səʋdjaɳɳə̃ʋ]	चवद्याण्णव
45	४५	45	[pəɲt͡ʃaɭis]	पंचेचाळीस	**95**	९५	95	[pə̃ɲt͡ʃaɳɳə̃ʋ]	पंच्याण्णव
46	४६	46	[ʃet͡ʃaɭis]	शेचाळीस	**96**	९६	96	[ʃaɳɳə̃ʋ]	शाण्णव
47	४७	47	[səttet͡ʃaɭis]	सत्तेचाळीस	**97**	९७	97	[səttjaɳɳə̃ʋ]	सत्त्याण्णव
48	४८	48	[əʈʈet͡ʃaɭis]	अड्डेचाळीस	**98**	९८	98	[əʈʈʰjaɳɳə̃ʋ]	अठ्ठ्याण्णव
49	४९	49	[ekuɳpənnas]	एकुणपन्नास	**99**	९९	99	[ɳəʋjaɳɳə̃ʋ]	णव्याण्णव
50	५०	50	[pənnas]	पन्नास	**100**	१००	100	[ʃɨmbɨr]	शंबर

Table 1: *Integer mapping rules for integers till 100.*

to [trẽsəʈ, त्रेंसठ]. Table 2 provides the details of positional transcription for hundred, thousand, lakh and crore (ten million) and so on. Table 2 can be consider as a snapshot of the algorithm for various lengths. For example, the integer 1234567 gets positional transcription from Sr. no 4 in Table 2. The algorithm calculates the length and splits the integer into two sub-strings and calls two sub-tasks which recursively calculate the transcription for these two integer strings. There are other interesting cases like, the numeral multiples of 50 usually have multiple pronunciations. for e.g. integer 150 has two common transcriptions and 1500 has four transcriptions. Table 3 shows a few examples for the similar cases. Pronunciation rules for powers of ten till 10^{30} are provided in Table 4. Technically, it can go to infinite with recursive logic for integer transcription.

6. Results and discussion

The output of the phonetic transcription system is shown in figure 2. The algorithm is coded using Python and can be accessed using this link[1]. As regards the phonetic transcription of the integers, we have attempted an accurate transcription of the same. However, one needs to remember that these pronunciations are with regard to our observations about the Konkani spoken in Goa. E.g., as pointed out earlier, the integer written as आठ [aːʈʰ] is mostly pronounced as आट [aːʈ] i.e., without aspiration unlike its written counterpart. The same is true for other integers having aspirated consonants in word final positions or

[1] `https://github.com/SwapnilFadte/Konkani_integer_transcription.git`

word medial positions (as in the case of अठरा, अठ्ठावीस, etc.). Loss of aspiration especially at word-final position is common in Konkani varieties spoken in Goa.

```
Data: $integer$
Result: $transcription\_text$
$y \leftarrow$ " ";
$X \leftarrow input$;
$N \leftarrow len(X)$;
if $N \leq 2$ then
    if $N == 0$ then
        $y \leftarrow$ " ";
    else
        $y \leftarrow integer\_mapping()$ ; /* Using table
         1 transcription rule */
    end
else
    if $N \leq 3$ then
        if $N == 100$ then
            $y \leftarrow integer\_mapping()$
        else
            $y \leftarrow integer\_mapping() +$
             $position\_mappings(3) +$
             $right\_transcription(RIGHT)$
        end
    else
        if $N \leq 5$ then
            $y \leftarrow integer\_mapping() +$
             $position\_mapping(4) +$
             $right\_transcription(RIGHT)$
        else
            if $N \leq 7$ then
                $y \leftarrow integer\_mapping() +$
                 $position\_mapping(6) +$
                 $right\_transcription(RIGHT)$
            else
                if $N \leq 9$ then
                    $y \leftarrow integer\_mapping() +$
                     $position\_mapping(8) +$
                     $right\_transcription(RIGHT)$
                else
                    $y \leftarrow integer\_mapping() +$
                     $position\_mapping(10) +$
                     $right\_transcription(RIGHT)$
                end
            end
        end
    end
end
```

Algorithm 2: right_transcription.

```
Data: $integer$
Result: $transcription\_text$
$y \leftarrow$ " ";
$X \leftarrow input$;
$N \leftarrow len(X)$;
if $N \geq 11$ then
    $RIGHT \leftarrow assign\ last\ 9\ digits$;
    $LEFT \leftarrow X/10^9$ ;        /* removing last 9
     digits from X */
    $y \leftarrow left\_transcription(LEFT) +$
     $position\_mapping(10) +$
     $right\_transcription(RIGHT)$
else
    $y \leftarrow right\_transcription(X)$
end
```

Algorithm 3: left_transcription.

Sr No	**Integer length**	**IPA**	**Devanagari Transcription**
1	3	[ʃɛ]	शे
2	4	[həd͡zar]	हजार
3	6	[lakʰ]	लाख
4	8	[koʈi]	कोटी
5	10	[ərəb]	अरब
6	12	[kʰərəb]	खरब

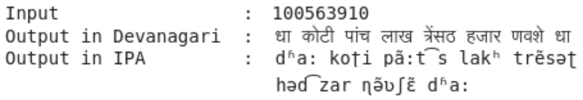
Table 2: *Position mapping rules.*

Sr No	**Integer**	**IPA**	**Devanagari Transcription**
1	150	[ekshɛ pənnas]	एकशे पन्नास
2	150	[deɖʃĩ]	देडशीं
3	250	[donʃɛ pənnas]	दोनशे पन्नास
4	250	[əɖid͡zʃɛ]	अडीजशे
5	350	[tinʃɛ pənnas]	तिनशे पन्नास
6	350	[saɖe tinʃɛ]	साडेतिनशे
7	1150	[ek həd͡zar ekshɛ pənnas]	एक हजार एकशे पन्नास
8	1150	[ek həd͡zar deɖʃĩ]	एक हजार देडशीं
9	1150	[ikraʃɛ pənnas]	इकराशे पन्नास
10	1150	[saɖe'ikraʃɛ]	साडेइकराशे

Table 3: *Multiple transcription rules for numerals above 100.*

Sr No	**Integer**	**IPA**	**Devanagari Transcription**
1	10^{1}	[dʰaː]	धा
2	10^{2}	[ʃɛ/ʃɨmbɨr]	शे/शंबर
3	10^{3}	[eːk həd͡zar]	एक हजार
4	10^{4}	[dʰaː həd͡zar]	धा हजार
5	10^{5}	[lakʰ]	एक लाख
6	10^{6}	[dʰaː lakʰ]	धा लाख
7	10^{7}	[eːk koʈi]	एक कोटी
8	10^{8}	[dʰaː koʈi]	धा कोटी
9	10^{9}	[eːk ərəb/ʃɨmbɨr koʈi]	एक अरब/शंबर कोटी
10	10^{10}	[dʰaː ərəb]	धा अरब
11	10^{11}	[eːk kʰərəb]	एक खरब
12	10^{12}	[dʰaː kʰərəb/lakʰ koʈi]	धा खरब/एक लाख कोटी
13	10^{13}	[ʃɨmbɨr kʰərəb]	शंबर खरब
14	10^{14}	[eːk həd͡zar kʰərəb]	एक हजार खरब
15	10^{15}	[dʰaː həd͡zar kʰərəb]	धा हजार खरब
16	10^{20}	[eːk ərəb kʰərəb]	एक अरब खरब
17	10^{25}	[eːk lakʰ ərəb kʰərəb]	एक लाख अरब खरब
18	10^{30}	[dʰaː ərəb ərəb kʰərəb]	धा अरब अरब खरब

Table 4: *Transcription rules for powers of ten.*



```
Input                  :  100563910
Output in Devanagari   :  धा कोटी पांच लाख त्रेंसठ हजार णवशे धा
Output in IPA          :  dʱa: koʈi pã:t͡s lakʰ trẽsəʈ
                          həd͡zar ɳə̃ʋʃɛ̃ dʱa:
```

Figure 2: *Integer Conversion.*

7. Conclusion and future work

In this work, we have presented a system that transcribes Konkani integers into the officially recognised Devanagari script along with the IPA transcriptions of the numerals. The current system can technically handle integers of infinite length. As a future work it can be extended for fractions, dates, scientific numbers, phone numbers, etc.

8. References

[1] *The Constitution of India*, Eighth schedule, Article(s): 344(1) and 351, Description: Official languages, 1950.

[2] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016.

[3] M. Junczys-Dowmunt, "Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation," in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 225–233. [Online]. Available: https://aclanthology.org/W19-5321

[4] G. K. Academy, *Konkani Shuddhalekhanache Nem 5th Edition*. Secretary,Goa Konkani Akademi 243, Patto Colony, Panaji , Goa - 403 001: Goa Konkani Academy, 1972.