# A New Approach to Syllabification of Words in Gujarati

Harsh Trivedi(✉), Aanal Patel, and Prasenjit Majumder

Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India
{harshjtrivedi94,ptl.aanal,prasenjit.majumder}@gmail.com
http://daiict.ac.in

**Abstract.** This paper presents a statistical approach for automatic syllabification of words in Gujarati. Gujarati is a resource poor language and hardly any work for its syllabification has been reported, to the best our knowledge. Specifically, lack of enough training data makes this task difficult to perform. A training corpus of 14 thousand Gujarati words is built and a new approach to syllabification in Gujarati is tested on it. The maximum word and syllable level accuracies achieved are 91.89 % and 98.02 % respectively.

**Keyword:** Syllabification Gujarati CRF

## 1 Introduction

This paper explains in detail a supervised system developed to split words written in Gujarati into its constituent syllables. This process of word syllabification has important applications in real world text processing. It plays an important role in speech synthesis and recognition [6] and is required for effective text-to-speech (TTS) systems. It is useful in calculating readability indices like Flesh Kincaid, Gunning Fox and SMOG, which require to count the number of syllables in the word. Besides, because of the dynamic nature of any language, a dictionary look-up for syllabification can never suffice. Even more, there is hardly any digital resource for word syllabification in Gujarati. And that is why, to address above mentioned problem an automated system to syllabify Gujarati words can be of immense use.

Many efforts for syllabification in various languages have already been made. Generic principles of the syllabification include Maximum Onset Principle [5], Legality Principle [3] and Sonority Principle [11]. A 99 % efficient statistical approach syllabification was proposed by Mayer [8], which involved counting of the syllables in order know about the best split possible. Hammond [4] showed how to use Optimality Theory for effective syllabification. Another approach demonstrated a discriminative approach that uses Support Vector Machine and Hidden Markov Model together for syllabification with 99.9 % and 99.4 % accuracies in English and German respectively [1]. Conditional Random Fields have also


R. Prasath et al. (Eds.): MIKE 2015, LNAI 9468, pp. 631–639, 2015.
DOI: 10.1007/978-3-319-26832-3_59

been used for syllabification [10]. All these approaches have been predominantly demonstrated in English and European languages.

To the best of our knowledge, no text based data-driven approach has been done on Gujarati and hence this paper attempts to make effort in this direction.

The paper is organized as follows. Section 2 describes the process of data collection and talks about CRF approach adapted for Gujarati, which is used for bootstrapping the training data. Section 3 details on our probabilistic approach for syllabification of Gujarati words. Section 4 elaborates on evaluation details. Conclusion and future scopes are described in Sect. 5.

The system input and output examples are shown in Fig. 1 with symbol (“hyphen”) denoting the syllabic break. Transliteration to English is shown for the purpose of readability. They are not part of the input or output of the system.

| Input | Output |
|---|---|
| ખજૂર */khajur/* | ખ-જૂર */kha-jur/* |
| હોળી */holi/* | હો-ળી */ho-li/* |

**Fig. 1.** Input/Output examples

## 2 Data Collection

Six thousand words and their syllabification were collected from “Babu Suthar’s Gujarati dictionary” [12]. For expanding the corpus to more words, an iterative bootstrapping model was used. Two iterations were conducted and suggestions of 4 thousand words in each iteration were generated using CRF based trained model. These suggestions were rectified by a Gujarati linguist and were added back to training data. The new 4 thousand words selected in each iteration were taken as the most frequently occurring words from a Gujarati newspaper corpus [9]. For rectification of faulty suggestions, an online interface was made and was provided to a language expert to make this process faster. As a result, a corpus of about 14 thousand word syllabifications had been generated.

### 2.1 Using Conditional Random Fields to Bootstrap Data

Modeling the problem of automatic syllabification as sequential tagging problem and usage of Conditional Random Fields [7] for the same has been done before. These approaches have been well demonstrated in languages following Roman script [2,10,14]. The method involves learning to predict the sequence of output labels (syllabic break or not prior/post character) by taking as input sequence of characters of word and their respective feature set.

In our implementation, each unicode character in the word is labeled ‘S’ if it marks the beginning of the syllable and ‘F’ otherwise. For example, for the syllabification ‘શુ-ભેચ્-છા’, the tags are: શ(S), ુ(F), ભ(S), ે(F), ચ(F), ્(F), છ(S), ા(F).

The software that we use as an implementation of Conditional Random Fields is CRF++ [13].

Unlike Roman script, Gujarati alphabet can be categorized in vowels, consonants and *matras*. *Matras* sound like vowel, but they do not exist isolated. They represent vowel-like sounds that are preceded by a consonant. As observed, these set of characters play an important role in deciding position of the syllabic breaks and hence are included in feature vector for CRF.
For feature vector for each character, categorization was done as follows:

- **Vowels:** { અ, આ, ઇ, ઈ, ઉ, ઊ, એ, ઍ, ઐ, ઓ, ઑ, ઔ, અં }
- **Consonants:** { ક, ખ, ગ, ઘ, ચ, છ, જ, ઝ, ટ, ઠ, ડ, ઢ, ત, થ, દ, ધ, ન, પ, ફ, બ, ભ, મ, ય, ર, લ, વ, શ, સ, ષ, હ, ળ, ક્ષ, જ્ઞ }
- **Matras:** { ા, િ, ી, ુ, ૂ, ે, ૅ, ૈ, ો, ૉ, ૌ, ં, ્, ૃ, ૄ, ઁ, ઃ , ઼, ૨ }

**Table 1.** Combinations of context features tried

| Context window | Character n-grams | Total context features |
|---|---|---|
| -1 to 1 | 1 to 3 | 6 |
| -2 to 2 | 1 to 5 | 15 |
| -3 to 3 | 1 to 7 | 28 |
| -4 to 4 | 1 to 9 | 45 |
| -5 to 5 | 1 to 11 | 66 |

For example, for the context window -1 to 1 : { w[-1], w[0], w[1], w[-1]/w[0], w[0]/w[1], w[-1]/w[0]/w[1] } are taken as context features. Context window of -3 to +3 with character 1-gram to 7-grams used as context features turn out to give best results. No significant gain was achieved there-after by increasing context features.

## 3 Our Approach

The Subsect. 3.1 describes a probabilistic model for syllabification, while the Subsect. 3.2 describes an add-on method that when applied prior to the former method, contributes to improve the overall performance.

### 3.1 Predicting Maximum Probable Syllabification

This approach focuses on statistically predicting the most probable syllabification from all the possible syllabifications of a word. It attempts to calculate the probability of each of the possible syllabification being correct and chooses the one with maximum probability. A word with n characters, can theoretically have $2^{n-1}$ possibilities of syllabification owing to n-1 positions where a '-' can be

kept. However, increasing word size increases search space exponentially. Hence, an assumption is taken considering the language constructs to reduce the search space.

It was observed[1] that the Gujarati speakers always pronounce the *matra* along with the preceding consonant or vowel. Hence, an **assumption** was made, that a vowel or a consonant along with all its subsequently occurring *matra's* would always fall into the same syllable, eliminating the possibility of break between them. Using this assumption, a word can be broken down into **units**, each unit being unbreakable any further. For example, શુભેચ્છા /*shubhechcha*/, the units of this words are:

- શુ (શ+ુ) /*shu*/
- ભે (ભ+ે) /*bhe*/
- ચ્ (ચ+્) /*ch*/
- છા (છ+ા) /*chha*/

and the possible syllabifications ($2^{4-1}$) are:

- શુ-ભે-ચ્-છા
- શુભે-ચ્-છા
- શુ-ભેચ્-છા
- શુભેચ્-છા
- શુ-ભે-ચ્છા
- શુભે-ચ્છા
- શુ-ભેચ્છા
- શુભેચ્છા

**Computing Scores:** Let word(W) be composed of n units, $W = u_1.u_2...u_n$ and let $S_i$ be the candidate syllabification to calculate the score for. $S_i$ would be composed of same units as word(W) but with '-' (syllabic break) present between 2 consecutive units. Hence for test syllabification $S_i$, we can define a set:

$$Pairs(S_i) = \{X_i | X_i = u_i u_{i+1} \ or \ u_i \text{-} u_{i+1}\} \tag{1}$$

where,

$u_i u_{i+1}$ indicates absence of syllabic break between units $u_i$ and $u_{i+1}$

$u_i$ - $u_{i+1}$ indicates presence of syllabic break between units $u_i$ and $u_{i+1}$

and using it, probability of $S_i$ being the correct syllabification is approximated as the product of probabilities of having a break or no break at each possible place:

$$P(S_i) \approx \prod_{X_i \in Pairs(S_i)} P(X_i) \tag{2}$$

where,

$$P(X_i = \text{“}ab\text{”}) = \begin{cases} \frac{N_s(\text{“}ab\text{”})}{N_w(\text{“}ab\text{”})} & \text{if } N_w(\text{“}ab\text{”}) \neq 0 \\ 0.5 & \text{otherwise} \end{cases} \tag{3}$$

$$P(X_i = \text{“a-b”}) = \begin{cases} \frac{N_s(\text{“a-b”})}{N_w(\text{“}ab\text{”})} & \text{if } N_w(\text{“}ab\text{”}) \neq 0 \\ 0.5 & \text{otherwise} \end{cases} \tag{4}$$

[1] Assumption verified and corrected by a Gujarati linguist.

$N_s(X)$ = Number of syllabifications containing expression X from training corpus
$N_w(X)$ = Number of words containing expression X from training corpus
For example,

$$\text{P(શુભે-ચ્છા)} \approx \text{P(શુભે)} \times \text{P(ભે-ચ્)} \times \text{P(ચ્છા)}$$

Finally out of all the possible syllabifications which does not contain any illegal syllable[2], one with the maximum score is chosen.

### 3.2 Predicting First and Last Syllable

Some character sequences when occur at beginning or ending of a word are always spoken separately from the word and hence form first or last syllables of that word respectively. The idea behind this can be understood more in English context. For example, 'non', 'un', 'ex' are common prefixes[3] and 'ism', 'ist', 'less', 'ness' are common suffixes in English which when occur at beginning / ending of a word, always form first and last syllables of word respectively. A similar pattern is followed in Gujarati also. 'શ્રી', 'બિન' and 'લીન', 'રણ' are the examples of common prefixes and suffixes respectively in Gujarati.

Instead of feeding system with static prefixes and suffixes, an attempt was made to make supervised model that learns about the important prefix and suffix from the given training data, which when appear at beginning / ending of the word, must be the first and last syllable of the word respectively. The remaining portion can be syllabified using the previous approach to find maximum probable syllabification. To define how important a prefix or suffix is, 2 entities are defined.

**Prefix Score ($S_p$):** It is the probability of the sequence of letters being first syllable, given the word is starting with that particular sequence. For example for sequence $(u_1u_2..u_k)$:

$$S_p(u_1u_2...u_k) = \begin{cases} \frac{N_{fs}(u_1u_2...u_k)}{N_{ws}(u_1u_2...u_k)} & \text{if } N_{ws}(u_1u_2...u_k) \neq 0 \\ \frac{1}{k+1} & \text{otherwise} \end{cases} \quad (5)$$

$N_{fs}(x)$ is number of words with first syllable 'x' in training data
$N_{ws}(x)$ is number of words that start with expression 'x' in training data

**Suffix Score ($S_s$):** It is the probability of the sequence of letters being last syllable, given the word is ending with that particular sequence. For example for sequence $(u_1u_2..u_k)$:

[2] Illegal syllables are the character sequences which do not occur as a syllable in the training data.

[3] Any reference to prefix and suffix in this paper henceforth would refer to first and last syllable of the word respectively.

$$S_p(u_1u_2...u_k) = \begin{cases} \frac{N_{ls}(u_1u_2...u_k)}{N_{we}(u_1u_2...u_k)} & \text{if } N_{we}(u_1u_2...u_k) \neq 0 \\ \frac{1}{k+1} & \text{otherwise} \end{cases} \quad (6)$$

$N_{ls}(x)$ is number of words with last syllable 'x' in training data
$N_{we}(x)$ is number of words that end with expression 'x' in training data

First and last syllables of all words from training data were extracted and prefix and suffix score for them were precomputed respectively. For a sequence of characters if the prefix/suffix score is above a specified threshold, then it is called **confident prefix/suffix** under that threshold respectively.

**Application of Precomputed Scores on the Word:** For a given word it is checked whether character sequence of any length starting from first character serve as a confident-prefix or not. Similarly, an attempt is also made to find the confident suffix. When confident prefix/suffix is found in the word, it is taken as first or last syllable respectively and rest of the word is syllabified with previous approach. If multiple confident prefix / suffix are found, one with maximum score should be selected and if none are found, nothing is to be done. Also, if confident prefix marks first split such that the split occurs between consonant and *matra*, then such confident prefix is not chosen because it would contradict our original assumption. The same is followed for confident suffix. Figure 2 shows process of finding confident prefix in word 'વિ ન ઉ પ યો ગી'.

| **Prefix Score** | **score > threshold ?** |
|---|---|
| P(વિ) | No |
| **P(વિ ન)** | **Yes** |
| P(વિ ન ઉ) | No |
| P(વિ ન ઉ પ) | No |
| P(વિ ન ઉ પ યો) | No |
| P(વિ ન ઉ પ યો ગી) | No |

**Fig. 2.** Example of finding confident prefix(es) in a word

## 4 Evaluation

To define syllabic break, each character would be tagged as 'S' if it marks beginning of the syllable and 'F' otherwise. For example, for syllabification ('abc-d-efg'), sequential tags would be a(S), b(F), c(F), d(S), e(S), f(F), s(F). Evaluation of results has been done in 2 ways. Percentage of words syllabified entirely correctly **(W)**, and percentage of the sequential tags **(S)** detected correctly.

For each of the experiments, 10 fold cross-validation has been done using corpus of about 14 thousand words in Gujarati.

**Table 2.** Evaluation examples

| Actual Tags | Predicted Tags | Compare | Word Accuracy (W) | Syllabic Accuracy (S) |
|---|---|---|---|---|
| ab-cd-ef (SFSFSF) | abc-d-ef (SFFSSF) | (SFSFSF) (SFFSSF) (✓✓✗✗✓✓) | 0/1 | 4/6 |
| ab-cd-ef (SFSFSF) | ab-cd-ef (SFSFSF) | (SFSFSF) (SFSFSF) (✓✓✓✓✓✓) | 1/1 | 6/6 |

### 4.1 Results and Error Analysis

Once the training data was built, 10 fold cross validation on 14 thousand words was done using CRF based approach and the approach[4] mentioned in Sect. 3. For prefix-suffix approach maximum result was obtained at threshold (th = 0.95)

**Table 3.** Results

| | CRF based appraoch | Maximum Probable Approach | Maximum Probable + Prefix/Suffix Approach (th = 0.95) |
|---|---|---|---|
| Word Accuracy **(W)** | 89.56 % | 88.98 % | 91.89 % |
| Syllabic Accuracy **(S)** | 97.58 % | 97.36 % | 98.02 % |

A sample of 10 thousand words was taken to analyse prefix-suffix approach. When operated at threshold 0.95, in total of 1963 words confident prefix was detected out of which 1877 (95.6 %) were correct. Similarly, in total of 8602 words confident suffix was detected out of which 8043 (93.5 %) were correct. These values show the accuracies of prefix-suffix algorithm to detect first and last syllables correctly.

On 10 thousand sampled words, when prefix-suffix approach was applied as an add-on to maximum probable approach, some words turned from wrongly tagged to correctly tagged and vice-versa. Table 4 summarizes the results.

**Credibility of Prefix-Suffix approach:** To support the claim of prefix-suffix model improving the overall performance, a paired T-test was done. 15 groups, each of 20 random sample words were taken and the number of words tagged

[4] Maximum Probable approach refers to method described in Subsect. 3.1 and Prefix/Suffix approach refers to add-on method described in Subsect. 3.2.

**Table 4.** Change of correctness of predicted syllabification on application of Prefix-Suffix approach on Maximum probable approach (10 thousand words)

| | Prefix detected | Suffix detected |
|---|---|---|
| wrong to right | 86 | 342 |
| right to wrong | 8 | 7 |

correctly prior and later to the application of prefix-suffix model were noted. When this data was passed to one-tailed paired T-test, T-value of 1.709 and P-value of 0.054 were obtained. This shows with almost 95 % confidence that that this improvement is not by a mere chance or randomness.

**Exceptions:** The proposed system will fail to work when the training data is not enough to disambiguate between situations of keeping or not keeping a syllabic break at some position. This can be due to lack of enough training data or certain exceptions intrinsic to the language itself. For certain words, the assumption made in Sect. 3 does not hold. For example, syllabifications 'મોર-િ-યો', 'મોર-િ-યો', 'ફ-ટક-ી-ચુ', 'હસ-િયો' include a syllabic break between consonant and following *matra* which is very unnatural in training data. Such words invariably fail to get syllabified correctly. However, the existence of such words are negligible and does not affect the overall performance adversely.

## 5 Conclusion and Future Scopes

Gujarati syllabification data has been bootstrapped from a smaller data-set using CRF model. The resulting data was verified and corrected by a linguist. This demonstrates the use of CRF for Gujarati syllabification. A new approach for syllabification is then tested on this data and compared with the CRF results. The proposed model works quite good at word and syllable level accuracies 91.89 % and 98.02 % respectively. These results are very much comparable with CRF results and hence is offered as its alternative approach for Gujarati syllabification which works on simple statistical calculations.

The assumption underlying this approach to syllabification is followed roughly by 99.34 % of 14 thousand words which shows its soundness. This assumption can also be extended to other Indian languages like Hindi, Bengali, Marathi etc. An active work for building of Hindi and Bengali corpus of syllabified words is going on and as a future work the same procedures can be tested and compared on these languages.

**Acknowledgements.** We are thankful to Dr. Nilotpala Gandhi (HOD in Department of Linguistics, Gujarat University) for her invaluable help in bootstrapping the training data by scrutinizing and rectifying the mistakes predicted by our model. We also acknowledge the occasional but really helpful discussions with Parth Mehta. Finally, the project could not have been possible without the contribution and initial thrust provided by Shubham Patel.

## References


1. Bartlett, S., Kondrak, G., Cherry, C.: On the syllabification of phonemes. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 308–316. Association for Computational Linguistics (2009)
2. Dinu, L.P., Niculae, V., Sulea, O.-M.: Romanian syllabication using machine learning. In: Habernal, I., Matoušek, V. (eds.) TSD 2013. LNCS, vol. 8082, pp. 450–456. Springer, Heidelberg (2013)
3. Goslin, J., Frauenfelder, U.H.: A comparison of theoretical and human syllabification. Lang. Speech **44**(4), 409–436 (2001)
4. Hammond, M.: Parsing syllables: Modeling ot computationally. arXiv preprint cmp-lg/9710004 (1997)
5. Kahn, D.: Syllable-based generalizations in English phonology, vol. 156. Indiana University Linguistics Club Bloomington (1976)
6. Kiraz, G.A., Möbius, B.: Multilingual syllabification using weighted finite-state transducers. In: The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis (1998)
7. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
8. Mayer, T.: Toward a totally unsupervised, language-independent method for the syllabification of written texts. In: Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, pp. 63–71. Association for Computational Linguistics (2010)
9. Palchowdhury, S., Majumder, P., Pal, D., Bandyopadhyay, A., Mitra, M.: Overview of FIRE 2011. In: Majumder, P., Mitra, M., Bhattacharyya, P., Subramaniam, L.V., Contractor, D., Rosso, P. (eds.) FIRE 2010 and 2011. LNCS, vol. 7536, pp. 1–12. Springer, Heidelberg (2013)
10. Rogova, K., Demuynck, K., Van Compernolle, D.: Automatic syllabification using segmental conditional random fields. Comput. Linguist. Neth. J. **3**, 34–48 (2013). http://www.clinjournal.org/node/37
11. Selkirk, E.O.: On the major class features and syllable theory (1984)
12. Suthar, B.: Gujarati- english learner's dictionary, 10 August 2015. http://ccat.sas.upenn.edu/plc/gujarati/guj-engdictionary.pdf
13. Kudo, T.: Crf++: Yet another crf toolkit. https://taku910.github.io/crfpp/
14. Trogkanis, N., Elkan, C.: Conditional random fields for word hyphenation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 366–374. ACL 2010, Association for Computational Linguistics, Stroudsburg, PA, USA (2010). http://dl.acm.org/citation.cfm?id=1858681.1858719