

## A Diachronic Approach for Schwa Deletion in Indo Aryan Languages

Monojit CHOUDHURY, Anupam BASU and Sudeshna SARKAR

Dept. of Computer Science & Engineering,  
Indian Institute of Technology, Kharagpur  
INDIA, PIN-721302

{ monojit, anupam, sudeshna } @cse.iitkgp.ernet.in

### Abstract

*Schwa deletion* is an important issue in grapheme-to-phoneme conversion for Indo-Aryan languages (IAL). In this paper, we describe a syllable minimization based algorithm for dealing with this that outperforms the existing methods in terms of efficiency and accuracy. The algorithm is motivated by the fact that deletion of schwa is a diachronic and sociolinguistic phenomenon that facilitates faster communication through syllable economy. The contribution of the paper is not just a better algorithm for schwa deletion; rather we describe here a *constrained optimization* based framework that can partly model the evolution of languages, and hence, can be used for solving many problems in computational linguistics that call for diachronic explanations.

### 1 Introduction

Linguists propose new models for languages in order to explain language acquisition and processing by humans. Irregularities and exceptions to the theories are often explained by evidence from diachronic linguistics and other social and external phenomena. Absence of diachronic analysis in computational modelling of languages results in a large number of exceptions, which are commonly handled by *ad hoc* rules or exhaustive enumeration. These techniques lead to poor scalability and lack of graceful degradation of the systems along with increased complexity. Although complete modelling of the evolution of language is impossible due to the involvement of myriads of socio-political and cultural factors, it is definitely possible to model certain basic principles of *language change*.

In this paper we describe an algorithm for schwa deletion in Indo-Aryan Languages (IAL) that is motivated by the diachronic evolution of the languages. The proposed computational framework models languages as a constrained optimization

system, where a language evolves by optimizing the rate of communication, subjected to a set of constraints such as *ease of articulation* and *learning*, and *acoustic distinctiveness*. A syllable minimization based optimization function fitted to the aforementioned model has been used for solving the problem of schwa deletion with considerable success.

The paper is organized as follows: Section 2 defines the problem and discusses some of the previous works. Section 3 describes the current models of language evolution, which has been used to develop a computational framework described in the next section. Section 5 and 6 presents the algorithm and its experimental analysis respectively. Section 7 concludes the paper summarizing our contributions.

### 2 The Problem

*Schwa* is defined as the mid-central vowel that occurs in unstressed syllables. The first vowel of the IAL alphabet {**a**}<sup>1</sup> is the schwa. Normally, it is pronounced as /ə/ in Hindi and Sanskrit, and as /ɔ/ in Bengali. Schwa deletion is a phonological phenomenon where schwa is absent in the pronunciation of a particular word, although ideally it should have been pronounced (Ohala, 1983).

Sanskrit and some of the modern IAL that have evolved from it (e.g. Hindi and Bengali), are written from left to right using Brahmi-derived scripts. All the vowels are explicitly represented using diacritical or non-diacritical marks around the consonant except for the schwa, which is the inherent vowel. Unlike Sanskrit, many modern IAL like Hindi and Bengali allow deletion of schwa in certain contexts. **Table I** illustrates this phenomenon for the three languages. In order to determine the proper pronunciation of the words, it is necessary to predict which schwas are deleted and which are not. Thus, schwa deletion is an

---

<sup>1</sup> The graphemes for Indo-Aryan languages are written within '{' and '}' according to the scheme adopted by the International Congress of Orientalists at Athens in 1992. The phonetic transcriptions are written within two '/' using the IPA symbols.

important issue for *grapheme-to-phoneme conversion* of IAL, which in turn is required for a good Text-to-Speech synthesizer (Narasimhan *et al*, 2001).

The Spelling	Pronunciation		
	Sanskrit	Hindi	Bengali
sāphaly a (success)	saϕəly ə (3)	saϕə ljə (3)	ʃaϕəl lo (3)
racanā (creation)	rəcəna (3)	rəcn a (2)	rəcəna (3)
veda (Veda)	vedə (2)	ved (1)	bed (1)

**Table 1.** Pronunciation of three different words in three different IAL. The number of syllables is denoted within parenthesis below the pronunciations. In Bengali {a} can also be pronounced as /o/ in certain contexts.

Several theories have been proposed on the linguistic aspects of schwa deletion in Hindi (Pray, 1970; Kaira 1976; Ohala, 1977, 1983) and its diachronic evolution (Misra, 1967). Ohala (1983) has summarized the rule for schwa deletion in Hindi as

$$ə \rightarrow \phi / VC \_ CV$$

**Condition 1:** There may be no morpheme boundary in the environment to the left.

**Condition 2:** The output of the rule should not violate the phonotactic constraints of Hindi

**Convention:** The rule applies from right to left

The explanation of the rule was based on psycholinguistic evidence; diachronic facts were used only to explain the exceptions. Narsimhan *et al* (2001) designed an algorithm for schwa deletion in Hindi based on this work. The reported accuracy of the algorithm is 89%. Some rules for word final schwa deletion in Bengali have been proposed by Chatterji (1926), but we do not know of any work on computational modelling.

### 3 Factors governing language change

The fact that schwa deletion in IAL is a diachronic phenomenon has been substantiated by Misra (1967). According to Ohala (1983) the deletion of schwas is more frequent in casual and fast speech compared to formal and slower ones. It can be inferred from these facts that the motivation behind

schwa deletion is *faster communication* through minimization of syllables (Tranel 1999).

Some recent works on mathematical and simulation based modelling of language evolution (Boer, 2000; Cangelosi and Parisi, 2002; Nowak *et al*, 2002) suggests that several features of languages emerge due to some basic cognitive and articulatory factors. These models assume a) *ease of articulation*, b) *ease of learning*, and c) *acoustic distinctiveness* as the primary driving forces behind language evolution. The three forces operate simultaneously over the language in order to maximize the rate of successful communication in terms of time and effort spent by the language users to generate, understand and learn the language. Thus, language can be modelled as a *multi-objective optimization* system, where the optimization criteria are

- Minimization of effort (in terms of energy and time spent while conveying a piece of information)
- Minimization of learning time and effort
- Minimization of probability of misunderstanding (in the sense of confusing one word with another)

These three criteria are mutually contradictory and therefore there exists no global optimum. Let us examine the phenomenon of schwa deletion under this multi-objective optimization model for language evolution. When a vowel is deleted from a word the number of syllables reduces by one. For example, in Table 1, for the second word, Sanskrit and Bengali have three syllables, whereas due to the deletion of a schwa, the Hindi pronunciation has only two syllables. Reduction of syllables implies shorter time for pronunciation of a word, and hence faster communication. However, deletion of schwas in certain contexts might result in a consonant cluster which the native speakers find very difficult or impossible to pronounce. This beats the very purpose of schwa deletion, i.e. the minimization of effort of articulation and therefore, is unacceptable. The second condition for the rule proposed by Ohala (section 2) refers to this constraint.

There are contexts where deletion of schwa would not give rise to inadmissible consonant clusters. For example, in the Hindi/Bengali word **pari** (fairy, /pəri/ in Hindi), if the first schwa is deleted, the pronunciation would be /pri/, which does not violate the phonotactic constraints of the

languages. The schwa, however, is not deleted, because /pəri/ and /pri/ are too distinct from each other to be interpreted as the same word. Moreover, /pri/ is closer to other Hindi words like **priya** (favorite, /prijə/). In this case, the deletion of schwa reduces the *acoustic distinctiveness* of the word from other words in the lexicon, which increases the probability of misunderstanding, and hence the schwa might not be deleted in such a context.

#### 4 Computational framework

We propose the following diachronic explanation for schwa deletion in IAL.

In old IAL none of the schwas are deleted. The modern IAL use the script and spelling conventions similar to Sanskrit. Due to a higher evolutionary pressure on the spoken forms of the languages than on the written forms, schwas are deleted in the pronunciation, but are still present in the graphemic forms. The deletion is a slow diachronic phenomenon, where in order to communicate faster, initially the speakers unintentionally deleted the schwas. Only those deletions were acceptable that did not lead to a syllable structure which was too difficult to pronounce, learn or understand for the native speakers. Gradually, the pattern of deletion spread across the population and over the different items in the lexicon.

In this section, we describe a computational framework for modelling the aforementioned hypothesis based on the three optimization criteria stated in the last section. The aim of the proposed framework is not to validate the hypothesis through *micro-simulation* (Cangelosi and Parisi, 2002); rather it tries to predict the schwa deletion pattern based on the optimizations that might have affected the deletion of schwas diachronically. In the next section, we present an efficient algorithm for schwa deletion in IAL, which can be automatically constructed from this model, without the help of any other evidence.

##### 4.1 Basic definitions

All the unexplained symbols used below stand for their usual meaning in the context of formal language theory. Please refer to (Hopcroft and Ullman, 1979) for details.

$\Sigma_g$  ( $\Sigma_p$ ): A finite set of the graphemes<sup>2</sup> (phonemes) in the language  
 $\Sigma_g = V_g \cup C_g, \quad \Sigma_p = V_p \cup C_p$

<sup>2</sup> Graphemes here do not refer to glyphs. Free vowels and their corresponding diacritical marks are considered to be the same symbol

Where

$V_g$  ( $V_p$ ): Finite set of graphemes (phonemes), which are vowels

$C_g$  ( $C_p$ ): Finite set of graphemes (phonemes), which are consonants. Semivowels are also considered as consonants.

$\alpha \in V_g$  is a special symbol that represents schwa.

We define,

$$f_{g2p}: \Sigma_g \rightarrow \Sigma_p$$

$f_{g2p}$  is the default mapping of the graphemes to the phonemes. This oversimplification is made here for two reasons. First, since IAL use a phonetic script, this in general is true<sup>3</sup> and second, this assumption does not have any affect on the schwa deletion algorithm.

A *word*  $w$  is defined as a 2-tuple  $\langle w_g, w_p \rangle$ , where

$$w_g \in \Sigma_g^+ \text{ and } w_p \in \Sigma_p^+$$

A *lexicon*  $\mathcal{A}$  is the union of all the valid words  $w$  of a language. A *grapheme-to-phoneme converter* is defined as a function  $F_{g2p}: \Sigma_g^+ \rightarrow \Sigma_p^+$ , such that

$$\forall w \langle w_g, w_p \rangle \in \mathcal{A}, F_{g2p}(w_g) = w_p$$

##### 4.2 Phonotactic constraints

In order to model the *ease of articulation*, we start with the modelling of *phonotactic constraints*. A *consonant cluster* is a string of the form  $C_p C_p^+$ . Phonotactic constraints restrict the presence of some of the consonant clusters in the phonetic representation of a *word* ( $w_p$ ). At the most generic level we can think of a *consonant cluster ranking* (*CCR*) function, where  $\mathbb{N}$  is the set of natural numbers

$$CCR: C_p^+ \rightarrow \mathbb{N}$$

The function *CCR* is independent of any language and every language has a threshold  $\tau_{CCR}$ , such that a consonant cluster  $x \in C_p^+$  is allowed in the language if and only if

$$CCR(x) \leq \tau_{CCR}$$

We define two special variants of *CCR*,  $O\_CCR$  and  $C\_CCR$ , which ranks the *admissibility* of the consonant clusters at the *onset* and *coda* positions respectively. The definition is similar to that of *CCR*, and  $\tau_{OCCR}$  and  $\tau_{CCCR}$  are the corresponding threshold values.

<sup>3</sup> This assumption is not strictly valid since a cluster of consonant might be mapped to a single consonant or a different cluster.

The *sonority hierarchy* (Vennemann, 1988) and *markedness conditions* (Kager, 1999) along with the physiology of the articulatory mechanism point towards the existence of a language independent ranking function as hypothesized above. However, there might be accidental gaps in the list of admissible consonant clusters of a language (Ohala, 1983), which can not be explained on the basis of *CCR* alone. Therefore, we define a Boolean function *ADM* that tell us about the admissibility of consonant clusters in a language.

$$\begin{aligned} \text{ADM}: C_p^+ &\rightarrow \{0, 1\}, \text{ such that for } s \in C_p^+ \\ (\text{ADM}(s) = 1) &\Leftrightarrow (s \text{ is an admissible cluster}) \end{aligned}$$

In general, we can derive this function from *CCR* as

$$\text{ADM}(s) = \text{sign}(\tau_{CCR} - \text{CCR}(s))$$

However, we might have to forcefully convert some values to 0 due to accidental gaps.

### 4.3 Syllable and Syllabification

We define a *syllable*  $\sigma$  as a regular expression, with the assumption that the *nucleus* contains a single vowel. Thus,

$$\sigma \in C_p^* V_p C_p^*$$

The *syllabification* function *SYL* maps the phonetic representation  $w_p$  of a word  $w$  to a string of syllables  $\sigma_1\sigma_2\dots\sigma_m$  such that the *effort of articulation* and *learning* are minimum.

We model the *effort of articulation* using a *syllable ranking* function *SR*, which is similar to *CCR*.

$$\text{SR}: C_p^* V_p C_p^* \rightarrow \mathbb{N}$$

*SR* is mainly dependent on the *structure* of the syllable. We enumerate the first few terms of the function *SR*.

$$\begin{aligned} \text{SR}(V_p) &= 1, & \text{SR}(C_p V_p) &= 2 \\ \text{SR}(C_p V_p C_p) &= 3, & \text{SR}(V_p C_p) &= 4 \\ \text{SR}(C_p C_p V_p) &= 5, & \text{SR}(C_p C_p V_p C_p) &= 6 \\ \text{SR}(C_p C_p C_p V_p) &= 7, & \text{SR}(C_p V_p C_p C_p) &= 8 \end{aligned}$$

For all other possible syllable structures  $\sigma'$ ,  $\text{SR}(\sigma') > 8$

Also, for any syllable  $\sigma$ ,

$$\begin{aligned} [\text{O\_CCR}(\text{onset}(\sigma)) > \tau_{occr}] &\vee \\ [\text{C\_CCR}(\text{coda}(\sigma)) > \tau_{cccr}] &\Rightarrow (\text{SR}(\sigma) = \infty) \end{aligned}$$

This means that if either the coda or the onset of a syllable is inadmissible, then the ranking function maps the syllable to the highest possible rank, represented symbolically by the infinity ( $\infty$ ). *onset* and *coda* are projection functions that project the longest valid prefix and suffix of a syllable respectively that are elements of  $C_p^*$ .

We define a *syllabification* to be *valid* if all the syllables are *valid* (i.e. strings of the form  $C_p^* V_p C_p^*$ ) and every symbol in the word is a part of one and only one *syllable* in the syllabification. We can define a partial ordering,  $\leq_\sigma$ , among the possible *valid syllabifications* of a given word based on *SR*, such that the syllabification with smaller number of high ranked syllables is preferred to one that has more hard (high ranked) syllables. Now we define *SYL* ( $w_p$ ) as the set of all possible syllabifications  $\sigma_1\sigma_2\dots\sigma_m$  such that (i)  $\sigma_1\sigma_2\dots\sigma_m$  is a *valid syllabification* of  $w_p$  and (ii) there exist no other *valid syllabification*  $\nu$  of  $w_p$  such that  $\nu \leq_\sigma \sigma_1\sigma_2\dots\sigma_m$ .

The definitions of syllable and syllabification are motivated by the *markedness conditions* (Kager, 1999) and experimental results on child language acquisition (MacNeilage and Davis, 2000), that show that some syllables and syllabifications are easier to learn and pronounce than others.

### 4.4 Acoustic distinctiveness constraints

Perceptual experiments show that speakers always articulate the onset of the syllables more clearly and correctly compared to the articulations of the vowel and the coda (Fosler-Lussier *et al*, 1999; Greenberg, 1999). Therefore, it is likely that the hearer distinguish between syllables by paying more weight to the onset than to the coda. A continuous distance metric  $D_\sigma$  might be defined based on these experimental results, such that the *probability of confusion* (interpreting one syllable as another) between two syllables  $\sigma$  and  $\sigma'$  increases as the value of  $D_\sigma(\sigma, \sigma')$  decreases. We can further define an *acoustic distance function*  $D_w$ , using the function  $D_\sigma$ , which measures the probability of confusion between two arbitrary words in the phonetic domain.

In the case of schwa deletion, however, we want the *acoustic distance* between the ideal pronunciation (without any schwa deletion) and the normal pronunciation (with schwa deletion) to be smaller, so that the word is not confused with other words in the lexicon. Formally, for the graphemic representation of a word  $w_g = x_1 x_2 \dots x_n$ ,

$D_w(f_{g2p}(x_1).f_{g2p}(x_2)...f_{g2p}(x_n), F_{g2p}(w_g)) < \tau_{critical}$ , where  $\tau_{critical}$  is the maximum allowable distance and ‘.’ is the concatenation operator. Rather than modelling this as an optimization criterion, we reformulate this as a constraint. The simplification in this case serves our purpose.

We define, where  $x \in C_p$

$$D_\sigma(x.f_{g2p}(\alpha), \phi) = 0 \quad (4a)$$

$$D_\sigma(\sigma, \sigma.x) = 0 \quad (4b)$$

For all other cases  $D_\sigma$  is *infinity* ( $\infty$ ), unless the two syllables are identical. (4c)

(4a) allows the deletion of a schwa from an open syllable; (4b) allows the concatenation of a consonant at the coda position. This is motivated by the fact that coda has least distinctiveness (Greenberg, 1999). (4c) restricts any change at the onset of a syllable or the vowels other than schwa.

On the basis of  $D_\sigma$  we can define  $D_w(w_{p1}, w_{p2}) = 0$  if and only if there exists an alignment between the sequences  $SYL(w_{p1})$  and  $SYL(w_{p2})$ , with possible *gaps* ( $\phi$  or null syllables) such that for all the corresponding pairs of syllable taken from the two sequences, the *acoustic distinctiveness* ( $D_\sigma$ ) between them is 0. Thus, only operations allowed are deletion of a schwa and addition of a consonant at the coda position. Anything else is forbidden for the sake of *acoustic distinctiveness*.

We conclude this section by summarizing below the salient features of the model by comparing it with the optimization criteria stated in section 3.

- The functions **SR**, **CCR** and its variants that rank the phonotactic constraints is a measure of the effort of articulation, learning and the probability of misunderstanding. Therefore we want to minimize it. However, it has been modelled as a constraint (**ADM**).
- The function **SYL** is so defined that the efforts of articulation and learning are minimized.
- $D_w$  models the *acoustic distinctiveness* i.e. the criterion **3c**, but it has been reformulated as a constraint as well.

## 5 The algorithm

We want to define  $F_{g2p}$  for a language given **ADM** and  $D_w$ .  $F_{g2p}$  should be such that it enables faster communication by minimization of syllables by deletion of schwa.

## 5.1 Formal definition

Let  $w_g$  be an input sequence of graphemes to the function  $F_{g2p}$ . Let  $w_p \in \Sigma_p^*$  be obtained by replacing all graphemes  $x$  in  $w_g$  by  $f_{g2p}(x)$ . Let  $w_p'$  be obtained by deletion of some (possibly all or none) of the schwas ( $f_{g2p}(\alpha)$ ) in  $w_p$ .  $F_{g2p}(w_g) = w_p'$ , if and only if  $D_w(w_p, w_p') = 0$  and  $(\forall v_p)[(v_p \text{ can be obtained by deleting schwas from } w_p) \wedge (D_w(w_p, v_p) = 0) \Rightarrow |SYL_g(w_p')| \leq |SYL_g(v_p)|]$

In words it means that among all  $w_p'$  obtainable by deletion of some of the schwas from  $w_p$ , that respects both the **ADM** (*phonotactic*) and  $D_w$  (*acoustic distinctiveness*) constraints, the one with the minimum number of syllables is chosen as the output of  $F_{g2p}$ .

```

procedure SYL :
input:  $w_p$ , O_CCR, C_CCR
output:  $\sigma_1\sigma_2... \sigma_m$  //The syllabification

1. Include up to the first vowel in  $w_p$  in  $\sigma_1$ 
2. If there are 2 consonants  $c_1c_2$  between the
current vowel and the next vowel, include  $c_1$  in the
current syllable and  $c_2$  in the next syllable.
3. If there are 3 consonants  $c_1c_2c_3$  between the
current vowel and the next vowel,
3.1 if  $O\_CCR_p(c_2c_3) \leq \tau_{OCCR}$ , include  $c_1$  in the
current syllable and  $c_2c_3$  in the next syllable
3.2 else if  $C\_CCR_p(c_1c_2) \leq \tau_{CCCR}$  include  $c_1c_2$  in the
current syllable and  $c_3$  in the next syllable
3.3 else NO syllabification is possible
4. If there is one or no consonant between the
current vowel and the next vowel, terminate the
current syllable and begin the next syllable
5. Continue from step 2 till there are symbols not
included in any syllable.
end procedure

```

**Figure 1.** Algorithm for syllabification

## 5.2 A greedy strategy

**Figure 1** describes a linear time algorithm for syllabification (**SYL**) that conforms to the definition provided in section 4.3. This uses the fact that the maximum length of allowable consonant clusters for IAL is three. After syllabification of  $w_p$ , we try to greedily delete the schwas so that the constraints specified by **4a**, **4b** and **4c** are not violated. **4a** states that only a schwa which is a part of an open syllable ( $c\alpha$ , where  $c \in C_p$ ) can be deleted and **4b** states that after schwa deletion,

the consonant  $c$  is appended to the coda of the previous syllable. Therefore, both of them together imply schwas in two consecutive syllables cannot be deleted. Along with that, the following constraints can also be derived from the  $D_w$

constraints (the reasons are omitted due to space constraints):

- R1. Schwa of the first syllable cannot be deleted
- R2. Schwa cannot be deleted before a consonant cluster.
- R3. The word final schwa can always be deleted unless the appending of the penultimate consonant to the previous syllable results in an inadmissible cluster.
- R4. For Bengali, which does not allow complex codas, schwas cannot be deleted after consonant clusters.
- R5. A schwa followed by a vowel cannot be deleted.

```

procedure  $F_{g2p}$ :
input:  $w_g$ ,  $ADM$ 
output:  $w_p$  //The pronunciation

1.  $w_p' = f_{g2p}(x_1) \cdot f_{g2p}(x_2) \dots f_{g2p}(x_n)$ , where  $w_g$  is
 $\langle x_1 x_2 \dots x_n \rangle$ 
2. Syllabify  $w_p'$  using procedure  $SYL$ 
3. Using rules R1 to R6 and  $ADM$  constraints mark
the schwas which cannot be deleted as  $F$ 
4. While traversing the word from right to left
4.1 Delete a schwa if it is not marked  $F$ 
4.2 Appended the dangling consonant to the
coda of the adjacent syllable (to the left)
4.3 If the adjacent syllable (to the left) has a
schwa which is unmarked, mark it  $F$ 
4.4 Go to 4.1 if there are more schwas to the left
of the current position.
5. At the end of step 4 we get the syllabified string of
phonemes  $\langle x'_1 x'_2 \dots x'_m \rangle$ , which is the required
output
end procedure

```

**Figure 2.** Algorithm for schwa deletion

We have the following rule that cannot be captured by the constraints:

- R6. Schwa following a  $y$  (pronounced as /j/) cannot be deleted if it is preceded by a high vowel because /j/ is a glide from high vowel to a low/medium vowel (schwa), deletion of schwa would make the presence of the glide imperceptible.

This rule could have been captured by the  $D_\sigma$  constraints but we state it here as a separate rule for the sake of simplicity. Figure 2 describes an algorithm for schwa deletion using the rules above.

It is easy to see that the time complexity of the algorithm is  $O(|w_g|)$ . Due to limited space, we omit the proof that the algorithm for  $F_{g2p}$  indeed minimizes the number of syllables without violating the constraints specified by  $ADM$  and  $D_w$ . However, there might be more than one (precisely 2) possible solutions and in that case the algorithm chooses one of the solutions on the basis of the direction of traversal at step 4. The *right to left*

traversal gives better results (as has been confirmed by Ohala, 1983) because the duration of syllables reduces towards the end of the word and hence the tendency to delete schwas at the word final position increases.

## 6 Experimental Results and Discussions

The algorithm was implemented for Bengali and Hindi and tested on a set of words. **Table 2** summarizes the results for Hindi (tested on the words in a pocket dictionary (Hindi-Bangla-English, 2001)). The algorithm for Bengali was tested on 1000 randomly selected words from a corpus and found to be around 85% accurate.

Some of the important features of the algorithm are as follows.

- **Efficiency:** The algorithm runs in linear time on the input word length. It scans the whole word just twice. Thus, the hidden constant is also very small.
- **Polymorphic Words:** The algorithm can handle polymorphic words, if the morphological information about the word is provided. This is because schwa deletion is not carried across morpheme boundaries. Morphological analyzer for Hindi and Bengali were implemented and integrated with the algorithm. For Hindi, the results were nearly perfect (99.89%)

**Exceptions:** For Hindi there was hardly any exception to the algorithm. For Bengali, the types of words that were incorrectly processed by the algorithm include a class of very frequently used, disyllabic modifier adjectives, certain suffixes, borrowed words from Sanskrit and compound words. In Bengali, the schwa which is retained (as opposed to the predictions by the algorithm) are pronounced as /o/ and not as /ɔ/. Since, /o/ is not a central vowel, deletion of /o/ is marked as compared to deletion of /ɔ/ which is unmarked. Transformation of schwa to some non-neutral vowel in Hindi is unknown and therefore, the algorithm works perfectly for Hindi.

Experimental results for Hindi	Test Size (No. of words)	Incorrect results	Accuracy
Without MA	1109 5	431	96.12%
With MA	1109 5	12	99.89%

**Table 2.** Experimental results for Hindi schwa deletion. The results are for individual words. MA stands for Morphological Analysis

## 7 Conclusion

In this paper, we have described the phenomenon of schwa deletion in the IAL and proposed a diachronic explanation for it. In order to model the diachronic evolution, we used the concepts of ease of articulation, ease of learning and acoustic distinctiveness. We developed a computational framework, where we reformulated some of the optimization criteria as constraints and one of them (the syllable minimization) as the basic optimization function. The outcome of this is an efficient and accurate algorithm for solving schwa deletion in IAL.

The contribution of this paper is not just a better algorithm for schwa deletion, which is necessary for developing Text-to-speech synthesizers for IAL, but a new approach based on a *constrained optimization* framework, motivated by the diachronic evolution of languages. A closer look at the algorithm will reveal that it is not much different from the schwa deletion rule proposed by Ohala (1983). However, Ohala's rule was based on psycholinguistic and empirical observations, whereas we have derived the rule from a set of very basic assumptions (minimization of syllables and certain constraints). The algorithm itself can provide an explanation for the phenomenon.

It must be mentioned that neither the aim nor the findings of this work are meant to propose a new model of *language change*. The models and concepts used here were all present previously and we have assumed and included some of them directly in our model. Our finding is not a proof of those models and can be considered only as a further validation. Our only claim here is that diachronic clues can help solve important problems in computational linguistics and for this we provide a computational framework and a specific example.

Some of the questions that we would like to address in the future include modelling of optional schwa deletion in Bengali compound words, evolution of morpho-phonology for Bengali verb systems, and modelling of dialect diversity using diachronic clues. More realistic, yet manageable computational frameworks for holistic or detailed modelling of language evolution can also be an interesting area of future research.

## References

Bart de Boer 2000. Self Organization in Vowel Systems. *Journal of Phonetics*, 28:441-465

Angelo Cangelosi and Domenico Parisi (Eds) 2002. *Simulating the Evolution of Language*. Springer-Verlag, London

Suniti K. Chatterji 1926. *The Origin and Development of the Bengali Language*. Rupa and Co.

Eric Fosler-Lussier, Steven Greenberg and N Morgan 1999. Incorporating contextual phonetics into automatic speech recognition. *Proc. Int. Cong. Phon. Sci.*, San Francisco, pp. 611-614.

Steven Greenberg 1999. Speaking in shorthand - A syllablecentric perspective for understanding pronunciation variation. *Speech Communication*, 29:159-176.

*Hindi Bangla English – Tribhasa Abhidhaan*. 2001 Sandhya Publication

John E. Hopcroft and Jeffery D. Ullman 1979. *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, USA

Rene Kager 1999. *Optimality Theory*. Cambridge University Press

S. Kaira 1976. *Schwa-deletion in Hindi*. Language forum (back volumes), Bhari publications, 2 (1)

Peter F. MacNeilage and Barbara L. Davis 2000. On the Origin of Internal Structure of Word Forms. *Science*, 288:527-31

B. G. Misra 1967. *Historical Phonology of Standard Hindi: Proto Indo European to the present*. Cornell University Ph. D. dissertation

Manjari Ohala 1977. The Treatment of Phonological variation: An example from Hindi. *Lingua*, 42: 161-76

Manjari Ohala. 1983. *Aspects of Hindi Phonology*, volume II. MLBD Series in Linguistics, Motilal Banarsidass, New Delhi.

Bhuvana Narasimhan, Richard Sproat and G Kiraz. 2001. Schwa-deletion in Hindi Text-to-Speech Synthesis. *Workshop on Computational Linguistics in South Asian Languages*, 21st SALA, Konstanz

Martin A. Nowak, Natalia L. Komarova and Partha Niyogi 2002. Computational and Evolutionary Aspects of Language, *Nature*, 417:611-17

B. R. Pray 1970. *Topics in Hindi – Urdu grammar*. Research Monograph 1, Berkeley: Center for South and Southeast Asia Studies, University of California

Bernard Tranel 1999. Optional Schwa Deletion: on syllable economy in French. *Formal Perspectives on Romance Linguistics*, Ed. By J. Mark Authier, Barbar S. Bullock, & Lisa A. Reed.

T. Vennemann 1988. *Preference Laws for Syllable Structures*. Mouton de Gruyter, Berlin