International Journal of Applied Research 2021; 7(10): 134-140

ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 8.4
IJAR 2021; 7(10): 134-140
www.allresearchjournal.com
Received: 03-08-2021
Accepted: 04-09-2021

Alak Chakraborty
Research Scholar for PhD, Department of English, Tripura University, West Tripura, India

Corresponding Author:
Alak Chakraborty
Research Scholar for PhD, Department of English, Tripura University, West Tripura, India

Coda cluster simplification of English loans in Chittagonian Bangla: An optimality theoretic account

Alak Chakraborty

DOI: https://doi.org/10.22271/allresearch.2021.v7.i10b.9028

Abstract
The paper investigates the phenomenon of cluster simplification process at word-final level of English loans to Chittagonian Bangla through vowel epenthesis and consonant deletion. An Optimality eoretic (OT) analysis follows and exemplifies the cluster simplification process. It has been well represented that the emergence of constraint *CLUSTER^{CODA} (Kager 1999) is in the driving seat in Chittagonian Bangla. MAX- C/V and CONTIGUITY are two more vital constraints beside two lower ranked faithful constraints MAX-IO and DENT-IO that rank in the vowel epenthetic and consonant deletion process. Further, ANCHOR-IO (R) works vital as the second lowest ranked constraint before DEP-IO. Deletion of 'r' in coda cluster is very distinctive in CB. Finally, it is apparently found that, the deletion of consonant /r/ results in a merger and thus influencing the lengthening of preceding mono-phthong to keep the syllable weight intact in CB.

Keywords: Chittagonian Bangla, epenthesis, deletion, optimality theory, *Cluster^{coda}

1. Introduction
The aim of this paper is to explore the phenomenon of coda cluster simplification process with special reference to loanword adaptation of English to Chittagonian Bangla (henceforth CB). Like other East Bangla Dialect, CB does not allow tautosyllabic Cluster both at onset and coda. The vowel epenthesis process in CB takes place to break consonant clusters at word-initial, word-medial and word-final level. On the other hand, consonant deletion also occurs at the same three levels. Here in the current study, we will focus on epenthesis and deletion at word-final, syllable-final or coda level cluster. An Optimality Theoretic (OT) account is exemplified to justify the process of vowel epenthesis and consonant deletion, and here in this study our full concentration will be on coda declustering process. The following examples demonstrate the declustering process in word-final cluster in CB.

Table 1: Declustering coda cluster

1.a.	English	CB	Gloss
	/fɔ:(r)m/	/ɸɔrɔm/	'form'
	/gɪlt/	/gilʈi/	'guilt'
b.	/dʒʌmp/	/za:m/	'jump'
	/pɑrk/	/ɸa:k/	'park'
	/lɪft/	/li:p/	'lift'

Loanword adaptation is a universal process. When a word from foreign diction is borrowed to a native language it goes through some phonological changes to conform to the target language. This nativization of foreign loans depends on different phonemic inventories, syllable structure and phonotactic constraints that exist between the foreign language (L_2) and the native one (L_1). In our studied language CB, we see many foreign words from Arabic, Persian, Portuguese, Hindi, Urdu and English, which are adapted. However, this adaptation follows the phonological system of L_1. CB phonology does not welcome tautosyllabic cluster both at onset and coda and this markedness constraint remains unchanged while adapting foreign loans to CB. As a result, these foreign words need to be simplified in the adaptation process. In this process, words with consonant cluster go through some simplification processes namely vowel epenthesis and consonant deletion. Vowel epenthesis happens both at edge and internal position, while consonant deletion happens with the deletion of either C_1 or C_2 depending on their sonority scale.

2. Literature Review

Sonority Sequencing Principle (SSP) is a driving force in cluster simplification process in CB. Sonority is intensity of a certain segment compared to other sound segments of same stress, pitch and length (Ladefoged, 1993) [15]. The sonority of the sound depends on the vocal tract. The more open the vocal tract, the more is the sonority of the sound. As a result, vowels are considered most sonorous. Steriade (1990) [24] proposed a band scale of 0 to 7 to make a hierarchy among the vowel and consonant sounds. He scaled from 0 (most sonorous) to 7 (least sonorous), where, 0 for vowels, 1 for glides, 2 for liquids, 3 for nasals, 4 for voiced fricatives, 5 for voiceless fricatives, 6 for voiced stops and 7 for voiceless stops. This scale of sonority can be shown below:

vowels » glides » liquids » nasals » vd fricatives » vl fricatives » vd stops » vl stops
(most sonorous) ⟶ (least sonorous)

Clement (1990: 291) suggests that the sonority scale is built into phonological theory as part of Universal Grammar. Clements' universal sonority scale only consists of the four major natural classes of sounds ranked from least sonorous to most sonorous which is presented below:
obstruents < nasals < liquids < glides
Selkirk (1984) [23] makes more distinctions of the non-syllabic segments especially the obstruents and liquids and proposes the following universal sonority scale:
p, t, k < b, d, g < f, θ < v, z, ð < s < m, n < l < r
Kenstowicz (1994: 256) [14] emphasizes that SSP has main role in the syllabification. He
observes that SSP ensures onset to rise in sonority towards the nucleus and codas to fall in
sonority from the nucleus. It ensures that in onset sonority moves towards the nucleus, but in coda it turns to least sonorous. But in CB cluster simplification at onset, consonant deletion at onset keeps the least sonorous one intact and deletes the more sonorous consonant, which goes against SSP. On the other hand, coda in CB keeps the most sonorous consonant intact and deletes the least sonorous one and thus it violates SSP constraint, which is exemplified below:
/drʌm/ > /ɖa:m/ ‘drum’, /ʤʌmp/ > /za:m/ ‘jump’

However, this is not fully right in case of ‘r+obstruent’ cluster at coda, as the most sonorous one gets deleted. As for example: /θɜ:(r)d/ > /t̪á:ɖ/ ‘third’. We can name this process ‘r’ deletion. This type of deletion process will be discussed later with sufficient illustration.

3. Epenthesis as Cluster Simplification Process

Epenthesis is an influential type of repair strategy in CB. It refers to insertion of a sound in a word, particularly to a syllable. In this process a new vowel (consonant insertion is very rare, which is discussed later on in brief.) may appear between two consonant clusters as for cluster simplification. In CB phonology, epenthesis is the addition of a vowel at edge or inside the consonant cluster. It occurs when words are borrowed from English that have consonant clusters, which are not permitted in CB. Epenthesis seems to be a preferred strategy to repair clusters in many languages such as Burmese (Chang 2009) [2], Fula (Pardis & LaCharite 1997) [20], Hindi (Singh 1985), Bangla (Karim 2011) [13], Assamese (Dutta 2016) [9], Shoan (Uffmann 2006) and some others. Vowel epenthesis occurs in case of ‘Liquid + Nasal’, ‘Liquid + Obstruent’, ‘Obstruent+Sonorant’, ‘Sonorant+ Obstruent’ and ‘S + Obstruent’ clusters. These are presented in the following:

Table 2: Different Types of Epenthesis at Coda

2. a.	Liquid + Nasal	lm, rm, rn	Anaptyxis
b.	Liquid + Obstruent	lb, rb	Anaptyxis
c.	Obstruent +Sonorant	bl, tl, dl, kl, gl, ðm, tm, tn, dn, ʃn, zn	Anaptyxis
d.	Sonorant + Obstruent	mp, lt	Paragogue
e.	s + Obstruent	st	Paragogue

In the above mentioned chart, ‘Anaptyxis’ is such kind of epenthesis where internal epenthesis is seen, whereas ‘Paragogue’ ensures edge epenthesis. In the coda cluster simplification process vowel epenthesis is a very regular phonological process. The examples of vowel epenthesis are exemplified below: In ‘Liquid + Nasal’ cluster a vowel is inserted inside the cluster to ensure cluster simplification process.

Table 3: ‘Liquid + Nasal’ Cluster Simplification

2. a. **Cluster**	**English**	**CB**	**Gloss**
lm	/fɪlm/	/ɸilim/	‘film’
rm	/fɔ:(r)m/	/ɸɔrɔm/	‘form’
rn	/hɔ:(r)n/	/hɔrɔn/	‘horn’

Similar process of ‘Anaptyxis’ is also seen in the simplification process of ‘Liquid + Obstruent’ cluster which is exemplified below:

Table 4: ‘Liquid + Obstruent’ Cluster Simplification

b. **Cluster**	**English**	**CB**	**Gloss**
lb	/bʌlb/	/bálɔp/	‘bulb’
rb	/vɜ:(r)b/	/bárɔp/	‘verb’

In English phonology segments /m, n, l, r/ can form a syllable of its own without any vowel. For example: m, n, l in English words like rhythm, tension, table function like the nucleus of a syllable. However, in CB phonology these consonants do not function like syllabic consonant. As a result, English words ending with bl, tl, dl, kl, gl, ðm, tm, tn, dn, ʃn, zn etc. need to be simplified while adaptation. It is also seen that the SSP is violated in those clusters where last consonant is more sonorous than the previous one. CB does not allow these syllabic consonants and as a result a vowel epenthesis happens to decluster the consonants. Thus it is seen that, ‘Obstruent+Sonorant’ cluster needs to be simplified because ‘Words can’t end with an obstruent followed by a sonorant’ (Dutta 2016) [9].

Table 5: 'Obstruent+Sonorant' Cluster Simplification

c. **Cluster**	**English**	**CB**	**Gloss**
bl	/ʈeibl/	/ʈebil/	'table'
tl	/bɒtl/	/bɔʈəl/	'bottle'
dl	/mɒdl/	/mɔɖɛl/	'model'
kl	/ləʊkl/	/lokal/	'local'
gl	/ga:gl/	/gargəl/	'gargle'
dm	/ri:ðm/	/riɖəm/	'rhythm'
tm	/sɪstm/	/sisʈɛm/	'system'
tn	/kɒtn/	/xɔʈən/	'cotton'
dn	/sʌdn/	/sadɛn/	'sudden'
ʃn	/tɛnʃn/	/ʈɛnʃən/	'tension'
zn	/kʌzn/	/xazin/	'cousin'

An epenthesis process named 'Paragogue' is seen in 'Sonorant + Obstruent' cluster simplification. In this process a vowel is inserted at edge. This process of 'Paragogue' in case of 'Sonorant + Obstruent' cluster is exemplified below:

Table 6: Sonorant + Obstruent' Cluster Simplification

d. **Cluster**	**English**	**CB**	**Gloss**
mp	/læmp/	/lɛmpə/	'lamp'
lt	/gɪlt/	/gilʈi/	'guilt'

Similar process of 'Paragogue' is also seen in the simplification process of 's + Obstruent', which is exemplified below. Like the earlier one (2.d) this process is also rare.

Table 7:'s + Obstruent' Cluster Simplification

e. **Cluster**	**English**	**CB**	**Gloss**
st	/lɪst/	/liʃʈi/	'list'
	/fɜ:st/	/faʃʈə/	'first'

4. Deletion as Cluster Simplification Process

Deletion is another prominent type of cluster simplification process found in CB. This happens in "Nasal + Obstruent", "[r] + Obstruent", "[l] + Obstruent" and "Obstruent+ Obstruent" clusters. Two types of deletion processes are found which are popularly known as Apocope and Syncope. When consonant at edge gets deleted, it is known as Apocope. On the other hand, when the internal consonant is deleted, it is known as Syncope. So all the deletion process in CB can be categorized below:

3. a.	**Nasal + Obstruent**	**mp, nd, ŋk, nt, ns**	**Apocope**
b.	[r] + Obstruent	rd, rk, rs, rt, rv (del. liq)	Syncope
c.	[l] + Obstruent	lb, ld, lt (del. obs.)	Apocope
d.	Obstruent+ Obstruent	ft, sk	Apocope

In the above mentioned chart, 'Apocope' is such kind of deletion where C_2 is deleted, whereas 'Syncope' ensures the deletion of C_1. In the coda cluster simplification process consonant deletion is a regular phonological process. The examples of consonant deletion are exemplified below:

3. a. Deletion in case of coda 'Nasal + Obstruent' clusters in CB:

Table 8: 'Nasal + Obstruent' Cluster Simplification

Cluster	**English**	**CB**	**Gloss**
mp	/kæmp/	/kɛm/	'camp'
nd	/bænd/	/bɛn/	'banned'
ŋk	/bæŋk/	/bɛŋ/	'bank'
nt	/pænt/	/ɸɛn/	'pant'
ns	/laɪsəns/	/laisɛn/	'license'

b. Deletion in case of coda '[r] + Obstruent' clusters in CB, which is well known as 'r' deletion. This is exemplified below:

Table 9: '[r] + Obstruent' Cluster Simplification

Cluster	**English**	**CB**	**Gloss**
rd	/kɑ(r)d/	/xa:ɖ/	'card'
rk	/pɑɹk/	/ɸa:k/	'park'
rs	/tɔ:(r)ʧ/	/ʈɔ:s/	'torch'
rt	/ʃɜ:(r)t/	/ʃa:ʈ/	'shirt'
rv	/nɜ:(r)v/	/na:b/	'nerve'

c. Deletion in case of coda '[l] + Obstruent' clusters in CB. In this process the obstruent gets deleted and thus it help maintain the SSP.

Table 10: '[l] + Obstruent' Cluster Simplification

Cluster	**English**	**CB**	**Gloss**
lb	/bʌlb/	/bɔ́:l/	'bulb'
ld	/gəʊld/	/go:l/	'gold'
lt	/bɛlt/	/bɛ:l/	'belt'

d. Deletion in case of coda 'Obstruent + Obstruent' clusters in CB. In this process the least sonorous consonant gets deleted.

Table 11: 'Obstruent + Obstruent' Cluster Simplification

Cluster	**English**	**CB**	**Gloss**
ft	/lɪft/	/li:p/	'lift'
sk	/flɑ:sk/	/fɛla:s/	'flask'

Here in 'Obstruent + Obstruent' clusters, CB deletes the less sonorous segment contrary to universal preference for low sonority coda. A logical explanation in this regard is the requirement for contiguity between the segments, which results in the deletion of the consonant at the edge of the syllables and the preservation of the consonant closer to nucleus. It is also noted that the consonant adjacent to the vowel is more salient perceptually than the marginal consonant.

5. Analysis of vowel epenthetic and consonant deletion process in coda cluster

It has been already mentioned earlier that CB does not allow tautosyllabic consonant cluster both at onset and coda level. So for cluster simplification process both epenthesis and deletion occur. Epenthesis seems to be a preferred strategy to repair clusters in many languages such as Burmese (Chang 2009) [2], Japanese (Teuber 2012) Fula (Pardis & LaCharite 1997) [20], Hindi (Singh 1985), Bangla (Karim 2011) [13], Assamese (Dutta 2016) [9], Shoan (Uffmann 2006) and some others. Here, we look only at English loans in CB and when our corpus data is analyzed, we get the following picture that can clarify the epenthesis and deletion process. In our corpus data presented in 3 and 4, it is quite evident that coda consonant cluster is not entertained in CB. Hence, a high ranked markedness constraint *COMPLEX^{CODA} (Kager 1999) [11] runs dominant in CB which is presented below with some other markedness and faithfulness constraints.

Table 12: Dominant Markedness and Faithfulness Constraints in CB

***Complexcoda**	**'Codas are simple'**	**(Kager 1999) [11]**
Anchor-R	'Any segments at the right periphery of the output has a correspondent at the right periphery of the input'	(Kager 1999) [11]
Contiguity	Elements adjacent in the input must be adjacent in the output.' (no medial epenthesis/deletion)	(Gouskova, 2001) [10]
Dep-Io	'Every segments of the output must have input correspondents'('No epenthesis')	(McCarthy & Prince, 1995) [17]
Max-Io	'Every segments of the input must have output correspondents'('No deletion')	McCarthy & Prince, 1995) [17]

However, in our corpus data it is shown that deletion at coda is more pracitsed process in CB than the epenthesis one. Now determining the deletion of C_1 or C_2 will be a vital question. That is why, we need to use one more constraint used by Cote (2004), which is mentioned below:

MAX-C/V 'Do not delete a consonant adjacent to a vowel' (Cote, 2004)

Only the least salient consonants may delete and frequency of deletion correlates with the relative perceptibility of the consonants (Cote 2004: 167). She even continues that postvocalic consonants are benefitted from the cues existent in vocalic transitions, which means the consonant adjacent to the vowel enjoys the priority not to be deleted in cluster simplification process. So in our OT analysis MAX-C/V will be 2nd most vital constraint after *COMPLEX^{CODA} in CB.

6. OT Constraint ranking in coda cluster simplification process

Based on the constraints mentioned above we can opine that the markedness constraint *COMPLEX^{CODA} remains the driving force constraint for declustering coda cluster in CB. Next, over the issue of epenthesis or deletion, we can depend on the *Preservation Principle* of Paradis and LaCharité (1997) [20]. Kang refers them and says that, if the cost of preservation is not too extreme…epenthesis should commonly be preferred over deletion (2010: 13). Thus, MAX-IO, the faithful constraint that disallows deletion wins over DEP-IO, another faithful constraint for cluster simplification process. But, in case of selecting the priority of consonant deletion, we can propose the constraint MAX-C/V next to *COMPLEX^{CODA}.

To ensure 'no medial epenthesis/deletion' we need a faithful constraint CONTIGUITY next. This constraint crucially ranked higher than MAX-IO and DEP-IO in case of consonant deletion process. Thus we can determine two constraint rankings for vowel epenthesis (ranking 1) and consonant deletion (ranking 2) which are mentioned below:

Ranking 1: *COMPLEX^{CODA}, MAX-C/V, MAX-IO>>CONTIGUITY, ANCHOR-R, DEP-IO

Ranking 2: *COMPLEX^{CODA}, MAX-C/V, CONTIGUITY>>MAX-IO, ANCHOR-R, DEP-IO

In 'liquid [r] + obstruent' cluster the liquid [r] gets deleted (as mentioned in 3.b). To validate this [r] deletion we can follow Cote's analysis where she says that, in postvocalic positions /r/ becomes a vocalic off glide which may reduce to nothing (2004). Actually it happens, if /r/ in postvocalic position is succeeded by an obstruent. This [r] deletion is termed by her as a "merger" and this merging of /r/ happens with the preceding vowel. According to her observation, /r/ can be considered a glide in postvocalic position as she referred to Quebec French. If we consider /r/ as a glide, then it does not violate the ranking constraint of MAX-C/V. However, this can be taken as a perception based clarification of loanword adaptation mentioned by Peperkamp and Dupuox in their Perception Approach (2003).

7. a. OT Analysis for Epenthesis

Here, the following Tableau 1 illustrates the epenthesis process in case of 'Liquid + Nasal' clusters in CB, i.e. adapting the word /fɪlm/>/ɸilim/.

Tableau 1: Illustration of the Word /fɪlm/>/ɸilim/

/fɪlm/	***COMPLEX^{CODA}**	**MAX-C/V**	**MAX-IO**	**CONTIGUITY**	**ANCHOR-R**	**DEP-IO**
(a) /ɸilm/	*!					
(b) /ɸim/		*!	*			
(c) ☞ /ɸilim/				*		*
(d) /ɸil/			*!		*	

In Tableau 1, the 'Liquid + Nasal' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (b) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates MAX-IO and ANCHOR-R. Thus, candidate (c) becomes the optimal candidate because of least violation, as it violates CONTIGUITY and DEP-IO, which are low ranked constraints.

Now, the following Tableau 2 illustrates the epenthesis process in case of 'Liquid r + Obstruent' clusters in CB, i.e. adapting the word /vɜ:(r)b/>/bárɔp/.

Tableau 2: Illustration of the Word /vɜ:(r)b/>/bárɔp/

/vɜ:(r)b/	*COMPLEX^{CODA}	MAX-C/V	MAX-IO	CONTIGUITY	ANCHOR-R	DEP-IO
(a) /bá:rb/	*!					
(b) /bá:b/			*!			
(c) ☞ /bárɔp/				*		*
(d) /bá:r/			*!		*	

In Tableau 2, the 'Liquid r + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (b) violates the next highest constraint MAX-IO, which is a fatal violation. Candidate (d) violates MAX-IO and ANCHOR-R. Thus, candidate (c) becomes the winning candidate because of least violation, as it violates CONTIGUITY and DEP-IO, which are low ranked constraints.

Next, the following Tableau 3 illustrates the epenthesis process in case of 'Obstruent+sonorant' clusters in CB, i.e. adapting the word /ʈeibl/>/ʈebil/.

Tableau 3: Illustration of the Word /ʈeibl/>/ʈebil/

/ʈeibl/	*COMPLEX^{CODA}	MAX-C/V	MAX-IO	CONTIGUITY	ANCHOR-R	DEP-IO
(a) /ʈeibl/	*!					
(b) ☞ /ʈebil/				*		*
(c) /ʈeib/			*!		*	
(d) /ʈel/		*!	*			

In Tableau 3, the 'Obstruent+Sonorant' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (c) violates the next highest constraint MAX-IO and ANCHOR-R, which are fatal violations. Candidate (d) violates MAX- C/V. Thus, candidate (b) becomes the winning candidate because of least violation, as it violates CONTIGUITY and DEP-IO, which are low ranked constraints.

Next, the following Tableau 4 illustrates the epenthesis process in case of 'Sonorant + Obstruent' clusters in CB, i.e. adapting the word /gɪlt/>/gilʈi/.

Tableau 4: Illustration of the Word /gɪlt/>/gilʈi/

/gɪlt/	*COMPLEX^{CODA}	MAX-C/V	MAX-IO	CONTIGUITY	ANCHOR-R	DEP-IO
(a) /gilʈ/	*!					
(b) ☞ /gilʈi/					*	*
(c) /giʈ/		*!		*		
(d) /gil/			*!		*	

In Tableau 4, the 'Sonorant + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (c) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates MAX-IO and ANCHOR-R. Thus, candidate (b) becomes the winning candidate because of least violation, as it violates ANCHOR-R and DEP-IO, which are low ranked constraints.

Next, the following Tableau 5 illustrates the epenthesis process in case of 's+ Obstruent' clusters in CB, i.e. adapting the word /lɪst/>/liʃʈi/.

Tableau 5: Illustration of the Word /lɪst/>/liʃʈi/

/lɪst/	*COMPLEX^{CODA}	MAX-C/V	MAX-IO	CONTIGUITY	ANCHOR-R	DEP-IO
(a) /list/	*!					
(b) ☞ /liʃʈi/					*	*
(c) /liʈ/		*!		*		
(d) /liʃ/			*!		*	

In Tableau 5, the 's + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (c) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates MAX-IO and ANCHOR-R. Thus, candidate (b) becomes the winning candidate because of least violation, as it violates ANCHOR-R and DEP-IO, which are low ranked constraints.

7. b. OT Analysis for Deletion

The following Tableau 6 illustrates the deletion process in case of 'Nasal + Obstruent' clusters in CB at coda, i.e. of the adaptation of the word /kæmp/, which changes to /kɛm/.

Tableau 6: Illustration of the Word /kæmp/>/kɛm

/kæmp/	*COMPLEX^{CODA}	MAX-C/V	CONTIGUITY	MAX-IO	ANCHOR-R	DEP-IO
(a) /kæmp/	*!					
(b) /kɛp/		*!	*	*		
(c) ☞ /kɛm/				*	*	
(d) /kɛmɛp/			*!			*

In Tableau 6, the 'Nasal + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA} as CB does not allow tautosyllabic cluster at coda. Candidate (b) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates CONTIGUITY and DEP-IO. Thus candidate (c) becomes the winning candidate because of least violation, as it violates MAX-IO and ANCHOR (R), which are low ranked constraints.

Next, the following Tableau 7 illustrates the deletion process in case of 'liquid [r] + Obstruent' clusters in CB, i.e. adapting the word /pɑɹk/>/ɸa:k/.

Tableau 7: Illustration of the Word /pɑɹk/>/ɸa:k/

/pɑɹk/	***COMPLEX^{CODA}**	**MAX-C/V**	**CONTIGUITY**	**MAX-IO**	**ANCHOR-R**	**DEP-IO**
(a) /ɸark/	*!					
(b) /ɸakɔ/		*!	*			*
(c) ☞ /ɸa:k/				*	*	
(d) /ɸarɔk/			*!			*

In Tableau 7, the 'Nasal + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (b) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates CONTIGUITY and DEP-IO. It seems that candidate (c) violates MAX-C/V, but it is not so, as [r] works as a merger with the succeeding vowel and thus lengthens it. So, candidate (c) becomes the winning candidate because of least violation, as it violates MAX-IO and ANCHOR (R), which are low ranked constraints.

Next, the following Tableau 8 illustrates the deletion process in case of 'liquid [l] + Obstruent' clusters in CB, i.e. adapting the word /bʌlb/>/bɔ́:l/.

Tableau 8: Illustration of the Word /bʌlb/>/bɔ́:l/

/bʌlb/	***COMPLEX^{CODA}**	**MAX-C/V**	**CONTIGUITY**	**MAX-IO**	**ANCHOR-R**	**DEP-IO**
(a) /balb/	*!					
(b) /bɔb/		*!		*		
(c) ☞ /bɔ́:l/				*	*	
(d) /bɔlɔb/			*!			*

In Tableau 8, the 'liquid [l] + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (b) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates CONTIGUITY and DEP-IO. Thus, candidate (c) becomes the winning candidate because of least violation, as it violates MAX-IO and ANCHOR (R), which are low ranked constraints.

Next, the following Tableau 9 illustrates the deletion process in case of 'Obstruent + Obstruent' clusters in CB, i.e. adapting the word /lɪft/>/lip/.

Tableau 9: Illustration of the Word /lɪft/>/lip/s

/lɪft/	***COMPLEX^{CODA}**	**MAX-C/V**	**CONTIGUITY**	**MAX-IO**	**ANCHOR-R**	**DEP-IO**
(a) /lift/	*!					
(b) /lit/		*!		*		
(c) ☞ /lip/				*	*	
(d) /lifit/			*!			*

In the above Tableau, the 'Obstuent + Obstruent' cluster is simplified. Candidate (a) violates the highest ranked constraint *COMPLEX^{CODA}. Candidate (b) violates the next highest constraint MAX-C/V, which is a fatal violation. Candidate (d) violates CONTIGUITY and DEP-IO. Thus, candidate (c) becomes the winning candidate because of least violation, as it violates MAX-IO and ANCHOR (R), which are low ranked constraints.

8. Conclusion

In this paper, I have provided an OT analysis to account for the simplification process of complex coda of English Loans to Chittagonian Bangla. Here, I have shown how English clusters are simplified in CB with the help OT constraint ranking. To validate this cluster simplification I have shown two major processes named the vowel epenthesis and consonant deletion processes, which are explained with proper examples. Vowel epenthesis occurs in case of 'Liquid + Nasal', 'Liquid + Obstruent', 'Obstruent+Sonorant', 'Sonorant+Obstruent' and 'S + Obstruent' clusters. Consonant deletion happens in 'Nasal + Obstruent', '[r] + Obstruent', '[l] + Obstruent' and 'Obstruent+ Obstruent' clusters. The constraint rankings that I have proposed here are a good mixture of both markedness and faithful constraints. The article demonstrates how markedness constraint *COMPLEX^{CODA} acts as the driving force behind the cluster simplification process. It has also shown how MAX-C/V and MAX-IO rank high after *COMPLEX^{CODA} constraint in case of vowel epenthesis. The next higher constraint is CONTIGUITY that ranks higher than ANCHOR-R and DEP-IO in CB in case of vowel epenthesis. Next, in case of consonant deletion we see a great role played by the constraint CONTIGUITY after MAX-C/V. Thus, the OT account of coda consonant cluster simplification process in CB projects that the phonological process is not arbitrary, rather rule-based. Finally, it is apparent that the [r] deletion in 'stop+liquid [r]' clusters in CB coda is a process of deleting [r] in the form of lengthening the preceding vowel. With the [r] deletion process at coda CB phonology ensures the preservation of weight by ensuring Compensatory Lengthening of the preceding vowel. Again, it can be observed as an auditory perception of deleting [r] by the CB speakers like the native speakers of English.

Reference

1. Bat-El. "Selecting the best of the Worst: The Grammar of Hebrew Blends". Phonology 1996;13:283-328.
2. Chang, Charles B. "English Loanword Adaptation in Burmese". Journal of the Southeast Asian Linguistic Society 2009;1:77-94.
3. Clements GN. The role of the sonority cycle in core syllabification. In J. Kingston & M. E. Beckman (Eds.), Papers in Laboratory Phonology I. Between the Grammar and Physics of Speech, Cambridge: Cambridge University Press, 1990, 283-333.
4. Côté, Marie-Hélène. Consonant Cluster Phonotactics: A Perceptual Approach. Doctoral Dissertation, Massachusetts Institute of Technology, 2000.
5. "Consonant cluster simplification in Québec French". Probus, 2004;16(2):151-201.
6. "Syntagmatic distinctness in consonant deletion". Phonology 2004;21(1):1-41.
7. Das, Shyamal. "Some Aspects of the Prosodic Phonology of Tripura Bangla and Tripura Bangla English". Doctoral Dissertation, Central Institute of English and Foreign Languages, Hyderabad, ROA 2001, 493.
8. Essays in Linguistics: Studies in Phonology, Syntax and Sociolinguistics. Akanksha Publishing House, 2009.
9. Dutta, Hemanga. "English Loans in Assamese: A Phonological Account". Paper presented in The IAFOR International Conference on Language Learning, Dubai 2016.
10. Gouskova, Maria. "Falling Sonority Onset, Loanwords, and Syllable Contact". The Main Session by the Chicago Linguistic Society 2001;37:175-185.
11. Kager, René. Optimality Theory. Cambridge University Press, Cambridge 1999.
12. Kang, Yoonjung. Tutorial overview: Suprasegmental adaptation in loanwords". Lingua 2010;120:2295-2310.
13. Karim, Khaled. "An Optimality Theoretic (OT) Account of Word Final Vowel Epenthesis and Deletion Processes in the Incorporation of loanwords into the Dhaka Dialect of Bangla". Working Papers of the Linguistic Circle of the University of Victoria 2011;21: 22-33.
14. Kenstowicz, Michael. Phonology in Generative Grammar. Blackwell Publishers, Cambridge 1994, 250-268.
15. Ladefoged, Peter. A course in phonetics. 3rd edition, Fort Worth, Texas, 1993.
16. McCarthy, John J, Alan Prince. Prosodic Morphology: Constraint Interaction in Generative Grammar and Satisfaction. Ms. University of Massachusetts, Amherst and Rutgers University, New Brunswick, NJ, 1993.
17. "Faithfulness and Reduplicative Identity". In University of Massachusetts Occassional Papers in Linguistics 18: Papers in Optimality Theory, ed. Jill N. Beckham, Stanislaw Urbanczyk and Walsh Dickey. Amherst, Massachusetts: GLSA, 1995, 249-384.
18. McCarthy, John J. Optimality Theory in Phonology. Blackwell Publishing Ltd, 2004.
19. Doing Optimality Theory: Applying Theory to Data, Blackwell, 2008.
20. Pardis, LaCharite. "Preservation and Minimality in Loanword Adaptation". Journal of Linguistics Cambridge University Press, United Kingdom 1997;33:379-430.
21. Peperkamp S, Dupoux, Emmanuel. "Reinterpreting loanword adaptations: The role of perception". Proceedings of the 15th International Congress of Phonetic Sciences, Barcelona, Causal Productions, 2003, 367-370.
22. Prince, Alan, Paul Smolensky. Optimality Theory: Constraint Interaction in Generative Grammar. Rutgers University Center for Cognitive Science Technical Report, 1993.
23. Selkirk E. "On the major class features and syllable theory". In M. Aranoff & R. Oehrle (Eds.), Language sound structure: Studies in phonology presented to Morris Halle by his teacher and students. Cambridge, MA: MIT Press, 1984.
24. Steriade D. "Gestures and autosegments". In M. Beckman, & J. Kingston (Eds.), Papers in laboratory phonology, Cambridge University Press 1990, 382-397.
25. Uffmann, Christian. Vowel Epenthesis in Loanword Phonology. Doctoral Dissertation, Phillips-Universität Marburg 2004.