Natural Language Processing for Tamil TTS

A. G. Ramakrishnan[#], Lakshmish N Kaushik[*], Laxmi Narayana. M[$]

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, INDIA

ramkiag@ee.iisc.ernet.in[#], lakshmish.kowshik@gmail.com[*], mln4u_ece@yahoo.co.in[$]

Abstract

This paper describes the development of the Natural language Processing module for the Tamil TTS Synthesis System. The input to a TTS system is not always pure text and it may contain some acronyms, abbreviations and non-standard words, which need to be first converted to the corresponding Tamil graphemic form. A text normalization module is developed for accomplishing this task. A G2P converter which converts the normalized orthographic text input into its phonetic form is necessary for both the input text and also for the text corpus under consideration. The phonetic transcription helps in identifying and analysing the basic units such as mono-phones, diphones and syllables and also for segmenting the speech corpus. A character to phoneme mapping interface is developed to map the Tamil graphemic text to the corresponding phonetic representation in Roman script. A Rule base is created which contains the inter and intra word rules for changing the default character phone mapping wherever necessary. A proper noun lexicon as well as foreign word lexicon is also incorporated for dealing cases where G2P fails. The NLP module designed is to be used for Tamil TTS synthesis in both Windows platform and Festival (Linux) environment.

1. Introduction

Over 65 million people worldwide speak Tamil, the official language of the south Indian state of Tamil Nadu, and also of Singapore, Sri Lanka and Mauritius. In addition to the above countries, it is spoken in Bahrain, Malaysia, Qatar, Thailand, United Arab Emirates and United Kingdom. Tamil is a syllabic language which contains 12 vowels and 18 consonants. There are five other phones introduced for representing some consonants of Sanskrit. The language has well defined rules, which introduce seven other phones depending on the relative positions of consonants with respect to the vowels or the other consonants. Hence there are 42 phones in the language.

Text to Speech (TTS) synthesis is an automated encoding process, which converts a sequence of symbols (text) conveying linguistic information, into an acoustic waveform (G L J Rama, 2002). The two major components of a TTS synthesizer are - Natural Language Processing Module (NLP), which is capable of producing a phonetic transcription of the given text and Digital Signal Processing module (DSP), which transforms this phonetic transcription into speech (Thierry Dutoit, 1997). One of the characteristics, based on which a TTS system is evaluated, is its ability to accurately synthesize the input text i.e., the input text should be spoken accurately prior to naturalness and expression. The Natural Language Processing module is responsible for the determination of the phonetic transcription of the incoming text. This involves normalizing the input text and mapping the graphemic representation to a corresponding phonetic representation. Since the orthographic representation and pronunciation do not match in some cases, the default mapping needs to be changed wherever necessary. Section 2 describes the need for Text Normalization and how the non-standard words like acronyms and abbreviations in the input text are dealt with in NLP. Section 3 describes the process of converting a pure word sequence into its phonetic equivalent, the inter and intra word rules which are made use of, for G2P conversion, and the creation of Foreign word lexicon. The results and conclusion are presented in section 4.

2. Text Normalization

The text input to the TTS system may not be pure Tamil text. It may contain some non-standard words like acronyms, abbreviations, proper names derived from other languages or clutters, phone numbers, decimal numbers, fractions, ordinary numbers, sequence of numbers, money, dates, measures, titles, times and symbols. The Natural Language Processing module of an advanced TTS should be able to handle such non-standard words also. Standard words are those, whose pronunciation can be obtained from the G2P rules. A G2P converter maps a word to a sequence of phones. All the non-standard words must be expanded into the corresponding Tamil graphemic form before sending to the G2P module for phonetic expansion. This module should also take a decision of how a non-standard word is being pronounced. For example, a phone number should not be read like an ordinary number. Each digit in the phone number must be treated as a single number and must be read in isolation.

The corresponding Tamil graphemic representations of possible non-standard words, English words and Tamil short forms are written in ‘iLEAP’ format. iLEAP is a software, where one could type in many Indian languages. The ISCII (Indian Script Code for Information Interchange) file are exported as an ASCII file (text file) and this file is used by the Text normalization module. The format of the text normalization file is shown in Figure 2.

The input text file is searched for abbreviations or acronyms (can be in Tamil or English). They are replaced

by the corresponding expansion (graphemic form) in Tamil in the output (ISCII) file. This is illustrated in Figure 1.

Ex: *aug* is replaced as ஆகஸ்ட் (normalization)

august is also replaced as ஆகஸ்ட் (Tamil transcription)

Further, there are a number of words used regularly in Tamil, which are originally Sanskrit or English words. The G2P fails to give the accurate phonetic transcription in case of such words. Hence, we have created a lexicon of foreign words.

Number expansion is a 'special' case in normalization, because a *decision* needs to be taken whether the encountered number is an ordinary number or phone number or date or time or currency, etc. If it is currency, it decides whether it is rupee or dollar or pound or yen. The input number is considered as a string. An *ordinary number* is expanded according to the length of the string. A module is written to expand a 3 digit number. The control chooses different paths according to the length of the string. If the number is 4 or 5 digits long, then it must be in thousands or ten thousands. In this case, this module will be called twice, first to convert the number of thousands to words (1 or 2 digits) and next to convert the remaining 3 digit number to words. For example, if the number is 12345, in the first call, the number 12 will be processed and in the next call, 345 is processed. If it is a 6 or 7 digit number, it must be in lakhs or 10 lakhs and if the number has 8 or 9 digits, then it must be in crores or 10 crores; then this module is called thrice or four times, respectively and so on. If the length of the number is less than or equal to 3, then the module is called only once.

If the number string is not an ordinary number, a parameter (a number corresponding to the decision taken) is set according to the type of the number string. If the number string is a decimal number (Ex: 23.8756) the number before the dot (.) is treated as one number and the digits after the dot are spoken in isolation. If the number string is a *date*, the delimiters can be '/' or '-' (Ex: 25-10-1999 or 25/10/1999). All the three values (date, month, and year) are extracted from the input string and processed separately. Similarly the different types of number strings like currency, range of numbers, arithmetic, phone numbers and time are identified by the delimiters present and expanded accordingly.

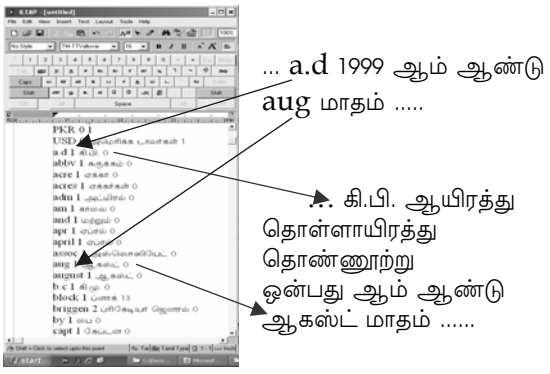

Figure 1: Ex. of Text Normalization using look up table

ph.no	தொலைபேசி எண்
jan	ஜனவரி
rs	ரூபாய்கள்
மி.மீ	மில்லி மீட்டர்

Figure 2: Format of Normalization file

3. Grapheme to Phoneme Conversion

In a TTS system, the G2P module converts the normalized orthographic text input into the underlying linguistic and phonetic representation (Kalika Bali, 2004). G2P conversion therefore is the most basic step in a TTS system (Anumanchipalli, 2005). The text normalization module gives a word sequence as input to the G2P module. The Grapheme to Phoneme conversion of the word sequence can be done using the Letter to Sound Rules. The rules are based on a pronunciation dictionary, in which a mapping of the spelling of a word into a sequence of phones can be found. For example, consider an English word – "speech". The pronunciation dictionary converts this word to the phone sequence – S P IY CH[1]. Traditional orthography in some languages, particularly French and English, often does not coincide with pronunciation. However, in other languages such as Spanish and Italian, there is a consistent relationship between orthography and pronunciation (Link).

If there is no pronunciation dictionary, a simple set of rules to convert the graphemic form of a word into the corresponding phonemic form is used. Using such rules is more relevant for Indian languages. In many cases, there is a direct correspondence between what is written and what is spoken. For example consider a Tamil word - "asiriyar", the corresponding phonetic transcription is - /A/ /s/ /i/ /r/ /i/ /y/ /a/ /r/ which is very similar to the word.

3.1. Tamil Character to Phone Mapping

The grapheme to phoneme (G2P) conversion module receives the sequence of Tamil words, from the Text normalization module, which then are converted to a phonetic transcription represented in Roman script. This is obtained by using a *character to phone mapping,* which gives the corresponding phonemic representation of Tamil graphemes, in Roman script. The Roman character (or sometimes, combination of letters in case of diphthongs like /ae/ or /au/ or genitives like /kk/ and /tt/) which represents a Tamil phonemic unit may not 'sound' exactly like the Tamil phoneme and is used only as an unambiguous representation. The *.wav* (speech) files in the inventory are labeled according to this Roman representation. The DSP module in the TTS engine picks up the corresponding speech units (which are labeled according to the mapping), given by the phonemic representation, for concatenation.

Words are converted one by one. Two kinds of rules – Inter-word rules and Intra-word rules are applied to the input text during conversion. If the last character in a word is *halanth*, and the last but one character is /k/ or /ch/ or /th/ or /p/ and the next word starts with /k/ or /ch/

[1]Courtesy: CMU Dictionary

or /th/ or /p/, respectively, the two words are concatenated. This inter word rule is very relevant here because, while speaking such words, a speaker does not pronounce the phoneme /k/ or /ch/ or /th/ or /p/ two times. Such junctions of words will combine into a single word in which those two phonemes at the junction would manifest as a genitive, for example /p/ /p/ becomes /pp/. The double letters are labeled with a suffix 'l' to the basic phoneme. This is illustrated in Figure 3(a) and 3(b). It was found that the duration of a double letter occurring in the middle of a word and that which was formed at the junction of two words is comparable.

Tamil, like most Indian languages, uses a syllabic script that is largely phonetic in nature. Thus, in most cases, there is a one-to-one mapping between graphemes and the corresponding phones. The architecture of the Natural language Processing module is shown in Figure 4. Language specific information is fed into the system in the form of mapping and rules. The default character to phone mapping is defined in the mapping file. The format of the mapping is shown below and explained subsequently.

Format: *Character Type Class Phoneme*

Example: அ V VOW a ⟶ ¤ V VOW a

Character: The orthographic representation of the character.

Type: Three types of characters are identified, C-Consonant, V-Vowel and H-Halanth

Class: The class to which the character belongs. These class labels can be effectively used to write a rule representing a broad set of characters.

Phoneme: The default phonetic representation of the character.

Type gives a broad classification of the characters while *Class* mainly classifies the C type characters (consonants) into different clusters like KA, CA, TA, tA, PA and YA. Some examples of the default mapping are shown in Figure 5.

3.2. Tamil G2P Rule Base

The Rule Base is a set of *rules* that modify the default mapping of the characters based on the context in which a particular phoneme occurs. Specific contexts are matched using rules. The system triggers the rule that best fits the current context. The rule format is given below.

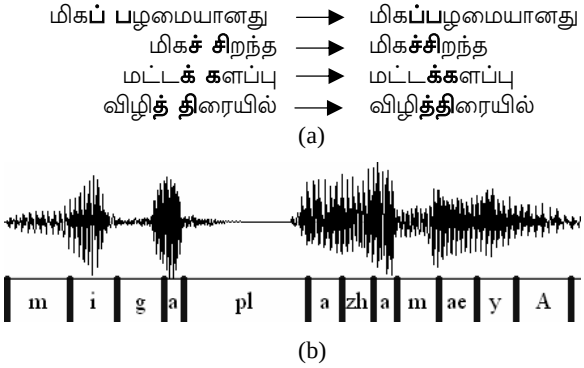

Figure 3(a) Examples of Inter-word rules
(b) /p/ /p/ becoming /pp/ (pl)

Format: $\alpha_1\ \alpha_2\ \ldots.\ \alpha_m\ \{\ \beta_1\ \beta_2\ \ldots\ \beta_n\ \}$

Example: `VOW KA VOW { K:1:X R:2:g K:3:X }`

α_i Class label of the i^{th} character as defined in the character phone mapping. Together these α_is represent the context that is being matched.

β_j j^{th} action specification node. Each such node has the form:

Action_Type *Action_Type:Pos:Phoneme_Str*

This field specifies the type of this action performed at this node. Possible values are *K* (Keep), *R* (Replace), *I* (Insert) and *A* (Append).

Pos The index of the character being covered by the context of the rule ($1 \le Pos \le m$)

Phoneme_Str represents phoneme string output by this action node.

The example rule given above says that if the grapheme /k/ (/k/ belongs to class KA) appears between two vowels (`VOW KA VOW`), keep the first character (vowel) as it is (`K:1:X`), replace the second character(/k/) with /g/ (`R:2:g`) and keep the third character (vowel) as it is(`K:3:X`).

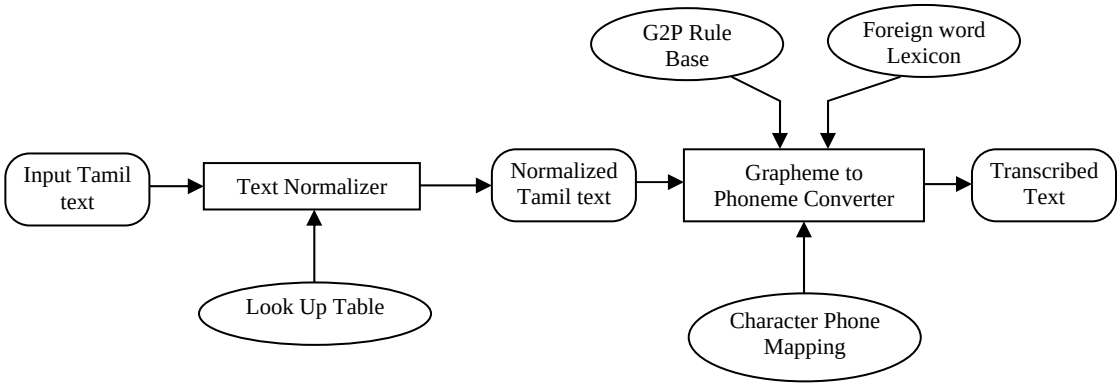

Figure 4: Natural Language Processing (NLP) module

அ	a	க	k
ஒ	o	ஆ	A
த	t	ஓ	O
ட	T	எ	e

Figure 5: Mapping examples

If the same consonant occurs twice consecutively, the genitive is represented by replacing the second grapheme by 'l'.

Ex: `TA HAL TA { A:1:l } (T T -> Tl)`

The reason for using this kind of representation for double letters is, the speech files in the inventory are labeled accordingly and if 'T T' is kept as it is, then the DSP module selects two 'T' segments instead of selecting a single 'Tl' segment. Also, linguistically, TT is a single phone (genitive), not two.

Only the letters /k/, /ch/, /T/, /th/, /p/ exist in Tamil script. But these five graphmes would manifest as /g/, /j/, /D/, /dh/, /b/, respectively if they occur between two vowels or prefixed by nasals.

Ex: `NAS1 HAL KA { K:1:X K:2:X R:3:g }`
`VOW PA VOW { K:1:X R:2:b K:3:X }`

However, there is an exception for /ch/. If it occurs between two vowels, it becomes /s/. Also, /ch/ becomes /s/ when it comes in the beginning of a word. Some more rules are listed in the Appendix. Figure 6 gives some sample input Tamil sentences, the normalized Text and the phonetised text.

3.3. Foreign Word Lexicon

The process of corpora phonetization or the development of phonetic lexicons for the western languages is traditionally done by linguists. These lexicons are subject to constant refinement and modification. But the phonetic nature of Indian scripts reduces the effort to building mere mapping tables and rules for the phonetic representation. These rules and the mapping tables together comprise the phonetizers or the Grapheme to Phoneme converters (Anumanchipalli, 2005).

The G2P rule base cannot be generalized to handle all the words in the input text, especially for the proper nouns derived from other languages like Sanskrit and other clutters. For example, the word 'Buddha' is written in Tamil as 'புத்தா' (putla); the grapheme /p/ in the beginning of the word should be pronounced as /b/ and the /tl/ should be pronounced as /dl/. But there are no such rules in the rule base, since it is basically a Sanskrit word used as it is in Tamil. If a rule is introduced for this purpose, that will effect other words. For example, the Tamil word 'புத்தகம்' (putlakam) is pronounced as 'putlagam' only. The initial /p/ doesn't change to /b/ or the genitive /tl/ doesn't change to /dl/.

So, an exception lexicon is created, which contains the phonetic transcription of such words and proper nouns. The lexicon dictates the phonetic composition, or the pronunciation of each entry in the list.

The G2P first looks for each input word in the lexicon file. If the word is present in the lexicon file, its phonetic transcription is taken from the lexicon itself. Otherwise, the G2P applies mapping and the rules to produce the phonetic transcription.

The phonetic transcription generated by the G2P converter can be used for segmentation of the speech corpus. The phonetic transcription can be aligned with the speech waveform and the phone boundaries can be adjusted manually or by automatic speech segmentation algorithms.

4. Results and Conclusion

The Natural Language Processing module for Tamil Text to Speech Synthesis has been developed effectively. Efficient rules have been designed in for grapheme to phoneme conversion, which cover most of the contexts in which the default mapping needs to be changed. A foreign word lexicon is created to handle cases where the general G2P rules do not give the exact phonetic transcription. This has been used in a Tamil TTS developed around Festival. The developed C code for NLP module is designed to be used for Tamil Text to speech synthesis in both Windows and Linux platforms.

திரிவேணியின் பிறந்த தேதி september 1, 1928. பல pages கொண்ட 21 நாவல்கள், 41 சிறுகதைகள் அடங்கிய 3 தொகுப்புகள், கன்னட இலக்கியத்துக்கு அவருடைய பங்களிப்புகள்.

(a)

திரிவேணியின் பிறந்த தேதி ஸெப்டெம்பர் ஒன்று / ஆயிரத்து தொள்ளாயிரத்து இருபத்து எட்டு பல பக்கங்கள் கொண்ட இருபத்து ஒன்று நாவல்கள், நாற்பத்து ஒன்று சிறுகதைகள் அடங்கிய மூன்று தொகுப்புகள், கன்னட இலக்கியத்துக்கு அவருடைய பங்களிப்புகள்.

(b)

```
> t i r i w E N i y i n # p i R a n d a # t E d
i # s e p T e m b a r # o n R u # # A y i r a tl
u # t o Ll A y i r a tl u # i r u b a tl u # e
Tl u # # p a l a # p a kl a ng g a L # k o N D a
# i r u b a tl u # o n R u # n A w a l g a L $ #
n A R p a tl u # o n R u # s i R u g a d ae g a
L # a D a ng g i y a # m U n R u # t o g u pl u
g a L $ # k a nl a D a # i l a kl i y a tl u kl
u # a w a r u D ae y a # p a ng g a L i pl u g a
L <
```

(c)

Figure 6: (a) Input Tamil text in ISCII format (b) Text after Normalization (c) Phonetic transcription (G2P Converter Output)

5. References

Anumanchipalli Gopalakrishna, Rahul Chitturi, Sachin Joshi, Rohit Kumar, Satinder Singh, R.N.V Sitaram and S.P. Kishore, 2005. *Development of Indian Language Speech Databases for Large Vocabulary Speech Recognition Systems*, Proceedings of International Conference on Speech and Computer (SPECOM), Patras, Greece, Oct.

Link (http://en.wikipedia.org/wiki/Phonetic_transcription)

Thierry Dutoit, 1997. *High-quality Text-To-Speech synthesis: an overview*, Journal of Elec. and Electronics Engineering, Australia: Special Issue on Speech Recognition and Synthesis, vol. 17 no 1, pp. 25-37.

G. L. Jayavardhana Rama, A. G. Ramakrishnan, R. Muralishankar and P. Prathibha, ``A Complete Text-to-Speech Synthesis System in Tamil," Proc. IEEE 2002 Workshop on Speech Synthesis, Sep. 11-13, Santa Monica, CA USA, 2002.

Kalika Bali, Partha Pratim Talukdar, N. Sridhar Krishna, A.G. Ramakrishnan,"Tools for the Development of a Hindi Speech Synthesis System", In 5th ISCA Speech Synthesis Workshop, Pittsburgh, pp.109-114, 2004.

6. Appendix

6.1. Character to Phone Mapping

ு	V	IG	X
அ	V	VOW	a
ஆ	V	VOW	A
இ	V	VOW	i
ஈ	V	VOW	I
�ு	V	VOW	u
ூ	V	VOW	U
எ	V	VOW	e
கூ	V	VOW	E
ஈ	V	VOW	ae
கூ	V	VOW	o
ஈா	V	VOW	O
கூள	V	VOW	au
ணா	C	KA	k
ணா	C	KA	k

க	C	KA	k
க	C	KA	k
ங	C	NAS1	ng
ச	C	CA	s
ச	C	CA	s
ஜ	C	CA	s
ஜ	C	CA	s
ஞ	C	NAS2	ny
ட	C	TA	T
ட	C	TA	T
ட	C	TA	T
ட	C	TA	T
ண	C	NAS3	N
த	C	tA	t
த	C	tA	t
த	C	tA	t

த	C	tA	t
ந	C	NAS4	n
ன	C	NAS4	n
ப	C	PA	p
ப	C	PA	p
ப	C	PA	b
ப	C	PA	b
ம	C	NAS5	m
ய	C	YA	y
ர	C	NL4	r
ற	C	RA	R
ல	C	La	l
ள	C	La	L
ழ	C	NL	zh
வ	C	WA	w

ஷ	C	NL1	S
ஷ	C	NL1	S
ஸ	C	NL2	s
ஹ	C	NL3	h
ா	V	VOW	A
ி	V	VOW	i
ீ	V	VOW	I
ு	V	VOW	u
ூ	V	VOW	U
ெ	V	VOW	e
ே	V	VOW	E
ை	V	VOW	ae
ொ	V	VOW	o
ோ	V	VOW	O
ௌ	V	VOW	au

6.2. G2P Rule base

```
IG PA { R:2:F }
ta VOW ka { K:1:X K:2:X R:3:h }
KA HAL KA { A:1:l }
CA HAL CA { R:1:cl }
TA HAL TA { A:1:l }
ta HAL ta { A:1:l }
PA HAL PA { A:1:l }
YA HAL YA { A:1:l }
la HAL la { A:1:l }
WA HAL WA { A:1:l }
La HAL La { A:1:l }
NL HAL NL { A:1:l }
NL1 HAL NL1 { A:1:l }
NL2 HAL NL2 { A:1:l }
NL3 HAL NL3 { A:1:l }
NL4 HAL NL4 { A:1:l }
NA HAL NA { A:1:l }
NAS HAL NAS { A:1:l }
NAS HAL KA { K:1:X K:2:X R:3:g }
NAS HAL CA { K:1:X K:2:X R:3:j }
NAS HAL TA { K:1:X K:2:X R:3:D }
NAS HAL tA { K:1:X K:2:X R:3:d }
NAS HAL PA { K:1:X K:2:X R:3:b }
YA HAL KA { K:1:X K:2:X R:3:g }
NL4 HAL KA { K:1:X K:2:X R:3:h }
la HAL KA { K:1:X K:2:X R:3:g }
WA HAL KA { K:1:X K:2:X R:3:g }
NL HAL KA { K:1:X K:2:X R:3:g }
La HAL KA { K:1:X K:2:X R:3:g }
YA HAL CA { K:1:X K:2:X R:3:j }
NL4 HAL CA { K:1:X K:2:X R:3:j }
la HAL CA { K:1:X K:2:X R:3:j }
WA HAL CA { K:1:X K:2:X R:3:j }
NL HAL CA { K:1:X K:2:X R:3:j }
La HAL CA { K:1:X K:2:X R:3:j }
YA HAL TA { K:1:X K:2:X R:3:D }
NL4 HAL TA { K:1:X K:2:X R:3:D }
la HAL TA { K:1:X K:2:X R:3:D }
WA HAL TA { K:1:X K:2:X R:3:D }
NL HAL TA { K:1:X K:2:X R:3:D }
La HAL TA { K:1:X K:2:X R:3:D }
YA HAL tA { K:1:X K:2:X R:3:d }
NL4 HAL tA { K:1:X K:2:X R:3:d }
la HAL tA { K:1:X K:2:X R:3:d }
WA HAL tA { K:1:X K:2:X R:3:d }
NL HAL tA { K:1:X K:2:X R:3:d }
La HAL tA { K:1:X K:2:X R:3:d }
YA HAL PA { K:1:X K:2:X R:3:b }
NL4 HAL PA { K:1:X K:2:X R:3:b }
la HAL PA { K:1:X K:2:X R:3:b }
WA HAL PA { K:1:X K:2:X R:3:b }
NL HAL PA { K:1:X K:2:X R:3:b }
La HAL PA { K:1:X K:2:X R:3:b }
VOW KA VOW { K:1:X R:2:g K:3:X }
VOW TA VOW { K:1:X R:2:D K:3:X }
VOW ta VOW { K:1:X R:2:d K:3:X }
VOW PA VOW { K:1:X R:2:b K:3:X }
NAS KA { K:1:X R:2:g }
NAS ta { K:1:X R:2:d }
NAS TA { K:1:X R:2:D }
NAS PA { K:1:X R:2:b }
VOW CA VOW { K:1:X R:2:s K:3:X }
NAS CA { K:1:X R:2:j }
RA HAL RA { R:1:TR }
ta ka { K:1:X R:2:h }
HSH { }
HAL HAL ( }
```