International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Idhazhi: A Min-Max Algorithm for Viseme to Phoneme Mapping

Suriyah M., Aarthy Anandan, Madhan Karky Vairamuthu

Abstract: ***With the advent of language tools, the barrier of language is diffusing fast. Content from one language is converted to other languages in different platforms especially in the entertainment industry. In this age, it is essential to increase the naturalness of syncing video and audio content while dubbing. It is a proven fact that a majority of people around the world prefer listening to audio content rather than reading in the language known to them. Detecting visemes sequences of words of a language in videos and identifying words from another language matching the detected viseme sequence becomes a valuable application in this scenario. In this paper we propose Idhazhi, an algorithm which suggests words as phonemes that match a particular viseme sequence, in four levels – perfect, optimized, semi-perfect and compacted. This is done by mapping specific oral positions of lips, teeth and tongue for the symbols of the IPA. The system was tested by recording 50 videos of testers pronouncing a word in each, playing it muted to a group 12 of testers who evaluated how much the words suggested by the system are relevant to the viseme sequence in the video; the accuracy was 0.73 after approximations. Apart from the suggested application, this algorithm can have wide application in security for finding out the list of words which may match a viseme sequence in a video like CCTV footage. It may also be extended to help persons with vocal disability by generating speech from their vocal movements.***

Keywords: ***Dubbing, Natural language processing, viseme-phoneme mapping.***

I. INTRODUCTION

Tremendous growth of language tools has facilitated the shrinking of language boundaries. Enormous amount of content across different languages is made available to audience across different cultures through the internet. Visual (video or image) content is universally understandable; audio content can be comprehended only by people understanding the language.

Apart from subtitling and lip sync dubbing, Koolstra et al [1] list three methods to adapt video content from one language to another.

Revised Manuscript Received on February 28, 2020.

***** Correspondence Author

Suriyah M.*, Research Writer, Department of Computer Science and Engineering, Karky Research Foundation, Chennai, India.

Aarthy Anandan, Software Developer, Department of Computer Applications, Karky Research Foundation, Chennai, India.

Madhan Karky Vairamuthu, Founder and Research Head, Department of Computer Science, Karky Research Foundation, Chennai, India.

They are off-screen narration, voice over and intertitles; but these methods are used for specific programs only. Kilborn [2] states that subtitling and dubbing are the most favorite methods to adapt foreign language programmes to domestic market.

Subtitling keeps the voice of the actors thus retaining the original flavor, but is useful for the educated only. Subtitles can also be distracting to some extent, diverting attention from the actors' performances. Dubbing enables everyone to understand the audio of the video content and is not distracting too, but the audience do not experience the original flavor of emotions conveyed by the voice of the actor. Subtitling is preferred in circumstances when there is a lack of writers who efficiently translate the dialogues in a language to another with the lip sync and timing intact. The cost of subtitling is much lesser than that of dubbing.

English movies have audience worldwide and their market can be improved by dubbing them in regional languages. Avengers endgame managing to earn Rs.400 crores in less than two weeks in India [3] stands proof of the demand for these movies across the globe. When dubbing a movie to another language, there is the task of finding words with the same meaning and similar viseme to the ones uttered by characters on screen. This requires expertise in both languages – the original language in which the movie was shot and the language to which the movie is getting dubbed.

A grapheme is the smallest unit of writing system in a language. It may be a symbol of the alphabet, numbers, punctuation marks etc; a phoneme is the basic unit of speech in a language which differentiates one word from another. A grapheme may or may not correspond to a single phoneme. Visemes are the visual equivalents of phonemes. They may be defined as a set of phonemes which have similar lip movements. A phoneme has only one corresponding viseme whereas a viseme may correspond to multiple phonemes [4].

Idhazhi, a system to suggest words which match a particular viseme sequence is proposed in this paper. This system finds words matching the viseme sequence of a particular word, in four levels - *perfect*, *optimized*, *semi-perfect* and *compacted*. The efficiency of the system is tested by recording videos of testers pronouncing a word and another group of testers evaluating how much the words suggested by the system are relevant for the lip movement in the video which is played muted.

In the UK, subtitling is preferred to dubbing when it comes to adapting foreign audio-visual content. However, Channel 4 of UK broadcasted twenty-six episodes of French soap *Châteauvallon* twice a week – once with subtitles and next dubbed. In spite of dubbing being considered inferior, all age groups of viewers showed significant preference to the dubbed version [5].

This establishes that dubbed versions are naturally preferred than subtitles.

II. LITERATURE SURVEY

Faizal M and Manzoor S [6] demonstrate that to improve speech recognition in noisy environments, visual information can be used. They do so by training two deep learning models – one for video and another for audio processing. Buchman et al [7] recorded a list of French words and minimal pairs in two modes – vocalized and non-vocalized and compared the amplitude and timing of articulatory movements. This was done to study silent speech communication after total laryngectomy rehabilitation. The non-vocalized mode showed reduction in the duration of words and hypoarticulation of lips depending on the vowel or consonant pronounced. The tongue movements however did not show substantial differences. Video clips of a million words, spoken by over thousands of people were collected from television broadcasts by Chung and Zisserman [8]. A two-stage convolutional neural network which learns the joint embedding between the uttered sound and the mouth movements from unlabeled data was developed to be used for speaker detection and audio to video synchronization. Chung and Zisserman also train convolutional and recurrent networks that learn and recognize over a hundred words from the dataset they collected. For computer recognition of spoken words, Suresh and Bhajantri [9] propose lip surface area as a feature. They propose measuring the horizontal and vertical distances between lips and calculating the elliptical lip surface area. When a movie is dubbed, the viseme sequence of the original language may not be synchronized completely with that of the language to which it is dubbed. This may lead to visual discomfort of the audience. VDub [10] is a system designed to alter the mouth movement of an actor to match the viseme of the language to which the movie is to be dubbed. The mouth portion of the actor is photo realistically rendered in high detailed 3D.

III. PROBLEM STATEMENT

To find a subset of words from a language that match a particular viseme pattern, minimizing the search space and maximizing the accuracy of viseme to phoneme identification.

IV. PROPOSED METHODOLOGY

Lips, teeth, alveolar ridge, hard palate, soft palate, uvula, glottis and tongue are the organs used in speech [11]. Idhazhi system considers the movements of lips, teeth and tongue only as the other organs do not contribute to the visual oral movement in a video.

Lips take five positions – *closed, near closed, open, big o* and *small o*. Teeth take three positions – *near closed, touch lower lip* and *open* whereas tongue takes three positions – *top, down* and *mid*.

Table-I. Idhazhi visemes description

Organ	Position
Lips	Closed
	Near closed
	Open
	Big o
	Small o
Teeth	Near closed
	Touch lower lip
	Open
Tongue	Top
	Mid
	Down

With these positions of the three organs taken into consideration, 135 combinations are possible. But Idhazhi system uses only 11 combinations which are sufficient to represent visual oral movement. These oral positions for the symbols of the International Phonetic Alphabet (IPA) [12] are studied. IPA is an alphabet system for phonemes based on the Latin alphabet. It can be used across languages to represent pronunciation. The symbols of the IPA can be accommodated in 11 combinations of the lip, teeth and tongue positions.

The words matching the viseme pattern of a given word are found in four levels using three different mappings illustrated in Table II.

In the *perfect* category, each lip, tongue, teeth combination is given a number from the hexadecimal number system. In *optimized* and *semi-perfect* categories, the combinations are grouped to reduce the search space and bring about approximation.

Let $W = \{w_1, w_2, \ldots w_{|W|}\}$ be the set of words in a language and $P = \{p_1, p_2, \ldots p_{|P|}\}$ be the set of patterns. $|W| > |P|$ the relationship between patterns and words is one to many. There can be many words sharing a single pattern. For example, the words *Incense, enlist* and *intend* share the same pattern, whereas *Dress, slide, slice, tress* and *sneeze* share a different pattern.

Table-II. Idhazhi mappings in three levels

Mappings			Oral positions			Image
Semi perfect	*Optimized*	*Perfect*	*Lips*	*Teeth*	*Tongue*	
1	1	1	close	open	down	
1	1	2	close	open	mid	
2	2	3	near close	near close	mid	
2	2	4	near close	near close	down	
2	2	5	near close	open	down	
1	3	6	near close	touch lower lip	down	
3	4	7	near close	near close	top	
4	5	8	o big	open	down	
4	6	9	o small	near close	mid	

4	6	a	o small	near close	down	
3	7	b	open	open	down	

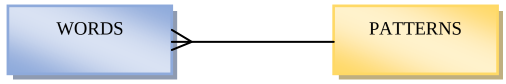

Fig. 1. Relationship between words and patterns

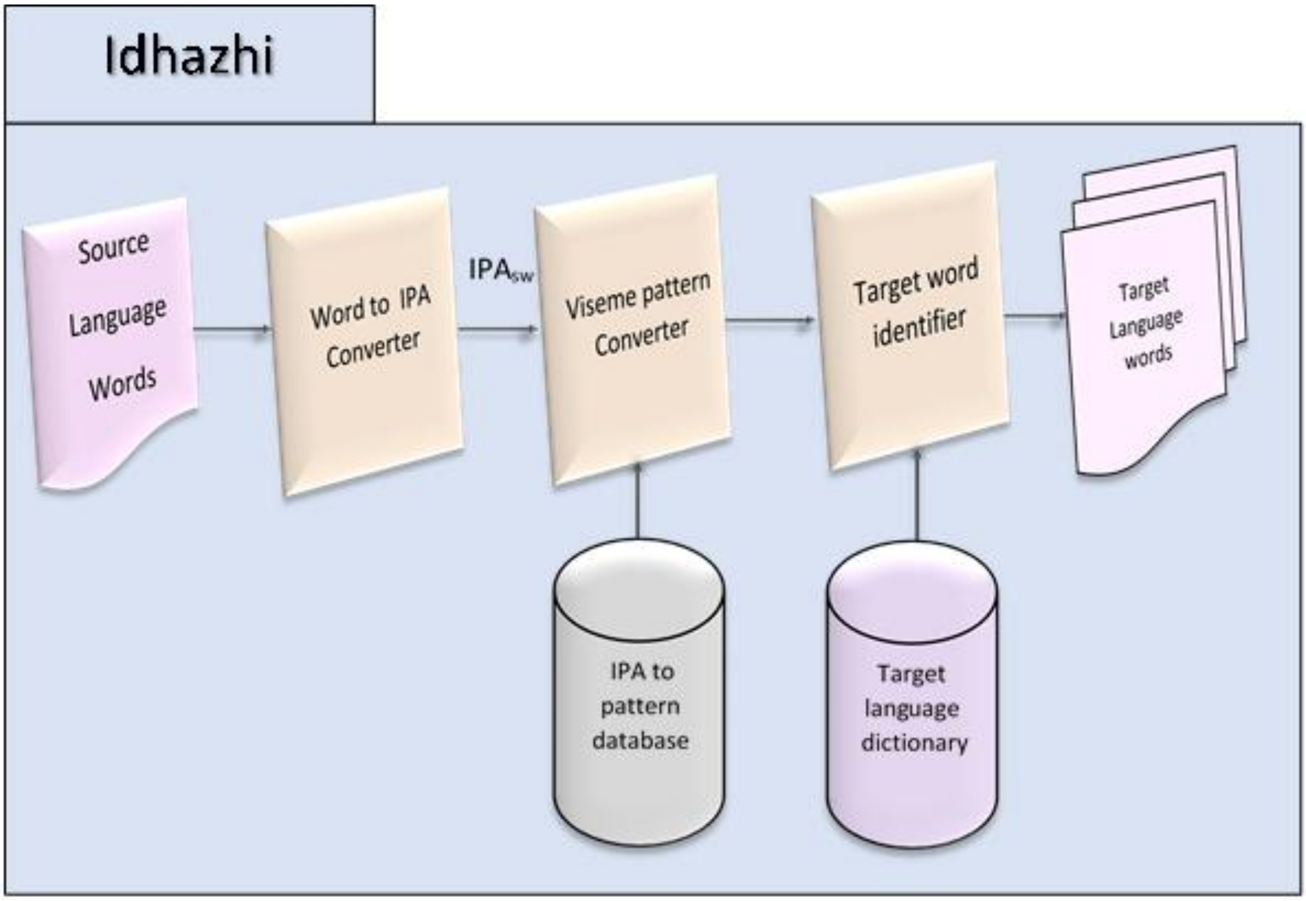

Fig. 2. System Architecture

Let P^p be the set of patterns for words in W taken in the *perfect* level. Let P^s and P^o be the set of patterns taken for words in W at the *semi perfect* and *optimized* levels. Comparing the cardinalities,

$$|W| > |P^p| > |P^s|$$

$$|W| > |P^p| > |P^o|$$

The process of finding words with matching visemes in four levels is explained below.

Let us consider the Hindi word "□□□□□" meaning story whose IPA equivalent is "*kəɦaːniː*". For this IPA equivalent, three Idhazhi patterns are found out using *perfect, optimized* and *semi-perfect* mappings. The three patterns are found out using the lip, tongue and teeth positions of the letters of IPA and the perfect, semi-perfect and optimized mappings. For the above-mentioned example, the patterns are as follows.

word ← □□□□□; word$_{IPA}$ ← *kəɦaːniː*

pattern$_{perfect}$ ← 733b73

pattern$_{optimized}$ ← 422742

pattern$_{semi\text{-}perfect}$ ← 322332

Now, to find the list of other language words whose viseme pattern matches with that of □□□□□, the words in the Tamil dictionary would have been indexed with their *perfect*, *optimized* and *semi-perfect* patterns. Those words whose Idhazhi patterns match with the input's patterns in the three categories are returned as words with matching viseme sequence at the respective levels. Hence, we have three sets of words matching the inputs viseme sequence at the three levels.

For finding the list of words at the fourth level, a process called *compaction* is carried out on the input's semi-perfect pattern. Compaction is done by condensing the continuous occurrences of a digit in the semi-perfect pattern of the input to one digit. For example, if the pattern$_{semiperfect}$ of an input word is 611111123445888, the continuous occurrences of 1, 4 and 8 are replaced with one occurrence each to get the compacted pattern (pattern $_{compact}$) 6123458.

Now, those words in the target language whose $pattern_{compact}$ is 6123458 are returned as words matching the viseme sequence of the input at the fourth level. Compaction is used to make up the differences in the pace of speaking. It is done at the semi-perfect level to increase the solution space and bring about approximation.

V. SYSTEM OVERVIEW

The components of the Idhazhi system are outlined in this section.

A. Source word

The word from the source language for which words matching the viseme pattern are to be identified from the target language.

B. Word to IPA converter

IPA [11] is the International phonetic alphabet system which is represents phonemes in Latin alphabet. This module yields the IPA equivalent of the given source word (IPA_{sw}).

C. IPA to pattern database

This module contains the lip, teeth and tongue positions described in Idhazhi for the phonemes of IPA.

D. Viseme pattern converter

Using the IPA equivalent of the source word (IPAsw) and the data in the IPA to viseme pattern repository, this module yields the Idhazhi mapping string for the source word given as input.

E. Target language dictionary

The words of the target language indexed with the respective Idhazhi pattern strings are stored in this module.

F. Target word identifier

Using the Idhazhi pattern string of the source language word and the *target language dictionary*, the words matching the source word are picked from the target language by this module. There are three stages of matching – *perfect, semi perfect* and *imperfect*.

VI. PROCESS DESCRIPTION

Let $S = \{s_1, s_2, \ldots s_{|S|}\}$ be the set of words in the source language

$T = \{t_1, t_2, \ldots t_{|T|}\}$ be the set of words in the target language

$P^p = \{p^p{}_1, p^p{}_2, \ldots p^p{}_{np}\}$ be the set of patterns in perfect level where $np = |P^p|$

$P^s = \{p^s{}_1, p^s{}_2, \ldots p^s{}_{ns}\}$ be the set of patterns in semi perfect level where $ns = |P^s|$

$P^o = \{p^o{}_1, p^o{}_2, \ldots p^o{}_{no}\}$ be the set of patterns in optimized level where $no = |P^o|$

$P^c = \{p^c{}_1, p^c{}_2, \ldots p^c{}_{nc}\}$ be the set of patterns in optimized level where $nc = |P^c|$

Table-III. Symbols used in the algorithm

Symbol	Description
S	The set of words in the source language
T	The set of words in the target language
s	Word from the source language input to the system for which words from the target language with matching viseme sequence are to be found.
s^i	IPA equivalent of s
s^p	Pattern of s in perfect category.
s^o	Pattern of s in optimized category.
s^s	Pattern of s in semi-perfect category.
s^c	Compacted perfect pattern of s
T^p	Set of words from the target language matching the viseme sequence of s in perfect category.
T^o	Set of words from target language matching the viseme sequence of s in optimized category.
T^s	Set of words from the target language matching the viseme sequence of s in semi-perfect category.
Tc	Set of words from the target language whose perfect pattern matches with the compacted pattern of s.

Table-IV. List of functions

Function	Input	Output
IPA(s)	Input word from the source language	The IPA equivalent of the input word
PER(s^i)	The IPA equivalent of the input word	The perfect level Idhazhi pattern of input
SP(s^i)	The IPA equivalent of the input word	The semi-perfect level Idhazhi pattern of input
OPT(s^i)	The IPA equivalent of the input word	The optimized level Idhazhi pattern of input
COMP(s^s)	The semi-perfect level Idhazhi pattern of the input word	The compacted level Idhazhi pattern of input
MW(p, l)	p - The pattern of the input word in any of the four levels. l - The numeric value which denotes the level of the pattern string passed as arg1 level is perfect; if l = 0 semi-perfect; if l =1 optimized; if l =2 compacted; if l =3	The set of words from the target language which match the pattern passed as p in level passed l

The sequence of operations are as follows.

i. Input $s \in S$

ii. $s^i \leftarrow$ IPA(s);

iii. $s^p \leftarrow$ PER(s^i);

Retrieval Number: D1414029420/2020©BEIESP
DOI: 10.35940/ijitee.D1414.029420
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$s^p \in P^p$

iv. $s^s \leftarrow$ SP(s^i); $s^s \in P^s$

v. $s^o \leftarrow$ OPT(s^i); $s^o \in P^o$

vi. $s^c \leftarrow$ COMP(s^s); $s^c \in P^c$

vii. $T^p \leftarrow$ MW(s^p , 0); $T^p \subseteq T$

viii. $T^s \leftarrow$ MW(s^s , 1); $T^s \subseteq T$

ix. $T^o \leftarrow$ MW(s^o, 2); $T^o \subseteq T$

x. $T^c \leftarrow$ MW(s^c, 3); $T^c \subseteq T$

The entire process occurs in two stages – offline and online. In the offline stage, the words from the source and target languages are indexed with their respective Idhazhi patterns in three categories – perfect, optimized and semi-perfect.

In the online stage, the source language word for which target language words with same or similar viseme are to be found is input to the system. Its IPA equivalent is found out by the Word to IPA converter and the corresponding Idhazhi pattern is calculated by the Viseme pattern converter modules. The Target word identifier identifies the words from the target language which match with the viseme of the input. This identification can be in either of the four levels - perfect, optimized, semi perfect or compact based on user preference.

VII. TESTING SETUP

To evaluate the efficiency of the system, five languages - English, Tamil, French, Telugu and Hindi, are considered. 567300 words are used for testing. 50 words (10 from each language) are input to the system and the output words at all levels are saved. For the result set of each input, 4 words are picked at random from the respective input word's output and the remaining 1 is a word not output by the system. The oral movements of the last word do not match with that of the input word.

Let $I = \{i_1, i_2, \ldots i_{50}\}$ be the set of words input to the system.

$R_j = \{r^j_1, r^j_2, \ldots r^j_5\}$ be the sets of five words picked as result set of i_j where $(0 < j <= 50)$

$\forall\ i_j \in I$, videos of testers reading i_j are shot $(0 < j <= 50)$.

Let $V^I = \{v_1, v_2, \ldots v_{50}\}$ be the set of video files which have recordings of all the elements of I.

Another group of testers are made to view $v_j \in V^I$ without audio and evaluate $r_j^k \in R_j$, depending on how much the oral movements of r_j^k match with that of v_j where $(0 < k <= 5$ and $0 < j <= 50)$. Each r_j^k is to be evaluated by the testers to belong to either of 5 categories - *bad*, *average*, *good*, *very good* and *perfect*. These are mapped to the levels in Idhazhi and user scores as shown in Table-V.

For example, let the telugu word [సాధారణ | saaDhaaraNa] be the input which is uttered in the video played muted. 4 words output by the system in various categories are given as options to the testers - [సాధారణ | saaDhaaraNa] in *perfect*, [నసనస | nasanasa] in *optimized*, [science | SaienS] in *compacted* and [□□□□□□□□□ | adarthal] in *semi-perfect*. The word [beaucoup | boku] in *imperfect* as it is a word not output by the system and its viseme pattern does not match with that of the input word. The testers have to give a whole number as score on the scale of 0-4 (mapping to specific system categories as in Table-V) denoting the appropriateness of the words to the viseme sequence in the video for each of the options. The scores and the category given by the system are compared for testing. This will give insight to the performance of the system.

Table-V. Idhazhi levels mapped with the categories in evaluation

System category	User Category	Score
No match	Bad	0
Compacted	Average	1
Semi-perfect	Good	2
Optimized	Very good	3
Perfect	Perfect	4

VIII. RESULT AND DISCUSSION

For the 50 words which were given as input to the system, 5 words each (4 output by the system and 1 random word not matching the viseme) are given as options. 12 testers are made to score the options on a scale of 0-4 as illustrated in Table-V based on their appropriateness to the visemes of the words in the videos. 50 words input to the system are given 5 options each. 12 testers participated in the process. This amounts to 3000 (50 * 5 * 12) word instances being checked for appropriateness to the visemes in the videos. The levels output by the system and the testers' categories for the respective words are studied. The confusion matrix of the same is illustrated in Table-VI. The rows correspond to system levels while the columns correspond to the categories marked by the testers. The elements of the diagonal in matrix ([0,0], [1,1], [2,2], [3,3] and [4,4]) correspond to the number of instances for which the testers' categories and system levels match. The [0,0] category corresponding to instances which do not match with the viseme in the video and the [4,4] category corresponding to instances with perfect match with the viseme in the video have accuracies of 0.9 and 0.6 respectively. The compacted, semi-perfect and optimized categories have an accuracy rate of 0.13, 0.14 and 0.24 respectively. The overall accuracy of the system is 0.43. The comparison of the actual number of instances with the number of matches in system and user categories is illustrated in Fig 3.

The reduced accuracy for the compacted, semi-perfect and optimized categories may be attributed to the testers' lack of familiarity with the languages used and the absence of some sounds in certain languages. This is compensated by approximation based on proximity window.

Table-VI. Confusion matrix

	0	1	2	3	4
0	554	25	12	5	4
1	289	94	85	124	80
2	249	59	83	111	74
3	127	35	58	116	132
4	46	16	38	127	457

Table-VI. Category-wise results

	No. of actual instances	No. of instances with match of system levels and testers' categories	Accuracy
0	600	554	0.9233
1	672	94	0.1399
2	576	83	0.1441
3	468	116	0.2479
4	684	457	0.6681

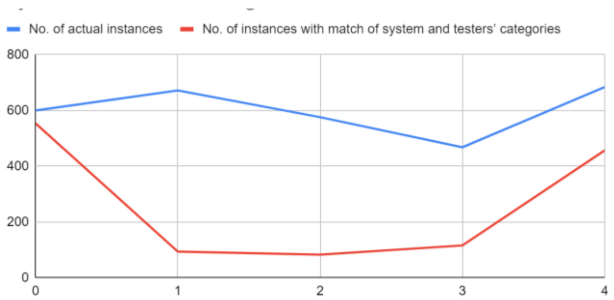

Fig. 3. Comparison of system and testers' categorization

The proximity windows in the confusion matrix are shown in Table-VII.

When calculating the accuracies of each category, the entries in the proximity window are also considered as positives. Rewriting Table VI with this approximation, we get Table-VIII and Fig.4.

From Table-VIII, it is seen that the accuracy of all the categories have improved after approximation. The overall accuracy has also increased to 0.73; it was 0.43 previously. The comparison of the number of instances categorized under a specific category after approximation illustrated in Fig.4 is better than what was before the approximation illustrated in Fig.3.

Table-VII. Proximity window for each category

	0	**1**	**2**	**3**	**4**
0	554	25	12	5	4
1	289	94	85	124	80
2	249	59	83	111	74
3	127	35	58	116	132
4	46	16	38	127	457

Table-VIII. Approximated category-wise instances

	Approximated positives	Total positives	Accuracy
0	579	600	0.965
1	468	672	0.696
2	253	576	0.439
3	306	468	0.654
4	584	684	0.854

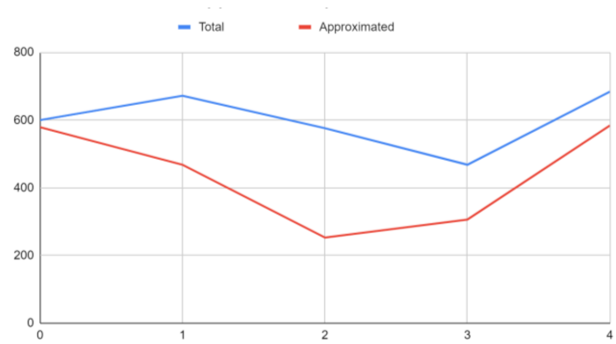

Fig. 4. Comparison of system and testers' categorization after approximation with proximity window

IX. CONCLUSION

This system attempts to list words from different languages which match a viseme sequence. Idhazhi system uses 11 combinations of tongue, teeth and lip positions to represent visual oral movements of the International Phonetic Alphabet. Lips take five positions – *closed*, *near closed*, *open*, *big o* and *small o*. Teeth take three positions – *near closed*, *touch lower lip* and *open* whereas tongue takes three positions – *top*, *down* and *mid*. The system finds four levels of matching words for a viseme sequence - *perfect*, *semi-perfect*, *optimized* and *compacted*. Since Idhazhi works on IPA, it makes it easy to extend it across different languages by incorporating language to IPA conversion.

The system was tested with 567300 words from five languages - Tamil, Telugu, English, Hindi and French. The overall accuracy was 0.43; it improved to 0.73 when an approximation strategy, the proximity window was applied.

Apart from dubbing, this tool may be extended to be used in the field of security for finding out the list of words which may match a viseme sequence in a video like CCTV footage. Idhazhi can also be improved to help persons with vocal disability by generating speech from their vocal movements.

REFERENCES

1. Koolstra, Cees M., Allerd L. Peeters, and Herman Spinhof. "The pros and cons of dubbing and subtitling." European Journal of Communication 17, no. 3 (2002): 325-354.
2. Kilborn, R. (1993) 'Speak my Language: Current Attitudes to Television Subtitling and Dubbing', Media, Culture and Society 15: 641–60.
3. HT Correspondent. "*Avengers Endgame India box office: Marvel film grosses Rs 400 cr, sets new benchmarks***".** https://www.hindustantimes.com/hollywood/avengers-endgame-india-box-office-marvel-film-grosses-rs-400-cr-sets-new-benchmarks/story-HM7w4GOz9csbozs7ACmL1M.html (14th November 2019)
4. Bear, Helen & Harvey, Richard. (2017). Phoneme-to-viseme mappings: The good, the bad, and the ugly. Speech Communication. 10.1016/j.specom.2017.07.001.
5. Mera, Miguel (1998) "Read my lips: Re-evaluating subtitling and dubbing in Europe" Links & Letters 6, 1999, pp.73-85.
6. Deep Learning for Lip Reading using Audio-Visual Information for Urdu Language, Muhammad Faisal and Sanaullah Manzoor
7. Lise Crevier-Buchman, Cédric Gendrot, Bruce Denby, Claire Pillot-Loiseau, Pierre Roussel, Antonia Colazo-Simon, Gérard Dreyfus, "Articulatory strategies for lip and tongue movements in silent versus vocalized speech", Colazo-simon, Antonia. "Articulatory Strategies for Lip and Tongue Movements in Silent versus Vocalized Speech." Proceedings of the 17th International Congress of Phonetic Science (ICPhS) 2011, 2011.

8. Joon Son Chung, Andrew Zisserman. "Learning to Lip Read Words by Watching Videos". Computer Vision and Image Understanding, 2018
9. System M. R. Suresh, Dr. Nagappa U. Bhajantri. "Lip movement features point Extraction for lip reading", International Journal of Engineering Development and Research, Volume 5, Issue 3.
10. Garrido, P., Valgaerts, L., Sarmadi, H., Steiner, I., Varanasi, K., Pérez, P. and Theobalt, C. (2015), VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track. Computer Graphics Forum, 34: 193-204. doi:10.1111/cgf.12552
11. https://en.wikipedia.org/wiki/Speech_organ
12. https://en.wikipedia.org/wiki/International_Phonetic_Alphabet

AUTHORS PROFILE

Suriyah M., is a post graduate in Computer Science and Engineering with 4 years' experience in teaching graduate engineering students. Now she is a research writer with Karky Research Foundation.

Aarthy Anandan, is a post graduate in Computer Applications. She works as Software Developer at Karky Research Foundation.

Madhan Karky Vairamuthu, is a Ph.D in Computer Science. He is the Founder and Research Head of Karky Research Foundation, a non-profit educational research organization which focuses on language computing and language literacy.