**United International Journal of Multidisciplinary Research**
ISSN: 3048-6726 (UIJMR) Impact Factor: 6.934 (SJIF)
An International Peer-Reviewed and Refereed Multidisciplinary Journal
www.uijmr.in Vol-3, Special Issue-II ,2026

# Syllables, Semantics, and Servers: Computational Insights into Telugu Verse

**Sree Ganesh Thottempudi**
Germany

## Abstract

The rise of computerization has transformed the literature and publishing industries. However, there is still significant scope for research and development of computational tools that can help analyze and preserve classics, particularly in Indian languages. This paper discusses one such computational tool that can assist poets, publishers, and linguistic researchers in enhancing the quality of their work. The Chandassu framework, which focuses on metrical poetry, is well-known in Telugu literature but requires expertise in both the language and the computation of various rules. We describe how we have designed a machine-based verification and identification tool for Telugu metrical poetry (Chandassu), outlining the limitations of current research in this area and the need for integration with other linguistic tools and languages. Additionally, we will present case studies involving a poet, a publisher, and a researcher at the end of the paper.

## Introduction

*"Since my childhood I have been writing many metrical poems, most of them have errors. Everyone was trying to correct my mistakes, still I haven't understand the complex system"*

– Sri Sri a famous poet of 19th Century about Telugu metrical poetry [1]

Metrical poetry in Telugu is called as Chandassu. *Chandas Sastra* is a literature framework with set of rules to be followed to write a poem or prose. It was first used in *Sanskrit Veda's* and other ancient Sanskrit literature. *Chandas Sastra* in Telugu was derived from Sanskrit but it has its own, wide and complex in nature set of rules. The literature available in Telugu till the last century was mostly used *Chandassu* framework. Padyam was historically used with other art forms like drawing, multitasking and ciphers [2]

The literature work that followed chandassu framework in Telugu is referred as padyam. It usually consists of 4 lines but not restricted to four simply >=1. chandassu

of a padyam means the rules that are to be followed to write the padyam. *gana , yati , prAsa* (Rhythm)and *prAsa yati* are four features that form *chandassu*. *gana*

**gana**

*chandassu* classifies syllables into guru and laghu based on the time that takes to pronounce. Independent short syllables are called as laghu and long syllables are called as guru. The syllable next to the syllable that is formed by multiple consonants is treated as guru with in the same word or the words that are grouped together to form a single word. *laghu* and *guru* are represented by Symbols **|,U** respectively. Sequences of laghu and guru will form the gana. gana's need not to be formed with in the single word, can be formed across multiple words too. *gana's* are classified based on the *guru* , *laghu* sequences it contain.

- **Named *gana*:** These sequences are given with names for 1,2,3 Syllable sequences (*la, ga, va, ha, ya, ma, ta, ra , ha , bha , na , sa*). The sequences with more than 3 Syllables or un-named sequences are given with compound names Ex: ya-la. The *gana's* with compound names are called as Compound *gana's*

- ***gana* Groups:** Set of the named gana's or compound gana's are grouped into Indra, Surya and Chandra gana's.

- ***matra* gana:** These gana's classification is based on the total time takes to pronounce a sequence. Unit of time takes for a syllable of type laghu is considered as 1 and 2 for guru. Ex: 4 *matra gana* can be formed with 2U, 1 U + 2 | or 4 |.

**yati**

*yati* is similar syllable to the first syllable of the line at a fixed syllable position within the line or given gana . *yati* rules of chandassu defines which syllables are similar. There are more than 30 such rules and exceptions defined.

**prAsa (Rhythm)**

*prAsa* is usually second or last syllable of the poem line. Same or rhythmic syllable to be followed at these places. prAsa rules defines which syllables are rhythmic to each other.

**prAsa yati**

Incase poet couldn't able to apply *yati* then there is a flexible characteristic to create rhythmic syllable group along with the next syllable of *yati*. There are some more rules and conventions about the nature of previous syllable of *prAsa* and *yati*.

**Classification of *padyam's***

Telugu *chandassu* classifies padyam's in to four based on their characteristics[3].

1. *jAti:* Has *yati* and prAsa with fixed grouped *gana* sequence.
2. *upajAti:* Has only *yati* but not *prAsa* with fixed grouped *gana* sequence.
3. *vrutta*: Has defined no. and sequence of syllables with *yati* and *prAsa*
4. *matra chandassu:* Has *matra gana's* with optional *yati* or *prAsa*.

Telugu ancient literature mostly used first three categories whereas the modern poets adopted matra *chandassu*. *matra chandassu* can be found in many languages in forms.

**Machine based *Chandassu* Rules Computation**

The world of padyam, a traditional poetic form, features an endless variety of syllable sequences. Yet, only a few are recognized by the chandas Sastra, the ancient poetic science. For any number of syllables n ($n \geq 1$), the combinations possible are 2^n. The total number of sequences up to n syllables is given by the formula 2^(n+1) - 2. Chandas Sastra acknowledges only sequences of up to 26 syllables, each with its own name—like Gayatri for six syllables and AkRuti for twenty-two. Sequences beyond this limit fall into the udhdhooramala category.

The total number of sequences possible with up to 26 syllables is an impressive 134,217,728, calculated as 2^27 - 2. Crafting a padyam that fits a specific chandassu demands not only linguistic skill but also a solid understanding of poetic structures. Unfortunately, the use of chandassus in Telugu literature has diminished over the last century due to social and political changes.

Poets must master the intricate rules while focusing on creativity, a challenge that typically leads them to specialize in a limited number of patterns. Additionally,

publishers and digitizers must carefully check texts against these rules to ensure accuracy and prevent errors.

**Chandassu Matching Engine**

If a machine can compute these rules, poets can focus more on the creative aspects of literary work, while publishers can quickly identify digitization errors in literature. Such a machine should primarily be able to verify a given padyam against a specified chandassu or identify the chandassu of a given padyam along with any errors (i.e., rule violations). We have made efforts to build a first-generation linguistic machine tool called Chandam © [4], and we will present its components and matching algorithm.

The primary objective was to design a chandassu matching engine that compares the features of a given padyam against a designated chandassu. This engine consists of two main components: a Text Analyzer and a Feature Matcher. Each of these components will be explained below.

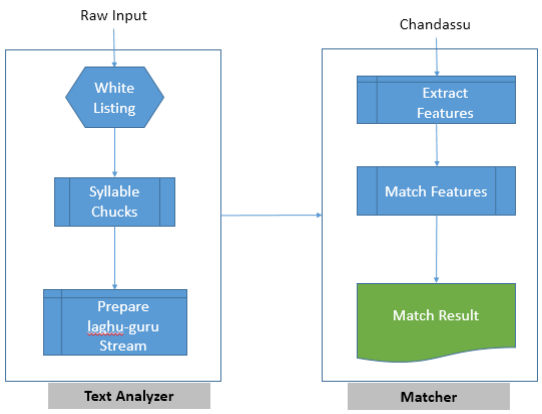



Fig1: **Matching Engine**

**Text Analyzer**

The text analyzer generates the *laghu-guru* stream of the given *padyam* text. To generate a *laghu-guru* stream the raw input text (*padyam* text) needs to be processed through below mentioned steps.

1. **Pre Text processing:** Even though Telugu has a Unicode block from 1998(?), many non-Unicode encodings like ISCII, Anu, Srilipi, RTS are widely in use.

Such encodings may be pre-processed and converted to Unicode using available convertors. This pre-processing of the text is not part of this matching engine.

2. **White Listing:** The raw input text will be cleaned first by removing foreign characters which doesn't belongs to targeted language not considered to attribute a *laghu* or *guru* value. The complete Unicode block doesn't participate in attributing syllables with corresponding laghu-guru value Ex: Numbers and fraction digits.

    **a.** Punctuations (Ex: White Space, Comma, Dot, Hyphen Exclamatory, Question mark and Quotes**)** are important in for reading and carries emotion.

    **b.** Line breaks are important to identify the lines.

    These special characters to be considered as a valid and may be presented with empty symbol value while preparing laghu, guru stream.

3. **Syllable Chunks:** Since *laghu-guru*s can be attributed to a syllable stream, it is very important to break words into syllables. The filtered or whitelisted text will be used to prepare syllable chunks. The text Splits into words. Each word will be identified with following standard procedure would be applied to break the words into syllables. Each word will processed to detect 3 types of syllable groups.

    a. **Consonant Syllable:** Contains one or more Consonants and ends with a Syllable modifiers or a *Halant* followed by Consonant. The following pattern may be used for detecting Consonant syllables.

    C+ [< H+ [<ZWNJ|ZWJ>] | {M} + [H]>] +[SM]
    Ex: The word కొత్త (read as *kotta* formed using following 5 characters *ka,O,ta,halant,ta*) has 2 Syllables కొ, త్త

    b. **Independent Syllable:** All other characters which doesn't match the Consonant Syllable may be treated as Independent syllable.

    Where

    - { } zero or more occurrences

- [ ] optional occurrence
- <|> "one of"
- ( ) one or two occurrences
- C consonant
- V independent vowel
- H *halant*
- ZWNJ zero width non-joiner
- ZWJ zero width joiner

Ex: The word గణన (read as *gaNana* formed using three characters *ga, Na, na* ) forms 3 syllables గ,ణ,న.

Each independent syllable and consonant syllable will be used to create a syllable stream for a word. These word-level syllable chunks will be combined to form a stream of syllables that corresponds to the complete padyam text. We have presented the syllable parsing mechanism used; however, there are many other syllable parsing techniques available.

4. **Building *Symbol* Stream:** Rules defined in *Chandassu* to be applied on the series syllable chucks to form the *laghu-guru* stream or Symbol Stream. For each syllable the natural symbol will be determined first and once the nature of the next syllable will be evaluated. If the nature of the next syllable influence, then the natural symbol will be overridden by new symbol and the process will be completed till the last syllable.

EX: Consider the Syllable stream కొ (*ko*), ట్ట (*tta*). The natural symbol for the కొ(*ko*) is | **(Laghu)** But the immediate syllable has a nature to influence the previous Syllable's symbol. Hence the symbol for కొ(*ko*) will become U **(guru).** The natural symbol for ట్ట (*tta*) is | **(Laghu).** The symbol stream or laghu-guru stream would be **U,I**

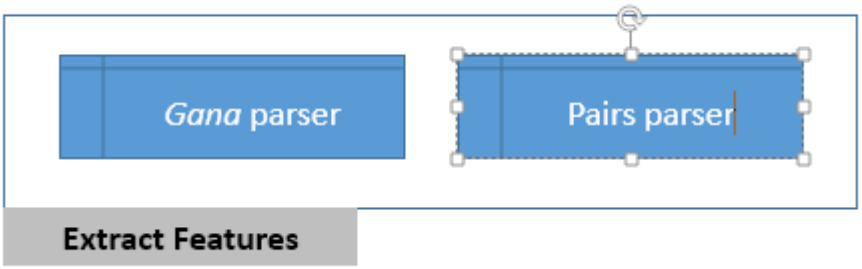
**United International Journal of Multidisciplinary Research**
ISSN: 3048-6726 (UIJMR) Impact Factor: 6.934 (SJIF)
An International Peer-Reviewed and Refereed Multidisciplinary Journal
www.uijmr.in Vol-3, Special Issue-II ,2026

**Feature Matching**

The main features that define a chandassu are gaNa, prAsa, yati, and prAsa-yati. The rules for evaluating these features are outlined in chandassu sastra. Each chandassu has distinct features, which need to be extracted from laghu-guru and syllable streams. The symbol stream is used to extract the gaNa feature, while the syllable stream is used to extract the other features. PrAsa, yati, and prAsa-yati are pairs of syllables located at specific positions.

Feature matching involves comparing the desired features against the actual features. This process extracts the features of the given chandassu from a unique identifier or the specified features first, and then uses the laghu-guru stream to determine the actual characteristics of the padyam.

1. **Feature Extraction:**




Feature2: Extraction

**a.** ***gana* parser:** gana is the main feature of the *chandassu*. Based on the characteristics of target *chandassu, symbol* stream will be parsed in to groups of named *gana* or grouped name or matra gana.

1. **Named *gaNa*:** A sets of symbol groups will be created from the symbol stream then the name for each set will be given as defined in *chandassu sastra.* The no. of symbols in group is uniform for a given *chandassu.*

Ex: Following Symbol stream **U||U||U||U||U||U||U||U** can be formed as 3 symbol groups as **U||, U ||, U ||, U||, U||, U ||, U||, U.** Name given

for UII is ***bha*** *and* Independent *U* is ***ga.*** The gana features actual would be ***bha-bha-bha- bha-bha-bha-bha-ga***

2. **Grouped *gana:*** For grouped gana's the max threshold of the next expected gana will be used to create the groups. Since grouped gana's can have different no. of symbol count. *Indra group* can be formed using 3 Symbols or 4 symbols. Surya group can have 2 Symbols or 3 symbols. *Chandra* group contains only 5 symbols.

Ex: Say for a given Chandassu which was defined as Surya, Surya, Surya, Indra , Indra with input symbol stream as |||U IU|U U|U|| . Symbol parsing will be done until for current group when the expected *gana* was *Indra* then it stops at the first occurrence of 3(max of 2 &3) Symbols found or *Indra* group found. The gana groups to form ||| -U| -U| -UU| -U|| i.e. *Surya-Surya-Surya-Indra-Indra.*

3. ***Matra gana:*** The formation of matra gana sequence is based the weight of the symbol. The weight of the Laghu is 1 and guru is 2. The *gana* feature will be extracted based on the next expected weight reached or exceeded.

Ex: Say a *chandassu's* gana feature defined as 4-4-4 with input symbol stream as | | | | U || | | | then symbols would be parsed until it reaches a weight of 4 then the first group can be found at end of 4th symbol. The final gana sequence would be | | | | -| U | -| | | |.

**b. Pair's parser:** Yati, prasa, and prasa-yati are pairs of syllables found within each line or padyam. In a given padyam line, a maximum of three pairs of these syllables may be identified based on the chandassu. The position of yati will be determined by the characteristics of the specified chandassu, and this position will be used to create a list of yati pairs. Likewise, prasa and prasa-yati pairs will also be extracted. Additionally, it is important to capture the previous and next syllables of each syllable pair, along with their corresponding symbols. If a syllable is part of a compound word, the pair should also include information about the words involved. These positions can be defined in either absolute or relative terms, such as

the 7th symbol position or the first position in the 3rd gana. In these cases, the gana feature will be utilized to extract these pairs.

**Ex:** Say for a given *chandassu* yati position is defined as 7 then a pair of syllables will be formed with syllables of 1st and 7th Symbol's.

2. **Match scoring:** Once the features are extracted these feature to be matched against the expected features. Each target feature will be compared with actual feature and will be assigned scores ranging from -1 to +2 as mentioned below.

   a. -1 → Key feature not found or mismatched.

   b. 0 → Feature not found or mismatched

   c. +1 → Feature found and matched.

   ***d.*** +2 → Key feature found and matched exactly.

   ***Percentage Match= (Sum of all features gained Score)*100 / ((2*No. of Key features) + No. of Normal Features);***

   Ex: *utpalamala* is a popular chandassu in Telugu. It is defined as bha-ra-na-bha-bha-ra-va as named gana sequence in each line with 4 lines. Yati at 10th place and prAsa at 2nd place of each line. The no. of features it can have are 7*3 gana features, 4*1 yati features, 4*1 prAsa features. If we have considered all features as normal then total features are (7*4+4*1+4*1) =36. If there is a mismatch yati pair at 4th line then Score = 35*100/36=**97.22%**

   In the above method I have considered only 2 types of features. A different scoring system may be adopted by weighting features in to 1-N scale. [N is the high valued feature].

3. **Results:** The match score and mismatch details may be published to in desired digital format like Text, HTML, PDF, Spared sheet etc. The standard rendering engines for each format may be used.

**Chandassu Identification**

The matching engine functions effectively when the chandassu of a given padyam is known. It serves the needs of writers and publishers under these circumstances. However, identifying the chandassu of an anonymous padyam can

be very challenging without an automatic chandassu identification mechanism. The process is time-consuming and labor-intensive. Although there are infinitely possible chandassu patterns, only a few have been recognized and utilized by ancient poets. Since the matching of the chandassu is based on actual and expected features, the most immediate way to identify the chandassu of a given padyam is to compare it against all available chandassu options and select the one with the highest matching score.

We have outlined a mechanism for building a Chandassu Identification System by reusing components from the matching engine. Key steps include:

**1. Eliminating Redundant Steps:**

The matching score obtained from the matching engine is crucial for determining the probable chandassu. The scores for each chandassu will help identify the closest match. Since text analysis can be performed independently of the target chandassu, the results of text analysis—specifically, the syllable and symbol streams—will be cached or reused for feature extraction and matching. This caching process reduces both processing time and resource consumption.

**2. Determining Eligible Candidates (Chandassus):**

Not all available candidates can be considered for matching, as only a subset may qualify. For instance, if the syllable stream has 20 syllables per line, it will not qualify for matching with a chandassu that has a minimum of 7 and a maximum of 12 syllables. To filter eligible candidates, we will use the number of syllables per line, as ganas are the primary feature contributing to the various characteristics of a chandassu.

a. **Range of Symbol count:** For all candidates no. of symbols per line may not be uniform. So the range of symbols count may be used to filter the eligible candidates. We determine the range of syllables with in the *padyam.* Ex: Say symbol count is 7, 12, 8, 15 in each line then range of symbol count in this case would be 7-15.i.e.

**Range =** Minimum no. of syllable count in any line to Maximum no. of syllable count in any line

b. **Extended Range of Symbol count:** Once the range of the symbol count is found we filter all candidates whose symbol count range falls under the mentioned range. The above range would filter the correct set of candidates if the given padyam text has no errors in any line. This range is un-reliable when some

**United International Journal of Multidisciplinary Research**
ISSN: 3048-6726 (UIJMR) Impact Factor: 6.934 (SJIF)
An International Peer-Reviewed and Refereed Multidisciplinary Journal
www.uijmr.in Vol-3, Special Issue-II ,2026

syllables were missed or added during digitalization. If we consider digitization errors could cause variance of ±**X%** of syllables then

Extended Range for **X1** to **X2** with threshold **t** is
**Floor(X1*((100-t)/100))** to **Ceil(X1*((100+t)/100))**

Ex: The extend range for the found range 7-15 with 5% of digitization errors is floor (7*0.95) to ceil (15*1.05)= 6 to 16.

c. **Eligible Candidates:** Extended symbol count range to be used for filtering of eligible candidates. We will pick all candidates whose expected range falls under the actual extended symbol count range.

2. **Probable Candidates:** After determining the eligible candidates we use the matching engine to find the matching score for each candidate. The candidate with highest match score may be treated as Identified "Chandassu" or closed match. By sorting on Match score, probable matches will be available for publishing results.

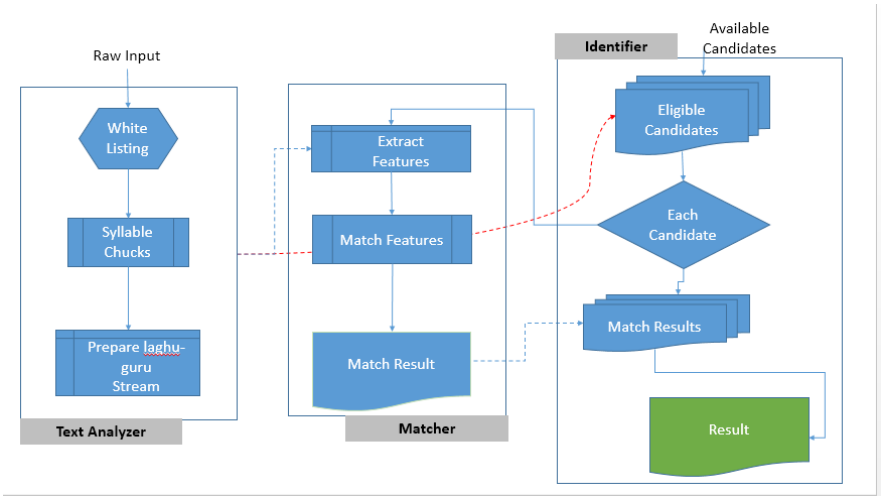



Fig3: Identification **Engine**

## Case Studies

1. **Telugu *Bhagavatam***: *The Bhagavatam is a sacred Hindu epic written by Vyasa between 500 BC and 1000 CE in Sanskrit. It was translated into Telugu*

*by Potana between 1400 CE and 1500 CE, using palm tree leaves for writing. The first printed version on paper was published in 1880. The Telugu Bhagavatam contains 10,061 padyams, with a total word count of approximately 900,000, including redundant words. In 2013, an organization named "Telugu Bhagavatam" (www.telugubhaghavatam.org) created a digital version of this work. This tool was used to identify errors in the digitization process, revealing that 26% of the padyams were reported to be defective. Subsequently, these results were compared with the original text, and the distribution of mismatches or defects is detailed as follows:*

**a.** **40%** of defects were made human errors like spelling mistakes or misplaced punctuations.

Ex: The syllable ఏ was used instead of ఎ in of the *padyams*. The same spelling mistake was a result of the publishing error in the printed version which was found in other publications too.

**b.** **30%** of defects were second kind of human errors. Misplaced "space" or ignored "space" resulted in Compound words to be treated as independent words or vice versa which had altered the features of *Chandassu.*

c. **30%** of defects reported were found to be falsely reported by the tool. It was due to the tool's limitations.

2. The total time taken to compute the complete Telugu Bhaghavatam is 18 minutes using a single process thread on an Intel® Core™ i5 desktop with 2GB RAM. This is significantly less compared to manual proofreading, which takes approximately 60 man-days with a confidence level of 80%. Regarding Telugu Wikipedia, a Satakam typically consists of at least 100 padyams (poetic lines). Many Satakams have been written by ancient poets, and digital versions of some of the more popular ones are available on Wikipedia. This tool was utilized to assess the quality of the digitization process. The percentage of errors reported for each Satakam ranged from 25% to 45%. In this case study, 10 Satakams, totaling 1,120 padyams, were analyzed.

3. **Poets**: Group of poets with primary or intermediate skills at writing *padyam* credited the tool for that it enhanced the quality of their work and now they could able to focus more on creative part rather than computation. Poets who are mastered in few patterns mentioned they found an easiest way to experiment and practice with the other chandassu's that are defined.

4. **Research:** With the help of this tool an analysis is being made to understand the patterns of syllable sequences and it's relation with music notes being carried on literature of *Annamaya* a 600 Year old musician and poet who written more than 30,000 Songs. This research also investigating to identify the influence of *chandassu* on his work.

**Limitations**

1. The compound words are very common in Telugu literature. Some specific word combinations influences the primary features of a *chandassu* like nature of syllables (laghu or *guru*), *yati* and *prAsa*. Development of Linguistic tools to identify the roots of a compound word. Results would be compromised till the time.

   Ex: Consider this line **ఆటగా ఛందాలన్నల్లించి, యలరించి** where **ఛందాలన్నల్లించి = ఛందాలను+ అల్లించి** is a compound word form with *yati* pair between **ఆ** and **అ** is valid whereas **ఆ** and **న** is not a valid pair

2. Some rules are not standardized or ambiguous among poets too. Machine should allow users to resolve such ambiguity. EX: Santa prAsa is popular in a regions influenced by Kannada language but it is not considered as standard rule Telugu *chandas sastra*.

3. Historically rules are modified with time & region. Rule sets may be formed to increase the quality of the results.

**Scope for further research and development of *Chandassu* Related Tools**

1. **Resolving the Lines:** Ancient documentation style used to write lines of Padyam in one line with separators, sometimes with no marking of *Chandassu* used. Resolving of such works and determining the *Chandassu* by automated mechanism could help the linguists in

understanding the evaluation and history of *Chandassu* in a faster and efficient manner.

2. **Discovering the Art Forms:** Ancient poets combined the padyam with other art forms like drawing. The connecting syllables could form a visual picture like series of dots on a canvas. The repeated Syllables will be treated as intersecting points. Some of the poets are mastering in such art and literature works. If a machine can help poets who are practicing such art forms it will definitely enhances the skills.

3. **Solving Ciphers:** It is another art form where by ignoring some specific syllables or positions could lead to different meaning by satisfying two or multiple Chandassu rules at a time. Such works are difficult to solve. Development of automated cipher solvers could reveal the mysteries and secrets of that age.

4. **Extending support to Other Languages:** Chandassu is also widely popular in other Indian Languages with some differences in their fundamental rules and features.

**References**

1. *SriSri kavitvaM by Miriyala Rama Krishana, Durga Press,Kakinada published in the year1980.*
2. *vaaDuka bhaashalO telugu kavitaa vikaasaM, http://eemaata.com/ by Bejjala Krishana Mohana Rao published in July 2008.*
3. *Kavijanaashrayamu-Chandashastram by Jayanthi Ramaiah published in the year 1932.*
4. chandaM : Implementation of Machine enabled Telugu Chandassu Verifier and Identifier http://chandam.apphb.com
5. *CaMd@Hpada kOSaM* by Kovela Sampathkumaracharya pubished in theYear 1977
6. *sulakshaNa saaramu* by Lingamagunta TimmaKavi published in the year 1965 (Original text 16th Century)
7. *anaMtuni CaMdassu* by Anamtudu, Vaavilla RamaSwamy Publishers in the Year 1921 (Original text 16th Centrury)

8. *appakavIyaM by* Appakavi , Vaavilla RamaSwamy Publishers in the year 1972 (Original text 15th Century)
9. Little Mater's Telugu *Vyakaranamu* by Sri M.Vishwanadha Raju,Nava Ratna Book Center published in the year 1998
10. *prauda VyakaraNa ghaMTaa padhamu* by Vemparala Suryanarayana Sastry,Visalandhra Publishers in the year 1999.
11. Telugu Dictionaries and Articles on chandassu http://www.andhrabharati.com
12. Articles of B.Mohana Rao on Eemata.com http://eemaata.com/em/authors?aa=జొజ్జాల కృష్ణ మోహన రావు
13. Data Sources: Telugu Wikipedia [ http://te.wikipedi.org ] and TelugBhaghavatam [ http://telugubhaghavatam.org ]
14. Libraries: Digital Library of India http://dli.gov.in and http://www.archive.org
15. Typography *http://www.microsoft.com/typography*