International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
ISSN: 2150-7988 Vol.2 (2010), pp.179-186
http://www.mirlabs.org/ijcisim

Unsupervised Stemmer to Improve Rule Based Morph Analyzer

KVN Sunitha, N.Kalyani
CSE Dept., G.Narayanamma Institute of Technology & Science
k.v.n.sunitha@gmail.com, narakalyani3@gmail.com

Abstract

Telugu is an Indian language spoken by more than 50 million people in the country. Language is very rich in literature, and it requires advancements in computational approaches. Applications like machine translation, speech recognition, speech synthesis and information retrieval need a powerful morphological generator to give morphological forms of nouns and verbs. The existing Telugu morphological analyzer (TMA) is rule based, the performance of it is further improved by our Novel approach which provides an Unsupervised Stemmer that gives information about possible decompositions of the word inflected by many morphemes. Using these possible decompositions the root word could be extracted for those words which were initially not recognized by rule based morphological analyzer. The experiment is conducted on CII Telugu corpus and the improvement in the performance is checked by the rule based morphological analyzer developed by LTRC group. In this present work we present an unsupervised stemmer for improving the performance of Telugu rule based morph analyzer. The main advantage is, increase in performance of rule based from 77% to 84.2% for words which are in hundreds. It can still be improved if the corpus is increased.

1. Introduction

Morphological analysis is an integral part of larger language processing projects such as Speech Processing, Information Extraction and Machine Translation. Tasks include transliteration of Telugu text, identifying syllables, separation of stems and affixes (prefixes, suffixes, infixes, crucifixes) and the identification of inflectional and derivational processes. These processes may be productive (i.e. apply to new words entering a language) and may also be combined (especially in agglutinative languages like Turkish) making an enumeration of morphological forms unfeasible as described in (Jurafsky et al., 2000)[1].

Unsupervised approaches to MA are important for less studied (and corpus-poor) languages, where we have small or no machine-readable dictionaries and tools. Ideally, an unsupervised morphological analyzer (UMA) would learn how to analyze a language just by looking at a large text in that language, without any additional resources, not even mentioning an expert or speaker of the language. The advantages of unsupervised approach for morphological analysis have been stressed by (Hammarstrom et al., 2006) [2]. and (Goldsmith et al.,2001) [3]. These advantages include elegance, economy, time and money, no additional resources, employment of same technology for new languages, appropriateness for languages with fewer resources (Hammarstrom et al.,2006) [2].

There is a body of related work that grows faster and faster as briefed in (Déjean 1998) [4] first induces a list of 100 most frequent morphemes and then uses those morphemes for word segmentation. His approach is thus not fully unsupervised. (Keshava et al., 2006) [5] combine the ideas of (Déjean 1998) [4]. On the Morpho Challenge 2005 datasets, they achieved the best result for English, but they did remarkably worse for Finnish and Turkish. Other UMA learning algorithms exploit the Minimum Description Length (MDL) principle(Mathias Creutz et al.,2002) [6,7]. Specifically, EM is used to iteratively segment a list of words using some predefined heuristics until the length of the morphological grammar converges to a minimum. (Brent et al. 1995) were the first to introduce an information theoretic notion of compression to represent the MDL framework. (Goldsmith 2001) [3] also used an MDL-based approach but applied a new compression system with different measuring of the length of the grammar. (Creutz 2003)[7] uses probabilistic distribution of morpheme length and frequency to rank induced morphemes. This method outperforms Goldsmith approach for Finnish but gets worse results for English. Given a low coverage morph and a corpus of raw text, our approach presents a simple algorithm for unsupervised stemmer for morphological analysis for inflectionally rich languages like Telugu, It assumes no particular theoretical model of morph, but can be used for any language.

Various approaches for unsupervised extraction of stems and suffixes have been reported for English, Assamies (Utpal Sharma 2006) [8], Dutch, German and Finnish among others. For the best of our knowledge, this is the first time such approach has been employed on extracting Telugu stems and suffixes. Broadly, our work is carried in three steps a) Processing the Telugu text corpus into syllable units. b) Developing an unsupervised stemmer which would give possible stems and suffixes for the given word and preparing new words list not given in the word list. c) Processing the given Telugu words using Telugu rule based Morph analyzer developed by LTRC group (IIIT Hyderabad and HCU). For the words that are unrecognized by

TMA, extra information provided by unsupervised stemmer and reprocessed to identify the stem components.

Our methodology for extracting valid stems and suffixes from a given text corpus is similar to Technique used for preparing new words based on probabilistic models as in (Jhon Chenet al., 2001)[9]. Apart from the assumptions given by Hammarstrom our methodology further assumes words constitute only of stems and suffixes, maximum length of a suffix is 8 and word can consist of only a single suffix. The experiments were conducted on CII Telugu corpus and the corresponding results are briefed in next section. Similar test is conducted on Hindi corpus. Last section consist of a conclusion and future scope.

2. Proposed system

The details of the complete system are shown in the following Figure1.

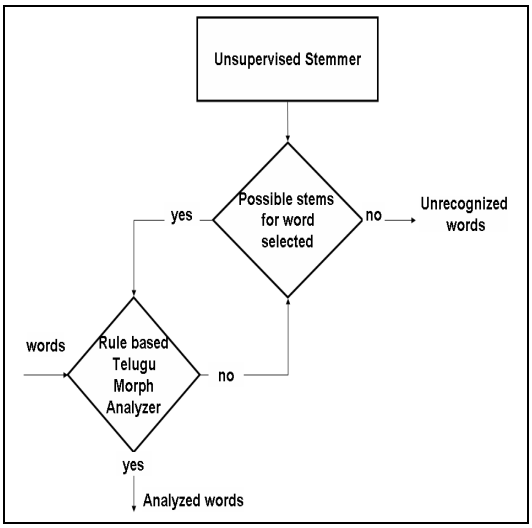


Figure 1. Proposed system

For giving the possible decompositions of the given set of words from an un-annotated text corpus we perform a surface level analysis of the corpus. An un-annotated text corpus presents two kinds of information about the language– first, the lexical space of the language, i.e., of the infinite possible letter sequences, the ones that form valid words in the language, and second, the morphological phenomena, i.e., the noticeable similarity in the structure of groups of words. In case of inflectional languages like Telugu, Assamese, Hindi, etc. the predominant morphological phenomenon is suffixation. We model the morphology acquired through analysis of the input training corpus, in the form of a collection of suffixes and the criteria for identifying the presence of these suffixes in different words. The knowledge given by Unsupervised Stemmer (US) about the words is used in improving performance of TMA. The morphological model and the US can be subsequently used for morphological analysis of words in texts.

Our first task is to process the Telugu text and identify the underlying suffixes in the language. Suppose, S_C is the set of suffixes identified by the computational process, and S_L is the set of suffixes that are actually there in the language. The ideal goal of the morphology acquisition process is to have S_C be the same as S_L. However, due to the constraints on the available evidence and the methods applied, S_L is usually not the same as S_C. Letter strings that are not really suffixes are identified as ones, while several valid suffixes are left unidentified. A morphology acquisition method is useful only if the S_C obtained is a close approximation of an underlying S_L. Similar issues arises in the next task of our approach. The next task is to build a lexicon from same training corpus. We use the suffixes acquired to decompose the words in the corpus. Simply, looking for matching of the suffixes at the end portion of words leads to a **large number of invalid decompositions.**

Methods discussed in (John Goldsmith 2001 and Eric Gaussier 99)[11] are representative of reported approaches to tackle the problem. In our approach, we apply heuristics based on statistics as well as other language specific and script specific aspects. We find that in case of Telugu, and possibly in other languages with similar features, our approach produces better results. There are approaches proposed before by Gaussier for identification of suffixes in (Eric Gaussier 1999)[11]. to acquire the suffixes used in a raw text corpus. In this method, a pair of decompositions using a common base is obtained for words W_i and W_j form the input corpus such that both are formed with common base or stem $\beta 1$ and suffixes α_1 and α_2 where β_1 is an accepted word in the corpus. The morphological extensions, α_1 and α_2, together referred to as a pseudo-suffix pair, are accepted as valid, if for some words W_k and W_l in the input, can be decomposed with valid stem β_2 and suffixes α_1 and α_2. Unsupervised morphology acquisition methods proposed by (Matthew G Snover et al., 2002) [11] are based on probabilistic models. Goldsmith (John Goldsmith 2001) proposed an approach which is based on Minimum Description Length (MDL) which is used in Linguistica tool.

3. Processing Telugu corpus

Telugu is one of the major Scheduled languages of India. It has the second largest number of speakers mainly concentrated in South India. It is the official language of Andhra Pradesh and second widely spoken language in Tamilnadu, Karnataka. There are number of Telugu language speakers have migrated to Mauritius, South Africa, and recently to USA, UK, and Australia. Telugu is often referred as "Italian of the East".

The Primary units of Telugu alphabet are syllables, therefore it should be rightly called a syllabic language. There is good correspondence in the written and spoken form of the south Indian languages. Any analysis done on written form would closely relate to spoken form of the language.

The Telugu alphabet can be viewed as consisting of more commonly used inventory, a common core, and an overall pattern comprising all those symbols that are used in all domains. The overall pattern consists of 60 symbols, of which 16 are vowels, 3 vowel modifiers, and 41 consonants.

3.1 Conversion of Telugu text to WX notation.

Since Indian languages are syllable-timed languages, syllable is considered as the basic unit in this work and analysis is performed to identify the words with syllables with high frequency and words with varying coverage of syllables.

3.1.1 Telugu to English letter translation

The WX notation of thirteen vowel signs అ,ఆ,ఇ,ఈ,ఉ,ఊ,ఋ,ఎ,ఏ,ఐ,ఒ,ఓ,ఔ is a,A,i,I,u, U,q,eV,e,E,oV,o,O occur as stand alone characters and In UNICODE Standard 3.0., each of these is assigned a hexadecimal code point 0C00-0C7F. When a vowel occurs immediately after a consonant it is represented as a dependent or secondary sign called, guNiMtaM gurtulu. The Telugu alphabet is a syllabic language in which the primary consonant always has an inherent vowel [a] / ◌ /. When a consonant is attached with another vowel other than [a] / ◌ / then secondary vowel sign is attached to the consonant after removing the inherent vowel /a/. There are exceptions where the primary vowel may be considered as secondary.

There are 41 consonants in the common core inventory. In Unicode Standard 3.0 they begin with 0C15 to 0C39 and 0C1A to 0C2F. The character set for consonants in Telugu is complex and peculiar in their function. These character signs have three or more than three distinct shapes depending on their occurrence.

- Base consonants or Primaries, when they are used as stand alone characters.
- Pure consonant or hanger, when used with a vowel other than the inherent vowel /a/
- Ottulu or Secondary consonant, when used as a constituent of a conjunct

The basic character set for consonants are called as primaries or stand alone characters as they occur in the alphabet. Each of which has an inherent vowel /a/ which often is explicitly indicated by sign / ◌ /. This graphic sign indicating the vowel /a/ is normally deleted and replaced with another explicit mark for a different vowel.

List of pure consonants carrying explicit secondary vowel /a/ sign and its corresponding WX notation are

క-ka గ-ga ఖ-Ka ఘ-G ఙ-fa , చ-ca ఛ-Ca జ-ja ఝ-Ja ఞ-Fa , ట-ta ఠ-Ta డ-da ఢ-Da ణ-Na , త-wa థ-Wa ద-xa ధ-Xa న-na , ప-pa ఫ-Pa బ-ba భ-Ba మ-ma య-ya ర-ra ల-la వ-va శ-Sa ష-Ra స-sa హ-ha ళ-lYa ఱ-rY

The Telugu text in Unicode format is converted to WX notation. The conversion is done character by character using Unicode value of the character. If the Unicode of the character is between 0C15 and 0C39 (క to హ), representation corresponding to Pure consonant is retrieved from WX table. If the Unicode of the character is between 0C3E and 0C4C (ా to ౌ), the last letter from Pure consonant is removed and secondary vowel representation is added. If Unicode of the character is 0C4D which correspond to stress mark ◌్, the last letter from the WX notation is removed indicating that the next occurrence of character is secondary consonant.

3.1.2 Algorithm

The algorithm for conversion is given below where englishtext is initialized to null.

```
string englishtext=null
read the contents and convert into character array
for each character till end of the file do
  if Unicode of the letter is between 0C15 and 0C60
      retrieve the corresponding English character for
      the Unicode, add to englishtext and increment i
      by 1
  else if Unicode of the letter is between 0C3E and
    0C4C
      remove the last letter from the englishtext,
      retrieve the corresponding English character for
      the Unicode, add to englishtext and increment i
      by 1
  else if Unicode of the letter is 0C4D
      remove last letter from the englishtext
  else
      copy the character into English text and
      increment i by 1
end for
Store in temp file for Syllabification.
End
```

The following Table:1 shows the output obtained for the input in Telugu text in UNICODE

TABLE I. OUTPUT FOR ALGORITHM 3.1.2

S. No	Input	Output of Algoritm 1
1.	కంపెనీకంటె	kaMpeVnIkaMteV
2.	ఖర్చుకంటె	KarcukaMteV
3.	లాభాలకు	lABAlaku

3.2 Syllabification

The scripts of Indian languages have originated from the ancient Brahmi script. The basic speech sounds units and basic written form has one to one correspondence. An important feature of Indian language scripts is their phonetic nature. The characters are the orthographic representation of speech sounds. A character in Indian language scripts is close to syllable and can be typically of the following form: C, V, CV, CCV and CVC, where C is a consonant and V is a vowel. There are about 35 consonants and about 15 vowels in Indian languages. The rules required to map the letters to sounds of Indian languages are almost straight forward. All Indian language scripts have common phonetic base.

The majority of the speech recognition systems in existence today use an observation space based on a temporal sequence of frames containing short-term spectral information. While these systems have been successful [10, 12], they rely on the incorrect assumption of statistical conditional independence between frames. These systems ignore the segment-level correlations that exist in the speech signal. The high-energy regions in the Short Term Energy function correspond to the syllable nuclei while the valleys at both ends of the nuclei determine the syllable boundaries.

The text segmentation is based on the linguistic rules derived from the language. Any syllable based language can be syllabified using these generic rules. To make the text segments exactly equivalent to the speech units.

The syllable can be defined as a vowel nucleus supported by consonants on either side, It can be generalized as a C*VC* unit where C is a consonant and V is a vowel. The linguistic rules to extract the syllables segments from a text are generated from spoken Telugu. These rules can be generalized to any syllable centric language. The text is preprocessed to remove any punctuation. The following algorithm divides the word into syllable like units.

3.2.1 Algorithm

- Read from the file which has text in WX notation.
- Label the characters as consonants and vowels using the following rules
  - Any consonant except(y, H, M) followed by y is a single consonant, label it as C
  - Any consonant except (y, r, l, lY, lYY) followed by r is taken as single consonant
  - Consonants like(k, c, t, w, p, g, j, d, x, b, m, R, S, s) followed by l is taken as single consonant.
  - Consonant like (k, c, t, w, p, g, j, d, x, b, R, S, s, r) followed by v is taken as a single consonant.
  - Label the remaining as Vowel (V) or Consonant(C) depending on the set to which it belongs.
  - Store the attribute of the word in terms of (C*VC*)* in temp2 file.
- For each word in the corpus get its label attribute from temp2 file.
  - If the first character is a C then the associate it to the nearest Vowel on the right.
  - If the last character is a C then associate it to the nearest Vowel on the left.
  - If sequences correspond to VV then break is as V-V.
  - Else If sequence correspond to VCV then break it as V-CV.
  - Else If sequence correspond to VCCV then break it as VC-CV.
  - Else If sequence correspond to VCCCV then break it as VC-CCV.
  - The strings separated by – are identified as syllable units.
- Repeat.
- Store the result in output file.

The following Table II: shows the output obtained for the input in Telugu text in UNICODE

TABLE II. OUTPUT FOR ALGORITHM 2

S. No	Output of Algoritm 1	Output of Algorithm 2
1.	kaMpeVnIkaMteV	kaM-peV-nI-kaM-teV
2.	KarcukaMteV	Kar-cu-kaM-teV
3.	lABAlaku	lA-BA-la-ku

4. Unsupervised Stemmer

To discover the set of suffixes in Telugu from a raw text corpus, our first step is somewhat similar to Gaussier's approach and is shown in Figure 2.

This approach comprises three major steps. First segment the words with word segmenter. We obtain all the decompositions in each of which a word, $\mathbf{W_1}$ of the corpus as two substrings $\mathbf{W_2}$, and $\boldsymbol{\alpha}$ such that

$\mathbf{W_1} = \mathbf{W_2} + \boldsymbol{\alpha}$

We refer to this exercise as initial decomposition. The idea is that α occurs as a suffix for other word in the corpus, it can be a valid suffix for some decomposition. Since word $\mathbf{W_1}$ has $\mathbf{W_2}$ as its leading portion, it is likely that $\mathbf{W_1}$ is derived from $\mathbf{W_2}$ with $\boldsymbol{\alpha}$ as a morphological extension.

Suppose, the set of words in the input corpus is w, we find the set of initial decompositions, D as

$D=\{[w = b + x] \mid w = bx$, and for some prefix b_1 and suffix x_1 bx_1 and $b_1x \in W\}$.

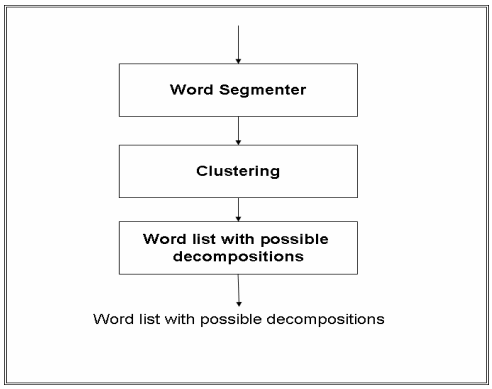


Figure-2 Flow chart for unsupervised stemmer

The set of morphological extensions obtained is

$E : \{x \mid [w = b + x] \in D\}$.

A few sample decompositions are shown below these are samples of telugu transcribed text.

ce-se = ce-se+@
(చేసే = చేసే + @)
ce-se-vA-ru = ce-se + vA-ru
(చేసేవారు = చేసే + వారు)
ce-si = ce-si + @
(చేసి = చేసి + @)
ce-si-na = ce-si + na
(చేసిన = చేసి + న)
pax-xu= pax-xu + @
(పద్దు = పద్దు + @)

4.1 Word segmentation Algorithm

The Figure 3 shows the flow of word segmenter.

1. Read text corpus in WX notation and clean corpus by elimination headers and punctuation marks.
2. Create unique word list (L) from Corpus.
3. Decompose each word to all possible suffixes and stems.
   Note: In this algorithm, maximum Suffix Length was taken as 8 and min. length of stem is 2. as the smallest word is formed with at least two letters.
4. For each stem, corresponding suffixes are identified and grouped ie. bag of suffixes is listed for each stem using the following procedure.

for each generated stem (S)

do

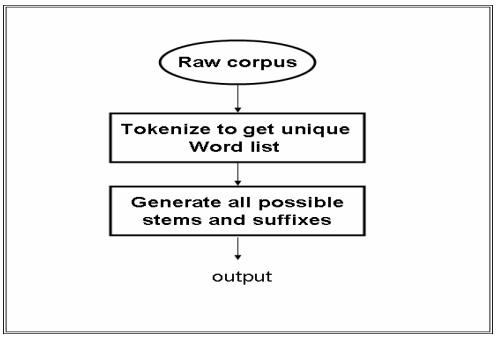
if (S) is not in the (stemList) append(S) into (stemList)

if (S) is in the (stemList)

if corresponding (suffix) is not in the (suffixesList) of particular stem (S)

append (suffix) in to the (suffixList) of particular stem (S)

next



లాభాల	లాభాలకు
లాభా+ల	లాభాల+కు
లా+భాల	లాభా+లకు
	లా+భాలకు

ఖర్చుకంటె	ఖర్చులకు
ఖర్చుకం+టె	ఖర్చుల+కు
ఖర్చు+కంటె	ఖర్చు+లకు
ఖ+ర్చుకంటె	ఖ+ర్చులకు

కంపెనీకంటె	కంపెనీలకు
కంపెనీకం+టె	కంపెనీల+కు
కంపెనీ+కంటె	కంపెనీ+లకు
కంపె+నీకంటె	కంపె+నీలకు
కం+పెనీకంటె	కం+పెనీలకు

Figure : 3 Word segmenter

4.2 Clustering Stems and Suffixes

Clustering algorithms divide data into meaningful or useful groups, called clusters, such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized. These discovered clusters can be used to explain the characteristics of the underlying data distribution and thus serve as the foundation for various data mining and analysis techniques.

CLUTO is a software package for clustering low and high dimensional datasets and for analyzing the characteristics of the various clusters. Our approach uses partition algorithm which is iterative procedure which divides the set into two groups and repeates until the number of clusters is equal to the specified number.

To use CLUTO tool the data should be converted to matrix form which is done by using doc2mat tool which converts the document data to matrix form.

4.3 Clustering Procedure

1. Drop all the stems that occur below a frequency count of 2 in the entire corpus.
2. Drop all the suffixes that occur below a frequency count of 2.
3. Generate vector matrix file for the pruned data.

Clustering is carried out by using CLUTO tool, with matrix file as input and the flow chart is shown in Figure 5.

By analyzing the output of Cluto tool, different clusters of stems are identified.

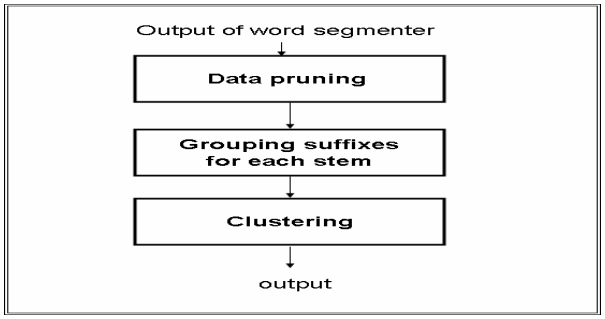


Figure : 4 Clustering procedure

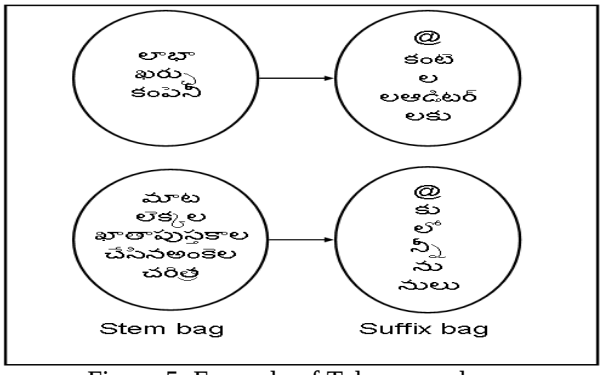


Figure 5: Example of Telugu words

4.4 Word list preparation with possible decomposition

New words are prepared by combining the stems belonging to a particular stem bag with the corresponding suffix bag. These words are treated as the training data for the next set of new samples. The Fig 6 gives an example of the word list prepared.

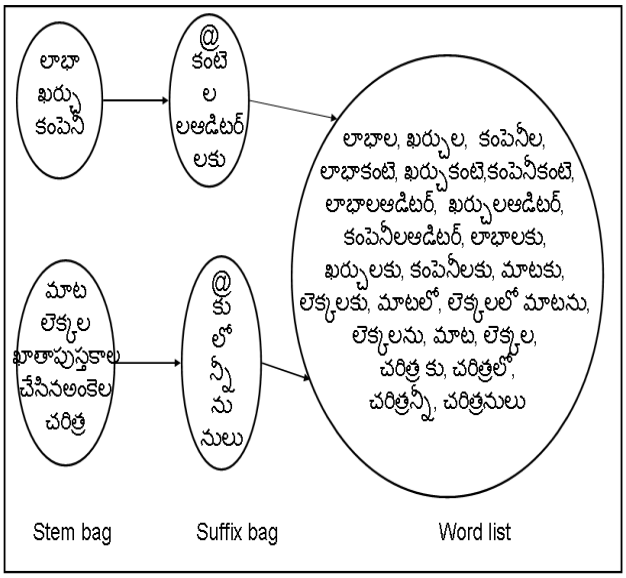


Figure : 6 Word list

5. Testing the results of US

The experiments were conducted on raw CII Telugu Corpus, collected from IIIT Hyderabad. The study was made on how the performance of rule based system developed by LTRC group, can be improved by unsupervised stemmer.

When the unique words list was given to the rule based Telugu analyzer there were some set of words that could not be recognized. For these set of words the relevant information is extracted from the out put generated by the unsupervised stemmer, in the form of possible decompositions. From this decomposition the stems are extracted and reanalyzed using the Telugu

morphological analyzer. Many words that were left unrecognized before are analyzed on giving the morph information of the partial word. By this the successive processing would be improved.

The following table gives the examples of words that are not recognized TMA. After extracting the stems from the unsupervised stemmer they are recognized as shown in following Table III.

Disadvantage is that it generates many words that are not valid. This can be controlled by having the control on the word generation. The best method is while forming the new word selecting a stem from stem bag and a suffix from corresponding suffix bag choose only those stems and suffixes that have a frequency count that is more than a threshold value. The best threshold value to be chosen is 4 or 5.

Unrecognized word by TMA	**Recognized as follows after providing information from US**
అభిప్రాయాలన్ని (aBiprAyAlanni)	అభిప్రాయం (aBiprAyaM)
ఆడిటనవచ్చు (Aditanavaccu)	ఆడిట్ (Adit)
అధికారినివేరు (aXikArinIveru)	అధికారి (aXikAri)
గుప్తులకాలం (gupwulakAlaM)	గుప్తి (gupwi)
గుప్తులకాలంలో (gupwulakAlaMlo)	గుప్తి (gupwi)
కంపెనీల (kaMpeVnIla)	కంపెని (kaMpeVnI)
కంపెనీలన్నీ (kaMpeVnIlannI)	కంపెని (kaMpeVnI)
మయిన (mayina)	మయి (mayi)
పన్నులే (pannule)	పన్ను (pannu)
ప్రారంభించడమైంది (prAraMBiMcadamEMxi)	ప్రారంభించు_అడం (prAraMBiMcu_adaM)
ప్రయోజనకరమని (prayojanakaramani)	ప్రయోజనకరం (prayojanakaraM)
పుస్తుకాలను (puswukAlanu)	పుస్తు (puswu)
వారిచేతే (vAricewe)	వారు (vAru)
విధింపే (viXiMe)	విధి (viXi)

TABLE III. RECOGNIZED LIST OF WORDS USING UNSUPERWISED STEMMER

6. Conclusion

Morphological analysis is a very significant step of NLP for highly inflectional languages such as Telugu.

Morphology and syntax are two complementary parts of the structural aspects of natural language expression. Because of the structural nature of morphology, simple computational methods can serve as the initial steps for acquisition of morphology of a language and morphological analysis. We have been largely successful in providing extra information to Telugu rule based morphological analyzer using unsupervised stemmer.

Our work is particularly significant because apart from providing required information we could focus on the increasing the word list with limited given corpus. The Unsupervised stemmer provided by our method can be directly used for processing of different languages of similar behavior. This kind of approach also helps in building models for sub word units at morpheme level which would help in speech recognition and speech synthesis process.

Acknowledgments

We thank Prof. Rajeev Sangal director of IIIT Hyderabad Mr.Srinivas Bangalore of AT&T labs and Mr Sriram research scholar from the LTRC group for providing us this opportunity through the NLP Winter school and summer school organized at IIIT Hyderabad which was sponsored by TCS.

References:

[1] Daniel Jurafsky and Patrick Schone. 2000. Knowledgefree induction of morphology using latent semantic analysis. In *Proceedings of CoNLL-2000 and LLL- 2000*, pages 67–72, Lisbon.

[2] Hammarstr¨om, H.: A naive theory of morphology and an algorithm for extraction. In Wicentowski,R., Kondrak, G., eds.: SIGPHON 2006: Eighth Meeting of the Proceedings of the ACL Special Interest Group on Computational Phonology, June 2006, New York City, USA, Association for Computational Linguistics (06) 79–88.

[3] Goldsmith, J.A. (2001). Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, 27:2 pp. 153-198.

[4] D´ejean, H. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. *Workshop on Paradigms and Grounding in Natural Language Learning*. Adelaide, Australia. 295–299.

[5] S. Keshava and E. Pitler. 2006. A simpler, intuitive approach to morpheme induction. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 31–35, Venice, Italy.

[6] Brent, Michael R., Sreerama K. Murthy, and Andrew Lundberg. "Discovering Morphemic Suffixes: A Case Study in MDL Induction." *The Fifth International Workshop on Artificial Intelligence and Statistics*. Fort Lauderdale, Florida, 1995.

[7] Creutz,M. 2003. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the Association for Computations Languages* (ACL'03). Sapporo, Japan. 280–287.

[8] Utpal Sharma 2006. "Unsupervised Learning of Morphology of a Highly Inflectional Language" Ph. D thesis submitted to Department of Computer science and Information Technology, Tezpur University, Napaam – Assam India.

[9] John Goldsmith. Unsupervised learning of the morphology of a natural language. Computational Linguistics, 27(2):153–193, 2001.

[10] Eric Gaussier. Unsupervised learning of derivational morphology from inflectional lexicons. In ACL '99 Workshop Proceedings: Unsupervised Learning in Natural Language Processing, pages 24–30. ACL, 1999.

[11] Matthew G Snover, Gaja E Jarosz, and Michael R Brent. Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In Workshop on Morphological and Phonological Learning, ACL-2002, pages 11–20, Philadelphia, July 2002.