Chapter 5

LemmaQuest: A Lemmatizer

5.1 Introduction

In this chapter, the discussion and deliberation are based on a more effective and efficient model for lemmatization which has been named as LemmaQuest. The previous model - LemmaChase has been extended for higher functionality computations. The existing popular online lemmatizers like the Stanford LemmaProcessor, spaCy Lemmatizer, LemmaGen, MorphAdorner, etc. generate the correct lemmas for all singular-plural nouns and all verbal words existing in different tenses, but all these lemmatizers are not able to derive the correct lemma for any type of derived words; specially for nominalized derived words. The proposed lemmatizer – 'LemmaQuest' is designed and implemented to overcome these limitations.

The processes of LemmaChase are optimized by statistical computation to create separate groups for morphed words where each group will contain all related morphed words. Each group will map to a single lemma. Placing related morphed words into the proper group leads to avoiding error in lemma generation. In LemmaQuest the number of word processing steps has been drastically decreased by generating single common lemma for each group of allied morphed words. This automatically minimizes the overhead of WordNet dictionary look-up.

In this chapter, the LemmaQuest model has been described in detail with appropriate illustration and comparative study with other lemmatizers, morphological analyzers and with the previous LemmaChase model.

5.2 Understanding LemmaQuest

Stemming and lemmatization are the two most important pre-requisite tasks of most of the NLP, TM, IE, IR etc. related applications. A better performing Lemmatizer is essentially required to find the correct dictionary root word for its any allied morphed word present in a text for text simplification, key-term identification, text summarization, topic

identification, etc. for morphologically rich languages, where one root word might have many morphological variants due to agglutination or inflection. In an IE or IR system, lemmatization can provide support to improve overall retrieval recall since a query will be able to retrieve more relevant documents when variants in both query and documents are morphologically normalized.

Since the focus of the research work is based on designing and developing a lemmatizer, it becomes important to understand certain facts and concepts about an English 'word' and its semantical and syntactical structure. This has been discussed in detail in Chapter 1. As mentioned earlier, a root word is called a morpheme and the English language has eight inflectional morphemes. So, it is easy to identify lemmas from the inflected words.

In English grammar, nominalization is one type of word generation process through which a verb or an adjective (or another part of speech) is transformed into a noun. This is also called ***nouning***.

The existing popular lemmatizers do not find lemmas for single, double or even triple suffixation words. The double or triple suffixation words (applicability, imaginatively, regretfully, developmental) are not accurately processed by the LemmaChase model to generate their corresponding lemma. Many double, triple suffixation surface-words (deployable, acidifiers, academicianship, civilization, collectivization and commercialization) are also unavailable in the WordNet dictionary. So, in the proposed model, all these dictionary-unavailable words are initially processed and are reconstructed into new dictionary surface words as part of one of the steps. This model morphologically analyzes the composite derived word to generate intermediate derived word which leads to generate the final lemma (Beautifully → beautiful → beauty; customarily → customary → custom; academicals → academic → academy).

Some non-allied morphed words are erroneously grouped with related allied morphed words based on String-similarity calculation using this model (polish/police; Algebra/Algeria). This model identifies and separates out all those odd words from the group and generates lemma for each odd word. Only corpus-based stemming algorithm is able to distinguish morphologically disconnected string similar word-pair and then able to generate stem for all those words.

LemmaQuest is an effective model for all those texts in which the maximum number of allied morphed words co-exist (e.g. dictionary words). This model analyses the structure of derivational and nominalized English words at group level. It first creates distinct groups for all allied morphed words like singular-plural nouns, verbs in all tenses, and nominalized

words. These groups are created based on a combination of different statistical distance measures considering all possible pairs of input words. After that, lemmas are generated for each group to minimize the analysis of all allied morphed words individually, which are merged into a single lemma. The main objective of this proposed model is to extract the correct lemma for a set of a large number of input words in an optimized time, which can lead to a vast improvement in text simplification, keyword extraction, text summarization and other text mining related and NLP related applications.

5.3 Input DataSet

The text from DUC, Brown Corpus and word list from MorphoLexDatabase are mainly taken as input in LemmaQuest. The dataset has been described in detail in Chapter 4.

5.4 LemmaQuest Model

Fig. 5.1 depicts the diagram for showing the major steps of LemmaQuest. The working of each step of this model is explained elaborately in Algorithm section.

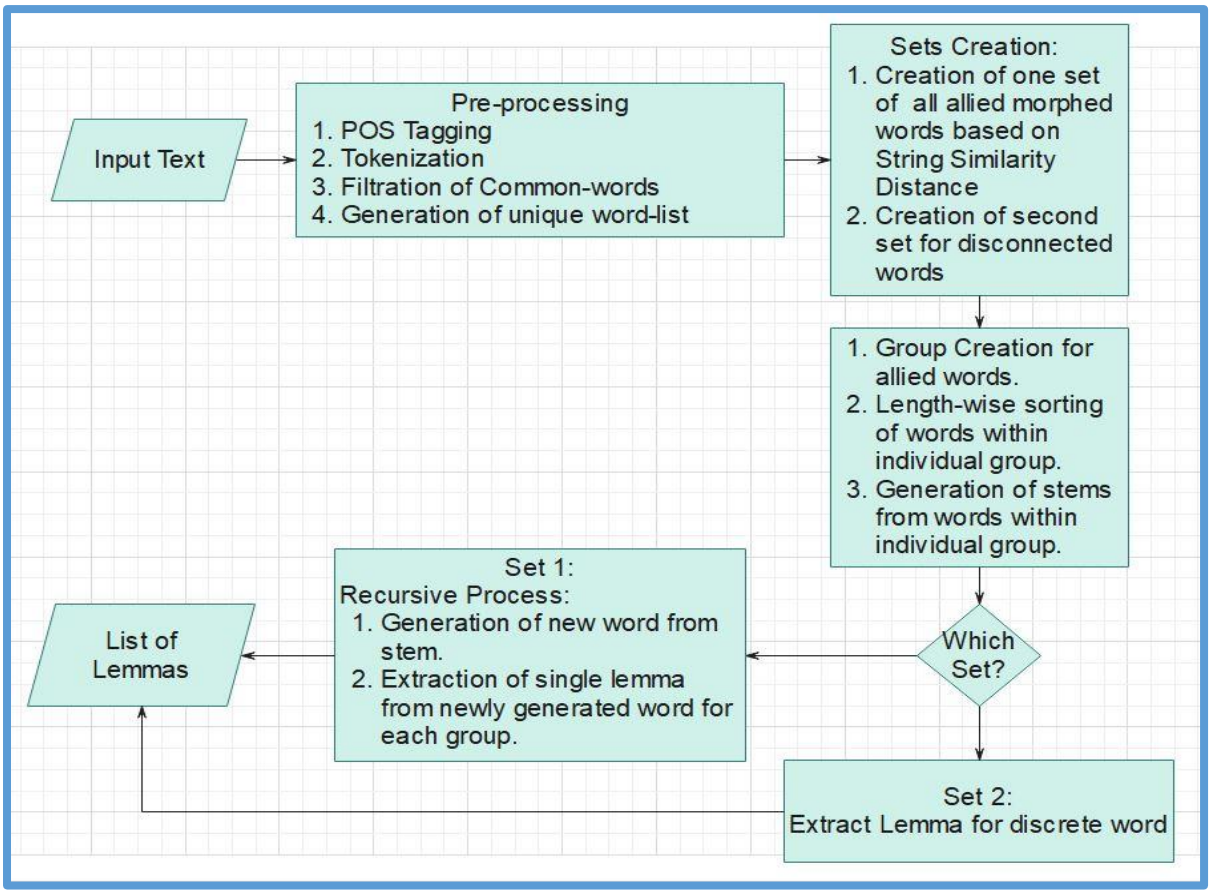

Figure 5.1 LemmaQuest Model

5.5 The LemmaQuest Algorithm

Before the algorithm is discussed, it is important to understand the standard preprocessing which is normally done on the text file for any text mining related applications. Steps of the preprocessing are as follows which are same as discussed in previous chapter.

Preprocessing of the File:

1. In this step, all unnecessary characters like punctuations, symbols will be removed.
2. Convert all words into lower case. All stop words are removed.
3. Tokenize the sentence into words.
4. Generate sorted unique word list.
5. Tag all unique words using Stanford POS tagger.

The LemmaQuest model implements the algorithm which is shown below.

5.6 Details of LemmaQuest: Lemmatizer

Input: List [w_s]={w_i/w_i_POS_tag, w_j/w_j_POS_tag…w_n/.. }:

w_i represents i^{th} word of the text. w_i_POS_tag represents the context (Part-of-speech) of w_i in the text.

Process: Each surface word (w_s/ w_i) to be lemmatized through invocation of function**.** **"$f_{LemmaQuest}$(List[w_s])".**

Output: S_L- List [lemmas]. S_L represents a compressive list of input-words' lemma;

S_L = {L_1, L_2, L_3.....}

Algorithm implemented in the LemmaQuest lemmatizer is depicted below.

1. Calculation of String Similarity measure in-between all probable pair of strings (words), available in the input text-file.
   1.1 Generate an output File which contains all probable word pairs with their distance value.
2. If value of string-distance (word1, word2) < δ (δ is a string similarity threshold value),

   Then Add all (word1, word2) pairs with their Distance value into a set **S_p.**

   Else

   Add all disconnected distinct words into another set S_p'**.**

   Repeat Step 2 until EOF of output text-file.
3. Find the collection of morphed words which are associated with w_i from set S_p.
4. Add all above mentioned word collection including w_i into dynamically created group G_i.

   Repeat Step 3, 4 for each record of new word pair [w_i, (w_j, w_{j+1}, $w_{j+2...}w_{k+1}$)] until EOF of set S_p.
5. Refinement and updating of each group G_i to another group G_i' .

   Pop up the shortest length morphed word on the top of the each group G_i'.

   Repeat Step 5 until the end of processing of all groups G_i.
6. Segment all words into stems for each group G_i' using LSV (Letter Successor Varieties) technique.
7. If the stem is not found as a complete dictionary word,

   Then construct a new word from the stem to apply word constructions' suffix/recoding rules.
8. Extraction of lemma for the new dictionary word or the stem.

   If the lemma and the new dictionary word are found identical or found only last indexed

   Character dissimilarity,

   Then that lemma is declined as the lemma for the new word.

   Else

   The lemma obtained from Step 8, is finalized as the lemma of the group.
9. Append the lemma into the Lemma list.

   Go to Step 14
10. Construction of a new dictionary word from the declined lemma.
11. Extraction of lemma for the new word.
12. If any valid suffix is still attached with new lemma

    Then, that lemma is parsed and reconstructed for generating new word.

    Repeat step 11 and 12 until valid suffix is not identified from the word.
13. Append the lemma into Lemma list.
14. Repeat step 6 until the end of processing of all groups G_i'.
15. Extraction of Lemma for each word of set S_p'.
16. Parsing of each word based on POS of the word.
17. Constructions of a new dictionary word to apply suffix/recoding rules from parsed word.
18. Extraction of lemma for this new word and append the lemma into Lemma list.
19. End

Algorithm of LemmaQuest

Process description: $f_{LemmaQuest}$(List[w_s])

Step: 1 -String Similarity Distance Calculation.

1.1. For each w_i, calculate string similarity distance with all w_j 's of input text file. Save word pair with their distance into a word-pair file.

If word-pair-similarity Distance [Sim(w_i, w_j)] < δ (δ is a string similarity threshold value), then build a set **S_p** from word-pair file.

Add all connected word pairs (w_i, w_j) into **S_p**.

S_p = { {w_i, POS(w_i), w_j, POS (w_j), Sim(w_i, w_j)}, {w_i, POS(w_i), w_{j+1}, POS(w_{j+1}), Sim(w_i, w_{j+1})},……}

Else If word-pair-similarity Distance [Sim (w_i',w_j')]=∞ (infinite) , then build a set S_p'.

Add all disconnected words (w_i',w_j') into the set S_p'.

1.2. Repeat Step 1 until the end of input text file.

Note: Word-pair-similarity Distance [Sim(w_i, w_j)] measure is formulated based on "YASS distance measure" and "Levenshtein distance measure". Levenshtein distance calculation is already described in Chapter 2.

In LemmaQuest, each below-mentioned formula of distance is calculated for each character mismatch instead of calculation of only 1st character mismatch as in YASS.

Agenda behind formulating these YASS distances was to give bonus point to long matching prefixes, and to penalize an early mismatch. The actual distances are obtained by multiplying the total penalty by a factor which is intended to reward a long prefix matching.

For given two strings w_i=[w_{i0} w_{i1} ... w_{in}] and w_j=[w_{j0} w_{j1} ..w_{jn}],

boolean function P_i (for penalty) as follows:

P_i=1, if there is a mismatch in the i^{th} position of w_i and w_j and P_i=0 for match.

If w_j and w_j are of unequal length, the shorter string is padded by null characters to make the string lengths equal (length of the strings: (n+1)).

Symbol 'm' indicates the position of the first mismatch character between w_i and w_j (i.e., w_{i0} = w_{j0}, w_{i1} = w_{j1}, $w_{i(m-1)}$=$w_{j(m-1)}$, but *w_{im} # w_{jm}). ["*" indication of mismatch]

$$D1(w_i, w_j) = \sum_{i=0}^{n} (\frac{1}{2^i})\, p_i \quad (1)$$

$$D2(w_i, w_j) = \frac{(1)}{m} \times \sum_{i=m,*wim\#wjm}^{n} \frac{1}{(2^{i-m})} \quad \text{if } m>0,\infty \text{ otherwise} \quad (2)$$

$$D3(w_i, w_j) = \frac{(n-m+1)}{m} \times \sum_{i=m,*wim\#wjm}^{n} \frac{1}{(2^{i-m})} \quad \text{if } m>0,\infty \text{ otherwise} \quad (3)$$

$$D4(w_i, w_j) = \frac{(n-m+1)}{n+1} \times \sum_{i=m,*wim\#wjm}^{n} \frac{1}{(2^{i-m})} \quad \text{if } m>0,\infty \text{ otherwise} \quad (4)$$

$$(Average\_DIST(w_i, w_j)) = \frac{(D1 + D2 + D3 + D4)}{4} \quad (5)$$

$$(Sim(w_i, w_j)) \equiv (Average\_DIST(w_i, w_j))$$

Step: 2- Group Generation.

2.1. For each w_i of S_p, find all w_i's connected morphed words.
Generate individual group for each w_i with its' allied morphed words dynamically.

2.2. Repeat Step 2.1 until the end of S_p set.

2.3. Display all groups of allied morphed words, derived from S_p set. G= $\{G_{1,..}G_m\}$: where m is the number of groups.

($G_i \subset S_p$ where 1>=i<=m). Here, ∀ w_s, m is considered as the number of groups ($G_1 \ldots G_m$) m<<n.

G_i ={{w_i, POS(w_i), flag}, {w_{i+1}, POS(w_{i+1}),.. }}

Step: 3- Group Refinement.

3.1. For each $G_{i,}$ generate G_i'.

3.1.1 Sort all words of G_i based on word length and then on alphabetical order in such a way that the shortest word will be popped on the top of each group G_i'.

3.2. Repeat 3.1 until end of processing of all groups G_i.

3.3. Display the top most morphed words (w_s_t) having the shortest suffix from each group G_i'. Generate the stem of the top-most word of each group.

$w\_s_t = (stem)_p + s_x$,

Step: 4- Word Segmentation and Generation of Lemma.

4.1. For each G_i', segment all w_s into $stem_p$ after applying "Word Segmentation by Letter Successor Varieties" (LSV) method (Hafer & Weiss)" on each word (w_s) of G_i'.

Pseudo Code of LSV is given below:

```
SegmentUsingSuccessor( w_s ){ s= w_s ;
for each substring s of each w_s { Calculate the successor count Sn;
 if found a local peak/plateau
        Save this position to an array of split points; }
 return array;}
```

4.2 If segmented $stem_p$ of the top-most word is found as a complete dictionary word,

Then this $stem_p$'s lemma is extracted;

Else,

Lemma of the top-most word (w_s_t) of G_i' is extracted using JWNL-WordNet APIs.

4.3. If extracted Lemma (from step 4.2) is found character-to-character identical with its input word ($stem_p$/ w_s_t) or one character dissimilarity with its input word in the last index-position,

Then, this Lemma is declined.

Else,

Go to step: 4.6.

[Step: 4.3 indicate that extracted lemma may not be the correct lemma for the input-word. e.g., incorrect lemma "application"/"employee" is extracted for input-word "applications"/"employees". The new word construction process $[employee]_{Noun} \rightarrow [employ]_{Verb}$ is still pending to do.]

4.4. **Non-allied word identification**:

If the words having similar POS and equal length in a single group, are identified,

then, all those words are marked as odd words in the group.

Odd words are individually processed to generate their lemma.

4.5. **Generation of new word** (Parsing & Recoding/morphological Analysis):

For any nominalized word and derived word (w_s_t/ $stem_p$), parsing of input-word is processed based on below-mentioned POS-class-wise suffix rules (Table 5.1). The $stem_p$ will be recoded to generate a new word. e.g., professional→ profession→ profess; professors → professor → profess.

Double and triple suffixed-derived words are also parsed step by step to generate intermediate derived words.

4.6. **Extraction of Lemma**:

The lemma (L_i) extracted for the newly constructed word is identified as input word for next iterative parsing process (Step 4.5).

For all words of the group G_i', L_i is extracted.

$L_i \equiv G_i'$ where G_i'= {w_s: w_s represents all allied morphed words}.

Append the final lemma into Lemma list.

Go to step 4.7.

4.7. Otherwise, extracted lemma (L_i) obtained from step 4.2 is finalized as the lemma for all words of G_i'.

Append the lemma into Lemma list.

4.8. Repeat step 4.1 to step 4.6 until all groups G_i' traversal is completed.

Step: 5- Generation of Lemma.

5.1. Generation of new word for S_p' set:

Each word w_s of S_p' is parsed based on POS-based derivative suffix and recoding rules to construct a new word.

5.2 Extraction of Lemma:

Lemma is extracted for the new word until bottom-most lemma is not retrieved from the input word.

Append the new lemma into Lemma list.

Step: 6- Generation of Lemma List.

6.1. Finally, ∀w_s (input-words), a set S_L is developed such that S_L = { L_1, L_2, L_3.....}.

S_L represents a list of lemma for all input-words (List [w_s]).

Note:-In the step 4.1 & step 5.1 of the algorithm, truncation process of the longest suffix and concatenation of the context-sensitive new suffix will be incorporated for generation of a new word from existing input surface-word.

Output: $S_L = f_{LemmaQuest}$(List[w_s]). $f_{LemmaQuest}$(List[w_s]) represents the function which takes list of words as a parameter and returns S_L(List of Lemmas). [Generation of single lemma from each group G_i' and a single lemma from each disconnected word of S_p'].

Note: The sample set of context-sensitive suffix list and their recoding rules are shown in below-mentioned Table 5.1. After parsing of input-words into stem and then reconstruction of the stem or input word into a new valid dictionary word based on below-mentioned set of rules, LemmaQuest is able to extract the correct lemma from the maximum number of English derivational surface words and nominalized words.

Table 5.1 Sample Set of Suffix Rules

	Suffix	**Input word**	**Lexical function within the item-and-process model**
Verb To Noun	-ation	don-ation, regul-ation, educ-ation	$[x]_V \rightarrow [[x]_V ation]_N$: [[donate]$_V$ ation]$_N$, [[educate]$_V$ ion]$_N$
	-er	teach-er, runn-er, writ-er, build-er, paint-er	$[x]_V \rightarrow [[x]_V er]_N$,: [[teach]$_V$ er]$_N$,[[run]$_V$ er]$_N$, [[build]$_V$ er]$_N$
Verb To Adjective	-ive	act-ive, decis-ive	$[x]_V \rightarrow [[x]_V ive]_{ADJ}$: [[act]$_V$ ive]$_{ADJ}$, [[decide]$_V$ ive]$_{ADJ}$
	-able	read-able, govern-able; manage-able	**$[x]_V \rightarrow [[x]_V able]_{ADJ}$:** [read]$_V$able]$_{ADJ}$, [govern]$_V$able]$_{ADJ}$
Verb To Verb	-er/-ify	chatt-er, patt-er, flutt-er, sign-ify	$[x]_V \rightarrow [[x]_V er]_V$: [[chat]$_V$ er]$_V$, [[flut]$_V$ er]$_V$, [[sign]$_V$ ify]$_V$
Noun To Adjective	-al	division-al, medicin-al, origin-al, univers-al	$[x]_N \rightarrow [[x]_N al]_{ADJ}$: [[medicine]$_N$al]$_{ADJ}$, [[origin]$_N$ al]$_{ADJ}$
	-ish	fool-ish, child-ish, self-ish	$[x]_N \rightarrow [[x]_N ish]_{ADJ}$: [[fool]$_N$ ish]$_{ADJ}$

5.7 Result and Discussion of LemmaQuest

This section discusses the output generated by LemmaQuest and the output is compared with that of existing popular lemmatizers and morphological analyzers. Further a comparison between LemmaChase and LemmaQuest has also been done.

Output of LemmaQuest: Maximum number of single, double and triple suffixation nominalized/derived words (Table 5.2) in the form of Noun, Adjective and Adverb are correctly handled by LemmaQuest to generate their corresponding lemma. LemmaQuest handles greater variety of nominalized and derived words more accurately in group level.

Table 5.2 Sample of Double suffixation Derived words and Lemmas

	Input Word	**Single-Suffix deleted intermediate word**	**Double-Suffix deleted intermediate word/Lemma**	**Triple Suffix deleted Lemma**
1.	academicianship	academician	academic	academy
2.	collectivization	collectivize	collect	
3.	beautifully	beautiful	beauty	
4.	collectivization	application	apply	
5.	commercialization	commercialize	commercial	commerce
6.	complaints	complaint	complain	
7.	complications	complication	complex	
8.	compliances	compliance	comply	
9.	civilization	civilize	civil	
10.	politically	political	politics	
11.	terrorization	terrorize	terror	
12.	acidifier	acidify	acidic	acid
13.	beautifully	beautiful	beauty	

Table 5.3 to Table 5.27 depict some sample input text files and their corresponding list of Lemma output generated by LemmaQuest.

Input 1: Sample input text of Brown corpus (w_s) is accepted as input-text file for generating lemmas.

Table 5.3 A. Sample of Input Text of Brown Corpus

The Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place. The jury further said in term-end presentments that the City Executive Committee, which had over-all charge of the election, deserves the praise and thanks of the City of Atlanta for the manner in which the election was conducted. The September-October term jury had been charged by Fulton Superior Court Judge Durwood Pye to investigate reports of possible irregularities in the hard-fought primary which was won by Mayor-nominate Ivan Allen Jr..Only a relative handful of such reports were received, the jury said, considering the widespread interest in the election, the number of voters and the size of this city. The jury said it did find that many of Georgia's registration and election laws are outmoded or inadequate and often ambiguous. It recommended that Fulton legislators act to have these laws studied and revised to the end of modernizing and improving them. The grand jury commented on a number of other topics, among them the Atlanta and Fulton County purchasing departments which it said are well operated and follow generally accepted practices which inure to the best interest of both governments. Merger proposed however, the jury said it believes these two offices should be combined to achieve greater efficiency and reduce the cost of administration. The City Purchasing Department, the jury said, is lacking in experienced clerical personnel as a result of city personnel policies. It urged that the city take steps to remedy this problem. Implementation of Georgia's automobile title law was also recommended by the outgoing jury. It urged that the next Legislature provide enabling funds and re-set the effective date so that an orderly implementation of the law may be effected. The grand jury took a swipe at the State Welfare Department's handling of federal funds granted for child welfare services in foster homes. This is one of the major items in the Fulton County general assistance program, the jury said, but the State Welfare Department has seen fit to distribute these funds through the welfare departments of all the counties in the state with the exception of Fulton County, which receives none of this money. The jurors said they realize a proportionate distribution of these funds might disable this program in our less populous counties. Nevertheless, we feel that in the future Fulton County should receive some portion of these available funds, the jurors said. Failure to do this will continue to place a disproportionate burden on Fulton taxpayers. The jury also commented on the Fulton ordinary's court which has been under fire for its practices in the appointment of appraisers, guardians and administrators and the awarding of fees and compensation. Wards protected the jury said it found the court has incorporated into its operating procedures the recommendations of two previous grand juries, the Atlanta Bar Association and an interim citizens committee. These actions should serve to protect in fact and in effect the court's wards from undue costs and its appointed and elected servants from unmeritorious criticisms, the jury said. Regarding Atlanta's new multi million dollar airport, the jury recommended that when the new management takes charge Jan. 1 the airport be operated in a manner that will eliminate political influences. The jury did not elaborate, but it added that there should be periodic surveillance of the pricing practices of the concessionaires for the purpose of keeping the prices reasonable. Ask jail deputies on other matters, the jury recommended that : 1 four additional deputies be employed at the Fulton County Jail and a doctor, medical intern or extern be employed for night and weekend duty at the jail. 2 Fulton legislators work with city officials to pass enabling legislation that will permit the establishment of a fair and equitable pension plan for city employees.
Total Number of words: 641 (Number of words which are actually executed as an input set)

Step-1: Input-text is processed using Stanford POS tagger to generate below mentioned output text [Stanford POS-tagger is executed for ∀ w_s] [Table 5.4].

Table 5.4 Sample of POS tagged Text

POS Tagged Sample Text
The/DT Fulton/NNP County/NNP Grand/NNP Jury/NNP said/VBD Friday/NNP an/DT investigation/NN of/IN Atlanta/NNP 's/POS recent/JJ primary/JJ election/NN produced/VBD no/DT evidence/NN that/IN any/DT irregularities/NNS took/VBD place/NN ./. The/DT jury/NN further/RB said/VBD in/IN term-end/JJ presentments/NNS that/IN the/DT City/NNP Executive/NNP Committee/NNP ,/, which/WDT had/VBD over-all/JJ charge/NN of/IN the/DT election/NN ,/, deserves/VBZ the/DT praise/NN and/CC thanks/NNS of/IN the/DT City/NN of/IN Atlanta/NNP for/IN the/DT manner/NN in/IN which/WDT the/DT election/NN was/VBD conducted/VBN ./. The/DT September-October/NNP term/NN jury/NN had/VBD been/VBN charged/VBN by/IN Fulton/NNP Superior/NNP Court/NNP Judge/NNP Durwood/NNP Pye/NNP to/TO investigate/VB reports/NNS of/IN possible/JJ irregularities/NNS in/IN the/DT hard-fought/JJ primary/JJ which/WDT was/VBD won/VBN by/IN Mayor-nominate/JJ Ivan/NNP Allen/NNP Jr./NNP ./. Only/RB a/DT relative/JJ handful/NN of/IN such/JJ reports/NNS was/VBD received/VBN ,/, the/DT jury/NN said/VBD ,/, considering/VBG the/DT widespread/JJ interest/NN in/IN the/DT election/NN ,/, the/DT number/NN of/IN voters/NNS and/CC the/DT size/NN of/IN this/DT city/NN ./. The/DT jury/NN said/VBD it/PRP did/VBD find/VB that/DT many/JJ of/IN Georgia/NNP 's/POS registration/NN and/CC election/NN laws/NNS are/VBP outmoded/JJ or/CC inadequate/JJ and/CC often/RB ambiguous/JJ ./. It/PRP recommended/VBD that/IN Fulton/NNP legislators/NNS act/VBP to/TO have/VB these/DT laws/NNS studied/VBN and/CC revised/VBN to/TO the/DT end/NN of/IN modernizing/VBG and/CC improving/VBG them/PRP ./. and/CC reduce/VB the/DT cost/NN of/IN administration/NN ./.

Step-2: POS tagged Unique Word list is generated (Table 5.5)**.**

Table 5.5 POS tagged Unique Sample Word List

Sample Unique Input words with POS							
Total Sample words 377 Total unique words 248							
exception NN	size NN	term NN	combined VBN	homes NNS	fair JJ	took VBD	policies NNS
Fulton NNP	durwood NNP	departments NNS	citizens NNS	clerical JJ	welfare NN	new JJ	best JJS
county NNP	receives VBZ	distribute VB	reports NNS	realize VBP	multi NNS	law NN	investigate VB
ambiguous JJ	Atlanta NNP	operating NN	medical JJ	might MD	thanks NNS	conducted VBN	wards NNS
granted VBN	charged VBN	takes VBZ	evidence NN	mayor-nominate JJ	executive NNP	accepted VBN	effective JJ
fit NN	items NNS	over-all JJ	added VBD	often RB	number NN	unmeritorious JJ	general JJ
bar NNP	actions NNS	legislation NN	term-end JJ	remedy VB	problem NN	establishment NN	equitable JJ
protected VBD	relative JJ	enabling VBG	administration NN	jail NN	officials NNS	less JJR	influences NNS
fire NN	nevertheless RB	receive VB	reasonable JJ	effected VBN	judge NNP	steps NNS	inure VBP
pension NN	fact NN	irregularities NNS	offices NNS	regarding VBG	greater JJR	produced VBD	friday NNP

Step-3: List of word pairs with their distance (S_p) is generated (Table 5.6)**.**

Table 5.6 List of Word-Pair with their distance value

Word-POS-Word-POS-Distance-Flag	**Word-POS-Word-POS-Distance-Flag**
county NNP court NN 0.6119791666666666 1	intern NN interest NN 0.5277343750000001 -1
county NNP counties NNS 0.5277343750000001 -1	intern NN interim JJ 0.34386160714285713 -1
granted VBN grand JJ 0.65234375 -1	administration NN administrators NNS 0.25305453213778406 1
protected VBD protect VB 0.246977306547619 -1	offices NNS officials NNS 0.6917317708333334 1
recent JJ receive VB 0.65234375 1	recommendations NNS recommended VBD 0.5806488037109375 1
received VBN receives VBZ 0.10463169642857142 -1	effected VBN effect NN 0.287109375 -1
received VBN receive VB 0.10463169642857142 -1	effected VBN effective JJ 0.4443359375 1
receives VBZ receive VB 0.10463169642857142 -1	elected VBN election NN 0.5277343750000001 1
charged VBN charge NN 0.12295386904761904 -1	legislature NNP legislators NNS 0.2426979758522727 1
departments NNS department NNP 0.07297141335227272 -1	interest NN interim JJ 0.5277343750000001 1
distribute VB distribution NN 0.3046739366319444 -1	pricing NN prices NNS 0.65234375 1
operating NN operated VBN 0.4443359375 1	plan NN place VB 0.571875 -1
takes VBZ take VB 0.190625 -1	laws NNS law NN 0.26041666666666663 -1
legislation NN legislature NNP 0.3397771661931818 1	generally RB general JJ 0.246977306547619 -1
legislation NN legislators NNS 0.3397771661931818 1	fees NNS feel VBP 0.26041666666666663 -1
employed VBN employees NNS 0.10463169642857142 -1	appointment NN appointed VBN 0.508938083400974 1

Step-4: A. Step 4.1 Display list of Groups which is generated from word-pair set (S_p).

Output: Lemmas (S_L={ L_1, L_2, L_3..}) for groups are shown in Table 5.7.

Table 5.7 Groups with Lemma List

Group-wise Allied morphed words (Generation of G_i' from **S_p**)					
Seq. No	**Group of allied morphed words (**G_i')	S_L **Lemma**	**Seq. No**	**Group of allied morphed words (**G_i')	S_L **Lemma**
1.	court NN 1 county NNP 1 counties NNS -1	Court country	2.	recommended VBD 1 recommendations NNS -1	Recommend
3.	grand JJ 1 granted VBN 1	Grand grant	4.	effect NN 1 effected VBN 1 effective NN 1	Effect Effect Effect
5.	protect VB 1 protected VBD -1	protect	6.	elected VBN 1 election NN 1	Elect
7.	recent JJ 1 receive VB 1 received VBN -1 receives VBZ -1	Recent receive	8.	prices NNS 1 pricing NN 1	Price Price
9.	charge NN 1 charged VBN -1	charge	10.	plan NN 1 place VB 1	Plan Place
11.	department NNP 1 departments NNS -1	department	12.	law NN 1 laws NNS -1	Law
13.	distribute VB 1 distribution NN 1	Distribute distribute	14.	general JJ 1 generally RB -1	General
15.	operated VBN 1 operating NN 1	Operate operate	16.	feel VBP 1 fees NNS 1	Feel Fees
17.	take VB 1 takes VBZ -1	take	18.	appointed VBN 1 appointment NN -1	Appoint
19.	legislation NN 1 legislators NNS 1 legislature NNP 1	Legislate Legislate legislate	20.	employed VBN 1 employees NNS 1	Employ employ
21.	intern NN 1 interim NN 1 interest NN 1	Intern Interim Interest	22.	investigate VB 1 investigation NN -1	Investigate
23.	administration NN 1 administrators NNS 1	Administrate	24.	cost NN 1 costs NNS -1	Cost
25.	offices NNS 1 officials NNS 1	Office	26.	protect VB 1 protected VBD -1	Protect

Step-4.2 B. List of Discrete words are depicted (Set- S_p') in Table 5.8.

Table 5.8 Discrete Word List

Seq. No	Input-Words	Input-Words	Input-Words	Input-Words	Input-Words
1.	exception	Awarding	act	concessionaires	Georgia
2.	Fulton	Primary	continue	found	major
3.	ambiguous	City	incorporated	procedures	find
4.	fit	Periodic	won	term	widespread
5.	bar	Portion	homes	over-all	funds
6.	fire	Weekend	clerical	enabling	criticisms
7.	pension	Operated	realize	receive	administrators
8.	state	Distribution	might	irregularities	employees
9.	undue	Modernizing	mayor-nominate	committee	took
10.	inadequate	Believes	often	charge	new
11.	pass	Orderly	remedy	keeping	law
12.	night	Federal	jail	lacking	conducted
13.	ivan	Act	regarding	possible	accepted
14.	court	Effective	take	services	unmeritorious
15.	airport	May	improving	many	establishment
16.	size	Topics	servants	practices	less
17.	durwood	Disable	failure	ask	steps
18.	Atlanta	Future	counties	combined	produced
19.	items	Foster	child	citizens	appointed
20.	actions	Juries	however	reports	proposed
21.	relative	september-october	next	medical	interim
22.	nevertheless	Pye	outgoing	evidence	swipe
23.	fact	Date	proportionate	added	deputies
24.	achieve	Allen	purchasing	term-end	populous
25.	association	Feel	jan.	reasonable	appraisers
26.	eliminate	Political	taxpayers	available	purpose
27.	seen	Manner	additional	extern	deserves
28.	dollar	Serve	title	hard-fought	policies
29.	result		assistance	jurors	best
30.	automobile		burden	outmoded	wards

Step-5 Final Output: S_L: List [lemmas]: {L_1, L_2, L_3.....} through function "$f_{LemmaQuest}$(List[w_s])". Table 5.9 shows list of lemmas generated from input text.

Note: The words and lemmas in red colour are incorrectly generated lemma.

Table 5.9 Input Words with their Lemmas

Total Number of words: 248 Total Number of Lemmas: 218								
	Input Words	**Lemma**	**Input Words**	**Lemma**	**Input Words**	**Lemma**	**Input Words**	**Lemma**
1.	accepted	accept	jr.	jr	combined	combine	officials	office
2.	achieve	achieve	judge	judge	commented	comment	often	often
3.	actions	act	jurors	juror	committee	commit	operating	operate
4.	act	act	jury	jury	compensation	compensate	operated	operate
5.	added	add	juries	jury	concessionaires	concessionaire	orderly	order
6.	additional	add	keeping	keep	conducted	conduct	ordinary	ordinary
7.	administration	administrate	lacking	lack	considering	consider	outgoing	outgo
8.	administrators	administrate	laws	law	continue	continue	outmoded	outmode
9.	airport	airport	law	law	costs	cost	over-all	overall
10.	allen	allen	legislation	legislate	cost	cost	pass	pass
11.	also	also	legislature	legislate	county	county	pension	pension
12.	ambiguous	ambiguous	legislators	legislate	counties	county	periodic	period
13.	among	among	less	less	court	court	permit	permit
14.	appointment	appoint	major	major	criticisms	critic	personnel	personnel
15.	appointed	appoint	management	manage	date	date	place	place
16.	appraisers	appraise	manner	manner	departments	department	plan	plan
17.	ask	ask	many	many	department	department	policies	policy
18.	assistance	assist	matters	matter	deputies	deputy	political	politics
19.	association	associate	may	may	deserves	deserve	populous	populous
20.	atlanta	atlanta	medical	medicine	disable	disable	portion	port
21.	automobile	automobile	merger	merge	disproportionate	disproportion	possible	possible
22.	available	avail	might	might	distribute	distribute	practices	practice
23.	awarding	award	modernizing	modern	distribution	distribute	praise	praise

24.	bar	bar	money	money	doctor	doctor	presentments	present
25.	believes	believe	multi	multi	dollar	dollar	previous	previous
26.	best	best	nevertheless	nevertheless	durwood	durwood	pricing	price
27.	burden	burden	new	new	duty	duty	prices	price
28.	charged	charge	next	next	effected	effect	primary	primary
29.	charge	charge	night	night	effect	effect	problem	problem
30.	child	child	nominate	nominate	effective	effect	procedures	procedure
31.	citizens	citizen	none	none	efficiency	efficient	produced	produce
32.	city	city	number	number	elaborate	elaborate	program	program
33.	clerical	clerk	offices	office	elected	elect	proportionate	proportion
34.	election	elect	proposed	propose	fair	fair	reduce	reduce
35.	eliminate	eliminate	protected	protect	federal	federal	regarding	regard
36.	employed	employ	protect	protect	fees	fee	registration	registry
37.	employees	employ	provide	provide	feel	feel	relative	relate
38.	enabling	enable	purchasing	purchase	hard	hard	remedy	remedy
39.	end	end	purpose	purpose	found	find	reports	report
40.	equitable	equitable	pye	pye	find	find	re-set	reset
41.	establishment	establish	realize	real	fire	fire	result	result
42.	evidence	evidence	reasonable	reason	fit	fit	revised	revise
43.	exception	except	received	receive	follow	follow	said	say
44.	executive	execute	receives	receive	foster	foster	seen	see
45.	experienced	experience	receive	receive	friday	friday	september	september
46.	extern	extern	recent	recent	fulton	fulton	servants	servant
47.	fact	fact	recommendations	recommend	funds	fund	serve	serve
48.	failure	fail	recommended	recommend	future	future	services	service
49.	inure	inure	wards	ward	generally	general	size	size
50.	investigate	investigate	weekend	weekend	general	general	state	state
51.	investigation	investigate	welfare	welfare	georgia	georgia	steps	step
52.	irregularities	irregular	well	well	governments	govern	studied	study
53.	items	item	widespread	widespread	grand	grand	superior	superior
54.	ivan	ivan	will	will	granted	grant	surveillance	surveillance
55.	jail	jail	won	win	greater	great	swipe	swipe
56.	jan.	jan	work	work	guardians	guard	takes	take
57.	handful	hand	take	take	october	october	fought	fight
58.	handling	handle	took	take	homes	home	taxpayers	taxpayers
59.	homes	home	taxpayers	taxpayer	however	however	term	term
60.	however	however	term	term	implementation	implement	term-end	term-end
61.	implementation	implement	term-end	term-end	improving	improve	thanks	thanks
62.	improving	improve	thanks	thanks	inadequate	inadequate	title	title
63.	inadequate	inadequate	title	title	incorporated	incorporate	topics	topic
64.	incorporated	incorporate	topics	topics	influences	influence	undue	undue
65.	influences	influence	undue	undue	interest	interest	unmeritorious	unmeritorious
66.	interest	interest	unmeritorious	unmeritorious	interim	interim	urged	urge

Input 2: Sample input text of DUC corpus (w_s) is accepted as input-text file for generating lemmas. Table 5.10 shows sample Brown input text.

Table 5.10 B. Sample of Input Text of Brown

Prince Philip Condemns IRA, Clergy Urge Forgiveness DEAL, Britain (AP) Prince Philip on Sunday condemned the ``senseless'' killing of 10 Royal Marines musicians in an IRA bombing, and Britain's defense secretary said he warned all military bases of the risk of similar attacks. Clergyman urged relatives and friends of dead and maimed musicians to forgive the bombers. ``Only forgiveness breaks the tie between the hater and the hated,'' the Rev. George Lings told mourners. The prince, husband of Queen Elizabeth II and captain general of the Royal Marines, visited injured men in the hospital and toured the severely damaged Royal Marines School of Music in Deal, southeast Britain. ``It will not help the IRA win anything,'' said Philip, who wore a Royal Marines tie. ``It is senseless. One simply wonders what sort of mentality can even contemplate such meaningless acts. It is appalling.'' He paid tribute to the 12 injured men, five of whom were critically wounded. The prince was accompanied by Viscountess Mountbatten, daughter of Lord Mountbatten, who was killed by an IRA bomb on his boat in 1979. Mountbatten was India's last viceroy and a cousin of the queen. British military installations are a frequent target of the Irish Republican Army in its campaign to end British rule in Northern Ireland and unite the predominantly Protestant province with the Roman Catholic Republic of Ireland. Defense Secretary Tom King said Sunday he has issued an alert to all military installations to prevent other attacks. He would not give details. ``The perpetrators of the latest outrage are at large and there is a risk of other attacks,'' King said. ``That is why we are taking a number of other steps.'' King defended the use of private security firms hired to guard the Deal school and 29 other ``low-risk'' military installations in Britain. Local residents and grieving relatives have said security was lax and should be turned back to the marines. King said the private firms will remain. ``It is important to remember that what we need in these cases are eyes and ears and observation,'' King said on British Broadcasting Corp. radio. ``Private security guards can be a very useful additional assistance. They also help to reduce the amount of time soldiers have to spend on what is not the most enjoyable part of their activity.'' At church services throughout the small port, clergymen urged mourners to pray for and forgive the bombers. The Rev. Charles Howard, a Royal Navy chaplain, asked the 300-member congregation inside the barracks, ``if you can find room in your hearts ... pray for the men who perpetrated this terrible act, that God will soften their hearts and turn them from their violent and evil ways.'' Many cried as Howard read aloud the names of the 10 servicemen killed during a coffee break between band practices. It was the worst IRA attack on the British mainland since July 1982. At nearby St. George's Church, formerly the Royal Marines' church, Lings said, ``no one says forgiveness is easy; no one says the terrorist deserves forgiveness. ``But ... forgive them, they not what they do.''

Step 1: Input-text is processed using Stanford POS tagger to generate below mentioned text [Stanford POS-tagger is executed for ∀ w_s] (Table 5.11).

Table 5.11 POS tagged Text

Prince/NNP Philip/NNP Condemns/VBZ IRA/NNP ,/, Clergy/NNP Urge/NNP Forgiveness/NNP DEAL/NNP ,/, Britain/NNP -LRB-/-LRB- AP/NNP -RRB-/-RRB- Prince/NNP Philip/NNP on/IN Sunday/NNP condemned/VBD the/DT ``/`` senseless/JJ "/" killing/NN of/IN 10/CD Royal/NNP Marines/NNPS musicians/NNS in/IN an/DT IRA/NNP bombing/NN ,/, and/CC Britain/NNP 's/POS defense/NN secretary/NN said/VBD he/PRP warned/VBD all/DT military/JJ bases/NNS of/IN the/DT risk/NN of/IN similar/JJ attacks/NNS ./. Clergyman/NN urged/VBD relatives/NNS and/CC friends/NNS of/IN dead/JJ and/CC maimed/JJ musicians/NNS to/TO forgive/VB the/DT bombers/NNS ./. ``/`` Only/RB forgiveness/NN breaks/VBZ the/DT tie/NN between/IN the/DT hater/NN and/CC the/DT hated/VBN ,/, "/" the/DT Rev./NNP George/NNP Lings/NNP told/VBD mourners/NNS ./. The/DT prince/NN ,/, husband/NN of/IN Queen/NNP Elizabeth/NNP II/NNP and/CC captain/NN general/NN of/IN the/DT Royal/NNP Marines/NNPS ,/, visited/VBD injured/JJ men/NNS in/IN the/DT hospital/NN and/CC toured/VBD the/DT severely/RB damaged/VBN Royal/NNP Marines/NNPS School/NNP of/IN Music/NNP in/IN Deal/NNP ,/, southeast/NN Britain/NNP ./.

Step 2: List of unique POS tagged words is generated. Table 5.12 shows unique word list.

Table 5.12 POS tagged Unique Sample Word List

Total number of words 319 , Total number of unique words 217		
Seq. No	**Words , POS**	**Words, POS**
1.	acts NNS	observation NN
2.	urge NNP	contemplate VB
3.	guards NNS	Mountbatten NNP
4.	music NNP	severely RB
5.	province NN	issued VBN
6.	hired VBD	lings NNP
7.	band NN	congregation NN
8.	meaningless JJ	reduce VB
9.	barracks NNS	similar JJ
10.	remain VB	give VB
11.	forgiveness NN	large JJ
12.	July NNP	urged VBD
13.	viscountess NNP	broadcasting NNP
14.	easy JJ	Sunday NNP
15.	activity NN	accompanied VBN
16.	critically RB	enjoyable JJ

Step-3: List of word pairs with their distance (S_p) is generated and Table 5.13 shows this list**.**

Table 5.13 List of Word-Pair with Their Distance Value

acts NNS act NN 0.26041666666666663 -1	turn VB turned VBN 0.4296875 -1
urge NNP urged VBD 0.190625 -1	defense NNP defended VBD 0.376953125 1
guards NNS guard VB 0.14947916666666666 -1	attack NN attacks NNS 0.12295386904761904 -1
music NNP musicians NNS 0.6917317708333334 -1	condemned VBD condemns VBZ 0.246977306547619 -1
forgiveness NN forgive VBP 0.508938083400974 -1	republic NNP republican JJ 0.21708984375 -1
servicemen NNS services NNS 0.38466796875 -1	bomb NN bombing NN 0.65234375 -1
clergymen NNS clergyman NN 0.16465153769841268 1	perpetrated VBD perpetrators NNS 0.3046739366319444 1
clergymen NNS clergy NNP 0.4443359375 -1	clergyman NN clergy NNP 0.4443359375 -1
bombers NNS bomb NN 0.65234375 -1	worst JJS wore VBD 0.571875 -1
bombers NNS bombing NN 0.65234375 1	breaks VBZ break NN 0.14947916666666666 -1
deal NNP dead JJ 0.26041666666666663 -1	britain NNP british JJ 0.65234375 1
killing NN killed VBN 0.65234375 1	hater NN hated VBN 0.190625 -1

Step-4: List **of** Groups with their Lemmas, are depicted**.** Table 5.14 shows list of groups.

Table 5.14 List of Groups with Lemmas

Seq. No	Group	Lemma	Seq. No	Group	Lemma
1.	act NN -1 acts NNS -1	act	2.	turn VB 1 turned VBN 1	turn
3.	urge NNP -1 urged VBD -1	urge	4.	hated VBN -1 hater NN -1	hate
5.	guard VB -1 guards NNS -1	guard	6.	attack NN -1 attacks NNS -1	attack
7.	music NNP -1 musicians NNS -1	music	8.	condemns VBZ -1 condemned VBD -1	condemn
9.	forgive VBP -1 forgiveness NN -1	forgive	10.	republic NNP -1 republican JJ -1	republic
11.	services NNS 1 servicemen NNS 1	service	12.	bomb NN 1 bombing NN 1	bomb
13.	clergy NNP -1 clergyman NN 1 clergymen NNS 1	clergy	14.	perpetrated VBD 1 perpetrators NNS 1	perpetrate
15.	bomb NN -1 bombers NNS 1 bombing NN 1	bomb	16.	clergy NNP -1 clergyman NN -1	clergy , clergyman
17.	dead JJ 1 deal NNP 1	dead, deal	18.	wore VBD -1 worst JJS -1	wear worst
19.	killed VBN 1 killing NN 1	kill	20.	break NN 1 breaks VBZ 1	break
21.	turn VB 1 turned VBN 1	turn	22.	Britain NNP 1 British JJ 1	Britain British
23.	defence NNP 1 defended VBD 1	defence	24.	hated VBN -1 hater NN -1	hate

Step-4.1 Discrete Word List ($S_{P'}$) is depicted in Table 5.15.

Table 5.15 Discrete Word List

Seq. No	Input Words	Input Words	Input Words	Input Words
1.	province	broadcasting	grieving	asked
2.	hired	Sunday	corp.	accompanied
3.	band	accompanied	one	clergy
4.	meaningless	enjoyable	services	congregation
5.	barracks	even	many	break
6.	remain	warned	practices	mentality
7.	July	catholic	god	observation
8.	viscountess	low-risk	violent	boat
9.	easy	wore	predominantly	help
10.	secretary	southeast	Philip	names
11.	critically	turned	inside	damaged
12.	firms	queen	tie	royal
13.	activity	installations	act	small
14.	dead	republican	attacks	navy
15.	eyes	guard	school	use
16.	bases	church	military	roman
17.	remember	Irish	details	viceroy
18.	cried	defended	mourners	outrage
19.	secretary	local	last	husband
20.	ira	injured	need	british
21.	George	tom	relatives	northern
22.	latest	ways	mainland	prLemmasay

Step-5 Final Output: S_L: List [lemmas] is generated and table 5.16 shows this output list.

Table 5.16 Input Words with Their Lemmas

Total Number of words: 216, Number of Lemmas: 198				
Seq. No	**Words**	**Lemma**	**Words**	**Lemma**
1.	accompanied	accompany	killing	kill
2.	acts	act	killed	kill
3.	act	act	king	king
4.	activity	activity	large	large
5.	additional	addition	last	last
6.	alert	alert	latest	latest
7.	aloud	aloud	lax	lax
8.	also	also	lings	ling
9.	amount	amount	local	local
10.	anything	anything	lord	lord
11.	critically	critical critic	low-risk	low
12.	appalling	appal	maimed	maim
13.	army	army	mainland	mainland
14.	asked	ask	men	man
15.	assistance	assist	many	many
16.	attack	attack	marines	marine
17.	attacks	attack	meaningless	meaningless
18.	urged	urge	terrorist	terror
19.	important	import	throughout	throughout
20.	India	India	tie	tie
21.	injured	injure	time	time
22.	inside	inside	tom	tom
23.	installations	install	toured	tour
24.	ira	ira	tribute	tribute
25.	Ireland	Ireland	turn	turn
26.	Irish	Irish	turned	turn
27.	issued	issue	unite	unite
28.	July	July	urge	urge
29.	killing	kill	urged	urge
30.	observation	observe	bombers	bomb
31.	musicians	music	perpetrated	perpetrate
32.	mentality	mental	perpetrators	perpetrate

Input 3: Another DUC text (w_s) is accepted as an input which is depicted in Table 5.17.

Table 5.17 Sample Input-DUC Text

Gilbert Reaches Jamaican Capital With 110 Mph wind. Hurricane Gilbert, packing 110 mph wind and torrential rain, moved over this capital city today after skirting Puerto Rico, Haiti and the Dominican Republic. There were no immediate report of casualty. Telephone communication were affected. ``Right now Hurricane Gilbert is actually moving over Jamaica," said Bob Sheets, director of the National Hurricane Center in Miami. ``National Hurricane Center have already had report of 110 mph wind on the eastern tip. ``Hurricane Gilbert looks like the eye is going to move lengthwise across that island, and National Hurricane Center are going to bear the full brunt of this powerful hurricane," Sheets said. Forecaster say Gilbert was expected to lash Jamaica throughout the day and was on track to later strike the Cayman island, a small British dependency northwest of Jamaica.

Table 5.18 Sample POS-tagged Text

Step 1-Assign POS tag with each word of the text in Table 5.18. **DUC Text Sample-Word Collection with POS**

Gilbert/NNP Reaches/VBZ Jamaican/JJ Capital/NNP With/IN 110/CD Mph/NNP Winds/NNP Hurricane/NNP Gilbert/NNP ,/, packing/NN 110/CD mph/NN winds/NNS and/CC torrential/JJ rain/NN ,/, moved/VBD over/IN this/DT capital/NN city/NN today/NN after/IN skirting/VBG Puerto/NNP Rico/NNP ,/, Haiti/NNP and/CC the/DT Dominican/NNP Republic/NNP ./. There/EX were/VBD no/DT immediate/JJ reports/NNS of/IN casualties/NNS ./. Telephone/NNP communications/NNS were/VBD affected/VBN ./. ``/`` Right/RB now/RB it/PRP 's/VBZ actually/RB moving/VBG over/IN Jamaica/NNP ,/, "/" said/VBD Bob/NNP Sheets/NNP ,/, director/NN of/IN the/DT National/NNP Hurricane/NNP Center/NNP in/IN Miami/NNP ./. ``/`` We/PRP 've/VBP already/RB had/VBN reports/NNS of/IN 110/CD mph/NN winds/NNS on/IN the/DT eastern/JJ tip/NN ./. ``/`` It/PRP looks/VBZ like/IN the/DT eye/NN is/VBZ going/VBG to/TO move/VB lengthwise/NN across/IN that/DT island/NN ,/, and/CC they/PRP 're/VBP going/VBG to/TO bear/VB the/DT full/JJ brunt/NN of/IN this/DT powerful/JJ hurricane/NN ,/, "/" Sheets/NNPS said/VBD ./. Forecasters/NNS say/VBP Gilbert/NNP was/VBD expected/VBN to/TO lash/VB Jamaica/NNP throughout/IN the/DT day/NN and/CC was/VBD on/IN track/NN to/TO later/RB strike/VB the/DT Cayman/NNP Islands/NNPS ,/, Forecasters/NNS at/IN the/DT center/NN said/VBD the/DT eye/NN of/IN Gilbert/NNP was/VBD 140/CD miles/NNS southeast/NN of/IN Kingston/NNP at/IN dawn/NN today/NN ./. Maximum/NNP sustained/VBD winds/NNS were/VBD near/IN 110/CD mph/NN ,/, with/IN tropical-storm/NN force/NN winds/NNS extending/VBG up/RP to/TO 250/CD miles/NNS to/TO the/DT north/NN and/CC 100/CD miles/NNS to/TO the/DT south/NN ./. Sunday/NNP night/NN :/: "/" Cuba/NNP 's/POS official/JJ Prensa/NNP Latina/NNP news/NN agency/NN said/VBD a/DT state/NN of/IN alert/NN was/VBD declared/VBN at/IN midday/NN in/IN the/DT Cuban/JJ provinces/NNS of/IN Guantanamo/NNP ,/, Holguin/NNP ,/, Santiago/NNP de/IN Cuba/NNP and/CC Granma/NNP ./. In/IN the/DT report/NN from/IN Havana/NNP received/VBD in/IN Mexico/NNP City/NNP ,/, Prensa/NNP Latina/NNP said/VBD civil/JJ defense/NN officials/NNS were/VBD broadcasting/VBG bulletins/NNS on/IN national/JJ radio/NN and/CC television/NN recommending/VBG emergency/NN measures/NNS and/CC providing/VBG information/NN on/IN the/DT storm/NN ./. Flights/NNS were/VBD canceled/VBN Sunday/NNP in/IN the/DT Dominican/NNP Republic/NNP ,/, where/WRB civil/JJ defense/NN director/NN Eugenio/NNP Cabral/NNP reported/VBD some/DT flooding/NN in/IN parts/NNS of/IN the/DT capital/NN of/IN Santo/NNP Domingo/NNP and/CC power/NN outages/NNS there/RB and/CC in/IN other/JJ southern/JJ areas/NNS ./.

Step 2- Unique Words List is generated (Table 5.19)**.**

Table 5.19 Unique Words List

Gilbert Reaches Jamaican packing mph winds torrential rain moved capital city today skirting immediate Reports casualties communications affected Right now actually moving said director have already Reports mph winds eastern tip looks like eye going move lengthwise across island are going

Step 3- List of word pairs with their distance (S_p) is generated which are depicted in Table 5.20**.**

Table 5.20 Word-Pair with Distance Value

agencies agency 0.5277343750000001	hurricane hurricanes 0.08104383680555555	packed packing 0.65234375
alert alerted 0.34386160714285713	Jamaican Jamaicans 0.09125434027777778	parish parishes 0.287109375
appeals appears 0.22924107142857145	move moved 0.190625	parish parts 0.6119791666666666
broadcast broadcasting 0.3046739366319444	move moves 0.190625	pass passed 0.4296875
caused causing 0.65234375	moved moves 0.190625	power powerful 0.5277343750000001
center central 0.65234375	state stayed 0.6119791666666666	providing provinces 0.6917317708333334
coast coastal 0.34386160714285713	west western 0.65234375	reaches reaching 0.5277343750000001
danger dangerous 0.4443359375	windows winds 0.65234375	receive received 0.10463169642857142
east eastern 0.65234375	north northeast 0.6917317708333334	report reported 0.287109375
flood flooded 0.34386160714285713	north northward 0.6917317708333334	report reports 0.12295386904761904
flood flooding 0.5277343750000001	north northwest 0.6917317708333334	reported reports 0.287109375
flooded flooding 0.5277343750000001	northeast northward 0.5072699652777778	seek seen 0.26041666666666663
forecaster forecasters 0.07297141335227272	northeast northwest 0.5533854166666666	south southeast 0.6917317708333334
heads heavy 0.571875	northward northwest 0.4443359375	south southern 0.5277343750000001
high higher 0.4296875	official officials 0.09125434027777778	southeast southeastern 0.3046739366319444
		southeast southern 0.4443359375

Step 4- Groups and Group-wise Lemmas are generated which are depicted in Table 5.21.

Table 5.21 Groups with Lemmas

Seq. No	Group of allied morphed words (G_i')	Lemma (L_i)
1.	agency , agencies	agency
2.	alert, alerted	alert
3.	appeals, appears, appear	appeal, appear
4.	broadcast, broadcasting	broadcast
5.	cause, causing	cause
6.	center, central	center
7.	coast, coastal	coast
8.	danger, dangerous	danger
9.	east, eastern	east
10.	flood, flooding, flooded	flood
11.	forecaster, forecasters	forecaster
12.	head, heads, heavy	head, heavy
13.	high, higher	high
14.	hurricane, hurricanes	hurricane
15.	jamaican, jamaicans	jamaica
16.	move, moved, moves	move
17.	north, northwest, northward	north
18.	official, officiate	office
19.	pack, packed, packing, parish	pack, parish

Step 4.1- Lemmas are generated from discrete words **(Set : S_p')** which is depicted in Table 5.22.

Table 5.22 Output: Collection of Lemmas of Discrete Words

Seq. No	Word	Lemma	Word	Lemma
1.	actually	actual	casualties	casualty
2.	adding	add	communications	communicate
3.	affected	affect	defense	defend
4.	arrived	arrive	discontinued	discontinue
5.	began	begin	government	govern
6.	boarding	board	information	inform
7.	bound	bind	instructions	instruct
8.	branches	branch	meteorologist	meteorology
9.	brushing	brush	news	news
10.	bulletins	bulletin	preparation	prepare
11.	canceled	cancel	vacationer	vacation

Input 3- Oxford Dictionary Morphed sample words are accepted as an input and are depicted in Table 5.23.

Table 5.23 Sample Text of Morphed words

Sample Morphed Word Collection
Abruptly abruptness/ Absconded absconders absconding absconds /Absence absences absented absentee absentees absenting absently absents absentia / Absorbed absorbency absorbent absorber absorbers absorbing absorption/ Academia academic academically academics academies/ Accented accenting accents accentual accentuate/ Acceptable acceptably acceptance acceptor acceptors/ Accessed accesses accessible accessibly accessing accessory/ Accidence accidental accidents/ Acclaimed acclaimers acclaiming acclaims /Achievable achieved achiever achievers achieves achieving/ Acidic acidities acidity acidly acidosis acidified acidifier acidifiers acidifies acidify acidifying/ Algebra algebraic algebras/ Algeria Algerian Algerians

Groups are created and single lemma is generated for each group. All words of sample text reside within groups. No discrete word list is generated for this input text. Table 5.24 shows set of morphed words of each group and list of lemmas for all groups.

Table 5.24 Group-wise Lemma

Group of allied morphed words (G_i')	**Lemma (L_i)**
abruptly RB , abruptness JJ	Abrupt
Absconds JJ, absconded VBD, absconders NNS, absconding VBG	Abscond
absence NN absents NNS absences NNS absented VBD absentee NN absentia JJ absently RB absentees NNS absenting VBG	Absent
absorbed VBN absorber NN absorbent JJ absorbers NNS absorbing VBG absorbency NN	Absorb
academia NN academic JJ academies NNS academically JJ	Academy
accents NNS accented VBD acceptor NN accessed VBD accenting VBG accentual JJ accentuate NN	Accent, accept
Acceptor NN acceptor NN acceptable JJ acceptable RB acceptance NN	Accept
Accessed VBD accesses NNS accessing VBD accessory NNS accessible JJ accessibly RB	Access
accidence NN accidents NNS accidental JJ	Accident
acclaims NNS acclaimed JJ acclaimers NNS acclaiming JJ	Acclaim
Aced VBD aces VBZ	Ace
Acidic JJ acidly RB acidify NN acidity NN acidified VBD acidifier NN acidifies VB acidifiers NNS acidifying VBG	Acid
Algebra NN Algeria FW algebras FW Algeria JJ algebraic JJ Algerians NNS	Algebra, Algeria

Input 4- Oxford dictionary morphed word list is accepted as an input file and is depicted in Table 5.25.

Table 5.25 Oxford Dictionary Sample Words with Lemmas

Total Words	**277 words**	**Number of Lemmas 196**							
Words	**Lemmas**	**Words**	**Lemmas**	**Words**	**Lemmas**	**Words**	**Lemmas**	**Words**	**Lemmas**
cheeriness	cheeriness	academicals	academy	advisable	advise	astonish	astonish	carbonate	carbonate
ability	ability	academically	academy	algebra	algebra	astronomical	astronomic	childish	child
abnormal	abnormal	academy	academy	algebraic	algebra	astronaut	astronaut	chloride	chloride
abolish	abolish	academic	academy	algerian	algeria	astronomically	astronomic	chosen	chosen
abortion	abort	academics	academy	algeria	algeria	astronomer	astronomy	collector	collect
abort	abort	accent	accent	allotments	allot	athletic	athlete	collection	collect
abortive	abort	accentuate	accentuate	amazing	amaze	attended	attend	collision	collide
abound	abound	accept	accept	amazement	amaze	attentive	attend	comical	come
abrupt	abrupt	acceptor	accept	amazed	amaze	attending	attend	communist	commune
abruptness	abrupt	accessibly	access	amused	amuse	attention	attend	complaints	complain
abscond	abscond	access	access	angrily	angry	attractive	attract	completely	complete
absent	absent	accessory	access	annoy	annoy	attractive	attract	complications	complex
absentia	absent	accessible	access	annoyance	annoy	attracted	attract	compliance	comply
absenteeism	absent	accidental	accident	annoying	annoy	attracting	attract	comprehension	comprehend
absolute	absolute	accomplishment	accomplish	applicability	apply	attraction	attract	conducting	conduct
absolutism	absolute	achievable	achieve	appliances	apply	awful	awe	conduction	conduct
absorb	absorb	acoustically	acoustic	applications	apply	bakery	bake	conductive	conduct
absorbent	absorbent	actor	act	appliers	apply	basic	basic	conductors	conduct
abstract	abstract	active	act	applicants	apply	beautiful	beauty	conductor	conduct
absurdness	absurd	activity	act	applicable	apply	beggar	beg	conductivity	conduct
absurd	absurd	actively	act	application	apply	believable	believe	conductivities	conduct
absurdity	absurd	action	act	appointee	appoint	blackness	black	conductresses	conductress
absurdum	absurdum	activeness	act	approachable	approach	blacken	blacken	conductress	conductress
abundance	abundance	administrations	administrate	approaching	approach	birth	born	costly	cost
abundant	abundant	admiration	admire	approached	approach	bravery	brave	courteous	court
abuse	abuse	admirable	admire	argument	argue	breakage	break	creamery	cream
abuser	abuse	admired	admire	arrival	arrive	burglar	burgle	creamery	cream
abusive	abuse	admirer	admire	artistically	artist	busily	busy	curiosity	curios

Input 5- Oxford dictionary nominalised sample double/ triple adverb words (input) with lemmas (output) are depicted in Table 5.26.
Output: S_L: List [lemmas]

Table 5.26 Nominalized Adverb Words with Lemmas

Total number of words 127		**Total number of lemmas 127**			
Words	**Lemmas**	**Words**	**Lemmas**	**Words**	**Lemmas**
abnormally	abnormal	leisurely	leisure	idly	idle
academically	academic academy	lengthwise	lengthwise	imaginatively	imaginative imagine
accidentally	accident	lifelessly	lifeless	irresponsibly	irresponsible
accurately	accurate	lovingly	loving	jealously	jealous
acoustically	acoustic	loyally	loyal	job-wise	job
affectionately	affection	luckily	lucky	jokingly	joking
angrily	angry	magically	magical magic	joyfully	joyful
anxiously	anxious	magnificently	magnificent	justly	just
artfully	artful	maturely	mature	kindly	kind
artistically	artistic artist	mechanically	mechanical, mechanic	knowledgeably	knowledge
awesomely	awesome	mindlessly	mindless	lawfully	lawful
awkwardly	awkward	miraculously	miraculous	leisurely	leisure
badly	bad	miserably	miserable misery	expertly	expert
beautifully	beautiful beauty	musically	musical music	extraordinarily	extraordinary
briskly	brisk	naturally	natural nature	famously	famous
brutally	brutal	neatly	neat	fashionably	fashion
busily	busy	nobly	noble	freely	free
calmly	calm	noisily	noise	furiously	furious fury
capably	capable	oddly	odd	gently	gentle
carefully	careful care	officially	official office	gracefully	graceful grace
cautiously	cautious	otherwise	otherwise	guiltily	guilty
cheerfully	cheerful cheer	painfully	painful pain	happily	happy
classically	classical class	personally	personal person	harshly	harsh
clearly	clear	politically	politics	helpfully	helpful help
cleverly	clever	possibly	possible	hopefully	hopeful hope
clockwise	clockwise	probably	probable probe	hurriedly	hurried hurry
colourfully	colourful colour	proudly	proud	energetically	energetic energy
comfortably	comfort	punctually	punctual	extraordinarily	extraordinary
competitively	competitive	purposefully	purposeful purpose	tastefully	tasteful tasty
completely	complete	quickly	quick	tenderly	tender
confidently	confident	readily	ready	terribly	terrible
counterclockwise	counterclockwise	really	real	thoroughly	thorough
cowardly	coward	regretfully	regretful regret	tragically	tragical tragic
crazily	crazy	religiously	religious	uniquely	unique
customarily	customary custom	rightly	right	universally	universe
definitely	definite	romantically	romantic	untruthfully	untruthful untruth
deliberately	deliberate	sadly	sad	vocally	vocal
delightfully	delightful delight	safely	safe	voluntarily	voluntary
dependably	depend	secretly	secret	warmly	warm
desperately	desperate	silently	silent	watchfully	watchful watch
distinctly	distinct	skilfully	skilful skill	weakly	weak
doubtfully	doubtful doubt	sleepily	sleep	willingly	willing will
eagerly	eager	steadily	steady	suspiciously	suspicious

Input-6: Dictionary Sample Nominalized/Derived Words (Verb/Noun/Adjective/Adverb POS) with their Lemmas.
Output: S_L: List [lemmas]: {L_1, L_2, L_3.....}.

Table 5.27 Nominalized/Derived Words with Lemmas

Total Nominalized/Derived Words : 177 Total lemmas generated : 29									
Input-words	**Lemma**	**Input-words**	**Lemma**	**Input-words**	**Lemma**	**Input-words**	**Lemma**	**Input-words**	**Lemma**
applies	apply	appointive	appoint	appropriates	appropriate	approximated	approximate	arrangers	arrange
applicability	apply	appointer	appoint	appropriator	appropriate	approximates	approximate	arranges	arrange
applicant	apply	appointers	appoint	appropriately	appropriate	archer	arch	arranger	arrange
applicators	apply	apportioning	apportion	appropriating	appropriate	arch	arch	arranged	arrange
applied	apply	apportion	apportion	appropriation	appropriate	arches	arch	arrangement	arrange
applicable	apply	apportionment	apportion	appropriations	appropriate	arched	arch	arrangements	arrange
applicator	apply	apportioned	apportion	approved	approve	arching	arch	arranging	arrange
application	apply	apportions	apportion	approvers	approve	archers	arch	arrange	arrange

appliances	apply	appraisals	appraise	approver	approve	archery	arch	arraying	array
appliance	apply	appraise	appraise	approves	approve	archly	arch	arrayed	array
apply	apply	appraisingly	appraise	approvals	approve	architecturally	architect	arrays	array
applications	apply	appraising	appraise	approval	approve	architectures	architect	array	array
applicants	apply	appraises	appraise	approvingly	approve	architects	architect	arrears	arrears
applying	apply	appraiser	appraise	approving	approve	architect	architect	arrestingly	arrest
appointing	appoint	appraised	appraise	approve	approve	architecture	architect	arrests	arrest
appointees	appoint	appraisers	appraise	approximately	approximate	arrogance	arrogant	arrested	arrest
appointments	appoint	appraisal	appraise	approximating	approximate	arrogant	arrogant	arrest	arrest
appoint	appoint	appreciate	appreciate	approximation	approximate	arrogantly	arrogant	arresting	arrest
appointment	appoint	appreciatively	appreciate	approximate	approximate	arrogate	arrogate	arrives	arrive
appoints	appoint	appreciating	appreciate	approximations	approximate	arrogates	arrogate	arrived	arrive
appointed	appoint	appreciation	appreciate	appropriated	appropriate	arrogated	arrogate	arrivals	arrive
appointee	appoint	appreciative	appreciate	appropriateness	appropriate	arrogation	arrogate	arriving	arrive
appreciable	appreciate	appreciates	appreciate	architecturally	architect	arose	arise	argument	argue
apprehending	apprehend	appreciated	appreciate	architectures	architect	arises	arise	argumentation	argue
apprehend	apprehend	apprenticeships	apprentice	architects	architect	arisen	arise	arguably	argue
apprehended	apprehend	apprise	apprise	architect	architect	arising	arise	arguments	argue
apprehensions	apprehend	apprising	apprise	architecture	architect	arise	arise	argument	argue
apprehension	apprehend	apprised	apprise	architectural	architect	arithmetically	arithmetic	argumentation	argue
apprehends	apprehend	apprises	apprise	architectonic	architectonic	arithmetic	arithmetic	arrangers	arrange
apprehensively	apprehend	approachability	approach	architectonics	architectonics	arithmetical	arithmetic	arranges	arrange
apprehensible	apprehend	approachable	approach	areas	area	armory	arm	arranger	arrange
apprehensive	apprehend	approach	approach	area	area	armfuls	arm	arranged	arrange
apprenticing	apprentice	approaching	approach	arenas	arena	arms	arm	arrangement	arrange
apprentice	apprentice	approaches	approach	arena	arena	arm	arm	arrangements	arrange
apprenticeship	apprentice	approached	approach	arguing	argue	armful	arm	arranging	arrange
apprenticed	apprentice	approbation	approbate	argues	argue	armed	arm	arrange	arrange
apprentices	apprentice	appropriators	appropriate	arguer	argue	arming	arm	apprise	apprise
apprenticeships	apprentice	appropriate	appropriate	argued	argue	argument	argue	apprising	apprise
arithmetically	arithmetic	achiever	achieve	argue	argue	argumentation	argue	apprised	apprise
arithmetic	arithmetic	achievable	achieve	arguable	argue	arguably	argue	appreciable	appreciate
arithmetical	arithmetic	achieved	achieve	arguers	argue	arguments	argue	apprehending	apprehend

Input-7 : MorphoLEX Dictionary double and triple suffixation Nominalized/Derived Words Verb / Noun / adjective / Adverb POS with their Lemmas which is depicted in Table 5.28.

Table 5.28 Double/Triple Suffixed Nominalized/Derived Words with Lemmas

Number of words 349, Number of lemma 227					
Words	**Lemmas**	**Words**	**Lemmas**	**Words**	**Lemmas**
academicianship	academy	classically	class	educationalist	educate
acoustically	acoustic	collectivization	collect	educationalists	educate
activation	act	colonialism	colon	educationally	educate
actuarially	actuary	colonialist	colon	egalitarian	egalitarian
additionally	addition	colonialists	colon	egoistically	egoist
adventurously	adventure	commercialization	commerce	egotistically	egotist
allegorically	allegory	communicational	communicate	egotistical	egotist
alternatively	alternate	confidentialities	confident	electrically	electric
altruistically	altruist	confidentiality	confident	electronically	electron
anatomically	anatomical	confidentially	confident	emotionality	emotion
anglicanism	anglican	conically	conical	emotionally	emotion
antagonistically	antagonist	conspiratorially	conspirator	emotionalisms	emotion
apocalyptically	apocalyptic	continentally	continent	emotionalism	emotion
architecturally	architect	conversationalist	converse	emotionlessness	emotionless
argumentatively	argument	conversationally	converse	emotionlessly	emotionless
artistically	artist	conversationalists	converse	enthusiastically	enthusiast
authenticator	authentic	cumulatively	cumulate	environmentalism	environment
authentications	authentic	decimalization	decimal	environmentalist	environment
authentication	authentic	decoratively	decor	environmentally	environment
beggarliness	beggar	decorativeness	decor	environmentalists	environment
behaviourally	behaviour	deferentially	deferent	episodically	episode
breathalyser	breathalyse	definitively	definite	equalization	equal
breathalysers	breathalyse	demagogically	demagogy	equalizers	equal
brutalization	brutal	demonically	demon	equatorial	equate
capitalizations	capital	derivatively	derivative	equitably	equity
capitalistic	capital	developmentally	develop	evangelicalism	evangel
capitalization	capital	deviationists	deviate	existentialists	existent
casuistically	casuist	deviationist	deviate	existentialism	existent
catastrophically	catastrophe	deviationism	deviate	existentialist	existent
categorically	category	diagrammatically	diagrammatic	farcically	farce
censoriousness	censorious	dictatorships	dictate	fatalistic	fatal
censoriously	censorious	dictatorship	dictate	fictionalize	fiction
centralization	centre	dictatorial	dictate	fictionalizes	fiction
centralizations	centre	differentiations	differentiate	fictionalized	fiction
certification	certify	differentiation	differentiate	fictionalizing	fiction
certifications	certify	directionally	direct	figuratively	figure
chauvinistically	chauvinist	directionality	direct	educationalist	educate

chemically	chemical	ecclesiastically	ecclesiastic	educationalists	educate
civilisation	civil	economically	economy	educationally	educate
fluoridation	fluoridate	justifiably	justifiably	injuriously	injury
fractionally	fraction	lecherousness	lecher	isolationistic	isolate
futuristically	future	lecherously	lecher	journalistic	journal
generalization	general	lexically	lexicon	juridical	juridic
generalizations	general	liberalization	liberal	localization	local
graphically	graphic	linguistically	linguist	longitudinally	longitude
harmoniousness	harmony	liquidation	liquid	materialistically	matter
harmoniously	harmony	liquidator	liquid	maturational	mature
heretically	heretic	liquidizers	liquid	mechanistically	mechanic
historically	history	liquidizer	liquid	mechanizations	mechanic
hysterically	hysteric	liquidators	liquid	mechanically	mechanic
idealistically	ideal	liquidations	liquid	imperialistic	imperial
identically	identity	lobularity	lobular	memorialized	memory
imaginatively	imagine	incidentally	incident	memorializes	memory
imitatively	imitate	industrialization	industry	memorializing	memory
imitativeness	imitate	inferentially	inferential	melodiously	melody

5.8 Comparative Study

Below mentioned Table 5.29 displays individual derived input-words including their corresponding lemma generated by different standard lemmatizers and also by LemmaQuest.

Table 5.29 Comparative Study between lemmatizers

Seq. No	Noun, Verb, Adjective, Adverb	LemmaQuest Lemmatizer	Stanford Lemmatizer	WordNet Lemmatizer	spaCy-Lemmatizer	Lemmagen	MorphAdorner Morphological Analyzer	CTS's Lemmatizer
	Input-Word	**Lemmas**						
1.	abilities	ability	ability	ability	ability	ability	ability	ability
2.	formulae	formula	formula	formula	formulae	formula	formula	formula
3.	insurance	insure	insurance	insurance	insurance	insurance	insurance	insurance
4.	women	woman	woman	woman	woman	woman	woman	woman
5.	existence	exist	existence	existence	existence	existence	existence	existence
6.	education	educate	education	education	education	education	education	education
7.	employment	employ	employment	employment	employment	employment	employment	employment
8.	government	govern	government	government	government	government	government	government
9.	legislature	legislate	legislature	legislature	legislature	legislature	legislature	legislature
10.	angrily	angry	angrily	angrily	angrily	angrily	angry	angrily
11.	academicals	academy	academicals	academicals	academicals	academicals	academicals	academicals
12.	academics	academy	academics	academics	academics	academics	academic	academics
13.	employee	employ	employee	employee	employee	employee	employee	employee
14.	completely	complete	completely	completely	completely	completely	completely	completely
15.	concern	concern	concern	concern	concern	concern	concern	concern
16.	concert	concert	concert	concert	concert	concert	concert	concert
17.	conclusion	conclude	conclusion	conclusion	conclusion	conclusion	conclusion	conclusion
18.	consideration	consider	consideration	consideration	consideration	consideration	consideration	consideration
19.	considerable	consider	considerable	considerable	considerable	considerable	considerable	considerable
20.	construction	consider	construction	construction	construction	construction	construction	construction
21.	considerably	consider	considerably	considerably	considerably	considerably	considerably	considerably
22.	consideration	consider	consideration	consideration	consideration	consideration	consideration	consideration
23.	creature	create	creature	creature	creature	creature	creature	creature
24.	creation	create	creation	creation	creation	creation	creation	creation
25.	creative	create	creative	creative	creative	creative	creative	creative
26.	critical	critic	critical	critical	critical	critical	critical	critical
27.	criticize	critic	criticize	criticize	criticize	criticize	criticize	criticize
28.	complex	complex	complex	complex	complex	complex	complex	complex
29.	complicated	complicate	complicate	complicate	complicate	complicate	complicate	complicate
30.	complicate	complicate	complicate	complicate	complicate	complicate	complicate	complicate
31.	component	component	component	component	component	component	component	component
32.	computer	compute	computer	computer	computer	computer	computer	computer
33.	concerned	concern	concern	concern	concern	concern	concern	concern
34.	concern	concern	concern	concern	concern	concern	concern	concern
35.	concert	concert	concert	concert	concert	concert	concert	concert
36.	conclusion	conclude	conclusion	conclusion	conclusion	conclusion	conclusion	conclusion
37.	consideration	consider	consideration	consideration	consideration	consideration	consideration	consideration
38.	considerable	consider	considerable	considerable	considerable	considerable	considerable	considerable
39.	construction	consider	construction	construction	construction	construction	construction	construction
40.	considerably	consider	considerably	considerably	considerably	considerably	considerably	considerably

Table 5.30 shows the name of the resources and shows the count of input words, unique words and lemmas for each resource.

Table 5.30 Resources with Word and Lemma Count

Resource	Words	Unique Words	Lemma
Brown Text	377	248	218
DUC Text-1	319	217	198
DUC Text-2	248	196	172
Oxford Dictionary-Set 1	320	320	220
Oxford Dictionary-Set 2	491	491	211
Oxford Dictionary-Set 3	470	470	376
Oxford Dictionary-Set 4	77	177	29
Oxford Dictionary-Set 5	277	277	196
MorphoLEXDictionary-1	418	418	123
MorphoLEXDictionary-2	349	349	227
MorphoLEX Dictionary-3	355	355	121
Nominalized Words-Set 1	282	282	212
Nominalized Words-Set 2	364	364	126
Nominalized Adverb Words	127	127	127

Fig. 5.2 depicts the number of lemma generated from the set of words extracted for the text of DUC & Brown corpus and for morphed word list from dictionary.

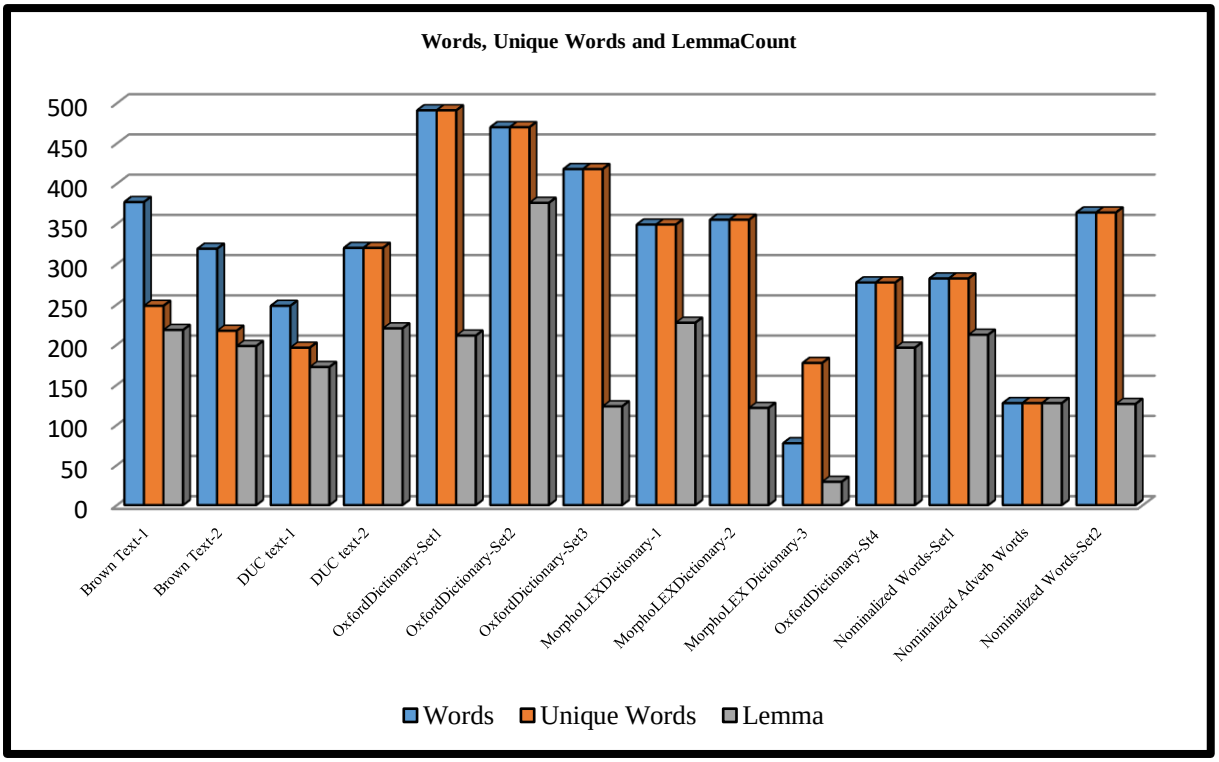

Figure 5.2 Bar Diagram for Word and Lemma Count

Table 5.31 shows the count of input-words, unique words of different resources and different error count generated by individual lemmatizer.

Table 5.31 Word, Lemma Count with Error Count

Resources	**MorphoLEX Dictionary**	**MorphoLEX Dictionary**	**Nominalized Dictionary Words**	**Oxford Dictionary**	**Oxford Dictionary**
Words	418	355	282	177	277
Unique Words	418	355	282	177	277
LemmaQuest : Number-of-Lemmas Generation	123	121	212	29	196
Number of Errors	4	5	2	4	4
Stanford Lemmatizer: Number-of- Lemmas Generation	418	355	282	177	277
Number of Errors	418	355	282	177	277
Python NLTK Lemmatizer : Number of Lemmas Generation	418	355	282	177	277
Number of Errors	418	355	282	177	277
spaCy Word Lemmatizer: Number of Lemmas Generation	418	355	282	177	277
Number of Errors	418	355	282	177	277

Fig.5.3 depicts the number of errors occurred while generating lemma by different existing Lemmatizers and by LemmaQuest. It is clearly observed that error generated by LemmaQuest is far less compared to those generated by others. Thus, LemmaQuest generates maximum number of correct lemmas.

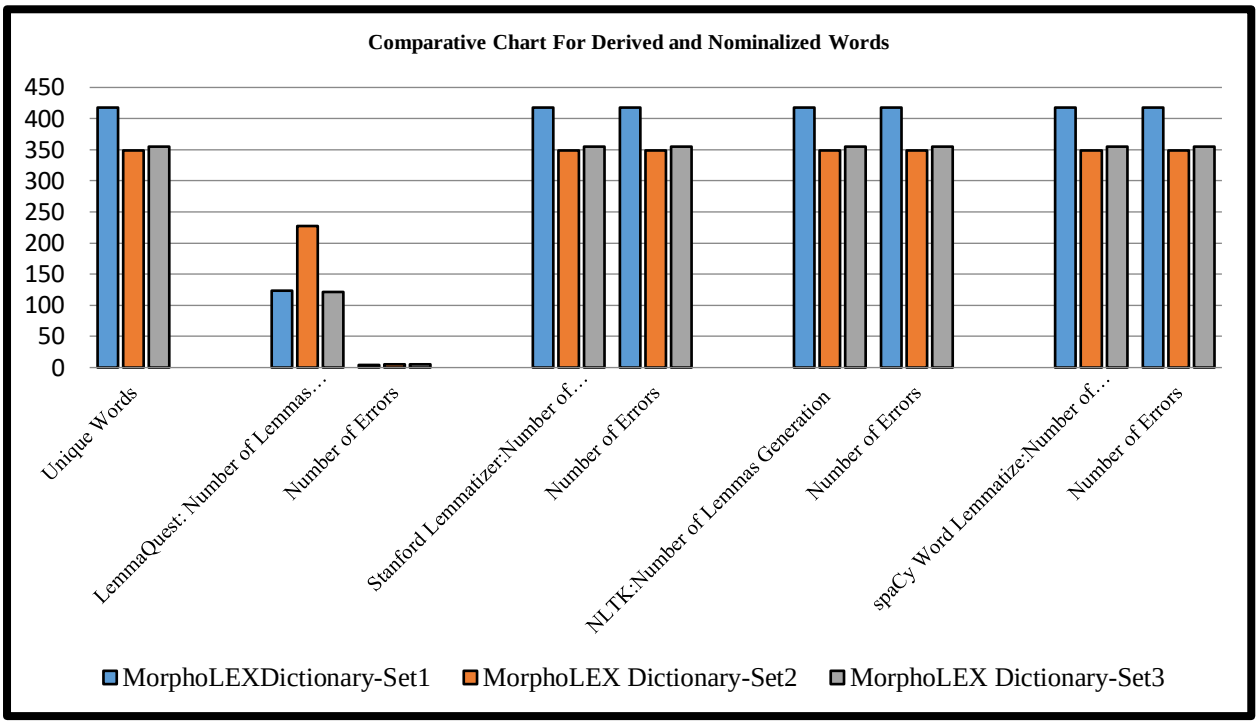

Figure 5.3 Comparative Diagram of Error Count of Different Lemmatizers

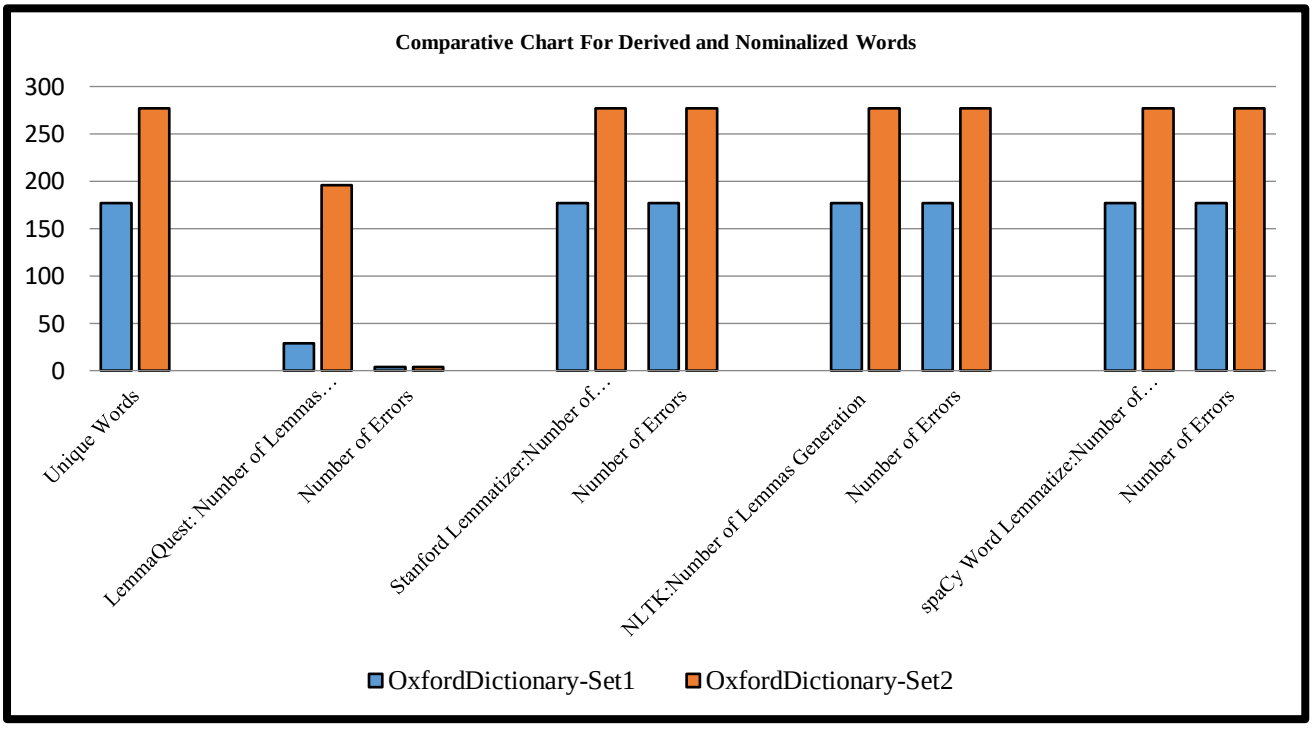

Figure 5.4 Comparative Diagram of Error Count of Different Lemmatizers

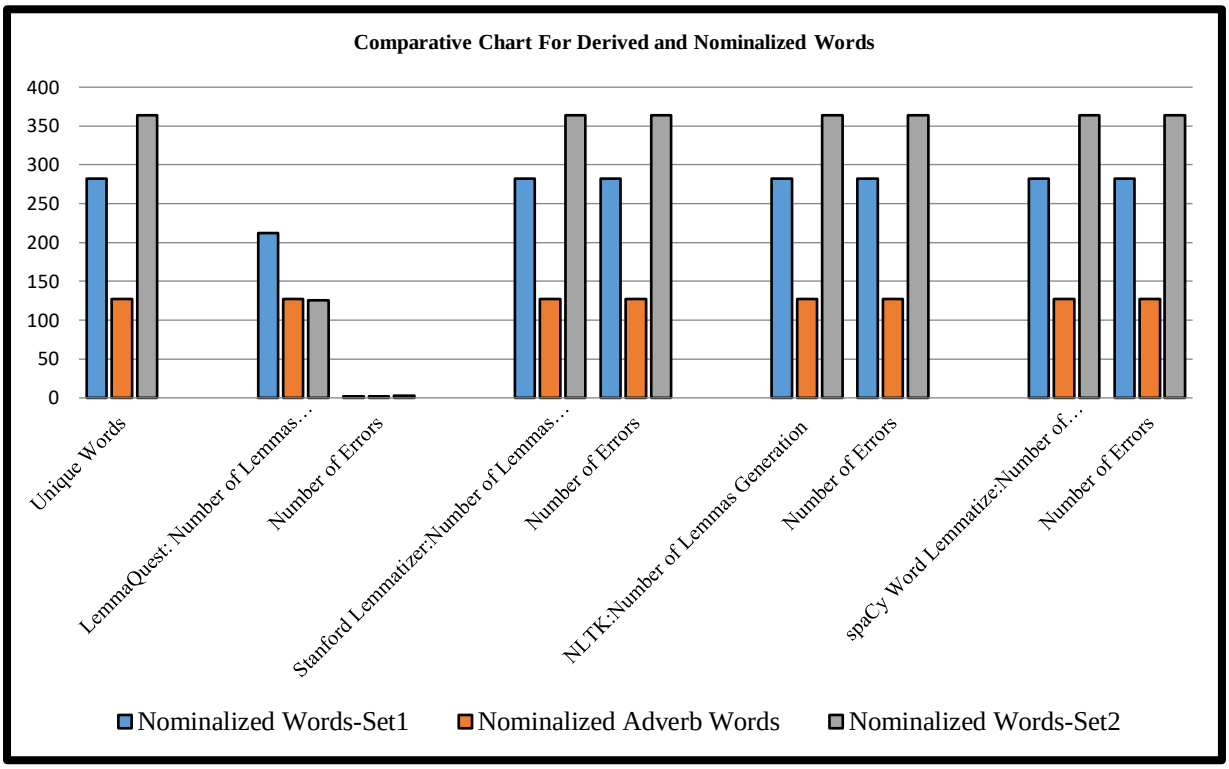

Figure 5.5 Comparative Diagram of Error Count of Different Lemmatizers

5.9 LemmaQuest vs. LemmaChase

- LemmaChase model handles WordNet dictionary single-suffixed derived words and some of the double suffixed words, whereas LemmaQuest handles double (-ment, -ive), triple suffixed derived words (-ative, -ness) and also non WordNet-dictionary derived words (argumentatively, argumentative, attractivenesses) to generate their correct lemma.
- LemmaChase processes each word of any input text file at individual word level, whereas LemmaQuest processes words at group level after creating groups of allied morphed words. One single lemma would be representative for all the allied morphed words of a group in this model. So processing iterations will be minimized as compared to LemmaChase.
- In the LemmaChase model, there is no chance of missing any input word for finding its lemma, whereas in LemmaQuest model, there is a high chance of some input words getting wrongly grouped together. These semantically un-related words like ‘policy/police’, ‘university/universal’, ‘Algebra/Algeria’ get grouped together here.

Based on the knowledge of POS of all such wrongly grouped un-related words, processing of words will be done at individual level through morphological parsing to generate the correct lemma.

- LemmaChase model is beneficial for dictionary based morphed input word list where maximum words are morphologically discrete, whereas LemmaQuest model is beneficial for input text in which a large number of allied morphed words co-exist.
- LemmaChase is effective for Keyword-extraction, Information Retreival (links to sites where information related to that word is available), whereas LemmaQuest is more effective for Text-summarization, topic identification.

  For Information Retrieval, if the input is a text or a sentence with allied morphed words, then the LemmaQuest is the best option.

Table 5.32 shows word count, lemma count and error count of generation in lemma by LemmaQuest, LemmaChase and Stanford lemmatizers.

Table 5.32 Comparative Output of LemmaQuest vs. LemmaChase vs. Stanford Lemmatizer

Resource	Words	Unique Words		LemmaQuest: Number of Lemmas Generation	Number of Errors		LemmaChase : Number of Lemmas Generation	Number of Errors		Stanford Lemmatizer: Number of Lemmas Generation
MorphoLEX Dictionary-double/triple suffixation words	349	349		227	10		349	150		349
MorphLEX-Set-2	320	320		219	20		250	150		320

Fig. 5.14 shows comparative bar diagram of LemmaQuest, LemmaChase and Stanford Lemmatizers in generation of lemma for double/triple suffixed derived words.

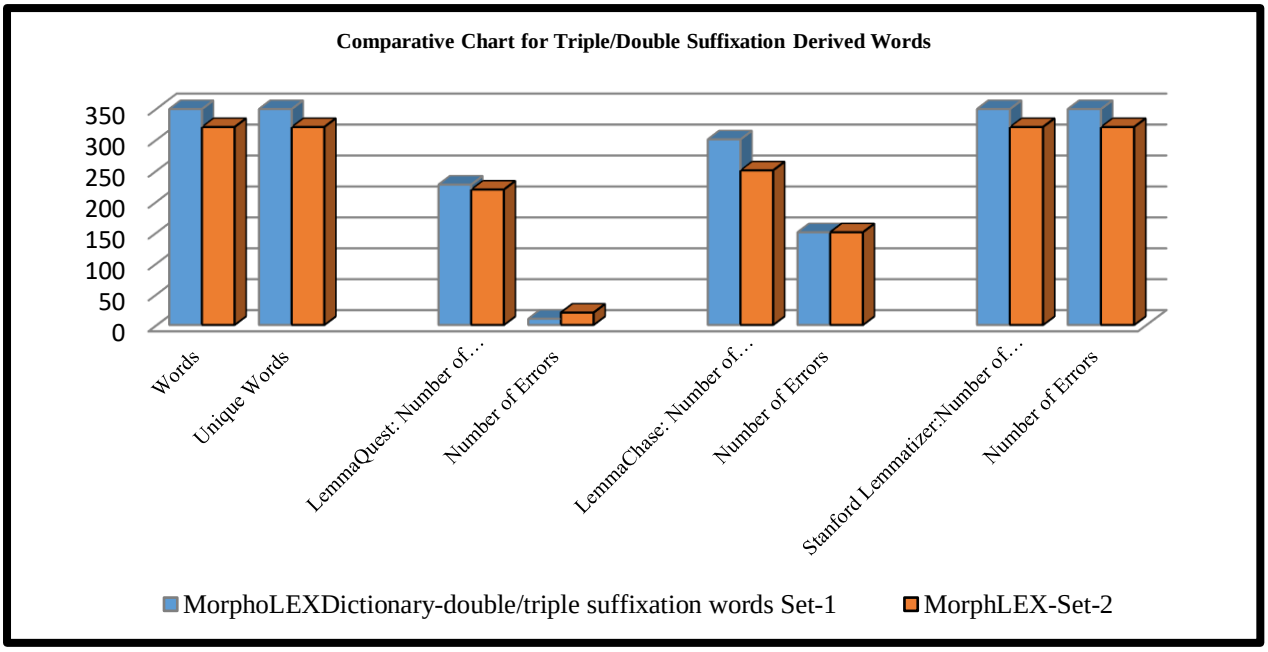

Figure 5.6 Comparative Bar Diagram of LemmaQuest, LemmaChase and Stanford Lemmatizer

Table 5.33 shows the accuracy of LemmaQuest model based on the value of Precision and F-Score. Stanford and all other lemmatizers do not identify lemmas for single, double and triple suffixed derived words. Hence their precision, recall and F-Score will be **zero.**

Table 5.33 Precision, Recall and F-Score value for LemmaQuest

LemmaQuest Model	Number of Words	Number of Unique Words	Number of Lemma Generation	Number of Incorrect Lemma Generation	Rate of Error	Precision	Recall	F-Score 2(P*R)/(P+R)
MorphoLEX Dictionary-Double/Triple Suffixation Words Set 1	349	349	227	10	2.86	339/(339+10)=0.97	1	0.98
MorphLEX- Double/Triple Suffixation Words Set-2	320	349	219	20	6.25	300/(300+20)=0.93	1	0.96
MorphoLEX Dictionary-Words Set 3	418	418	123	4	0.95	414/(414+4)=0.99	1	0.99
MorphoLEX Dictionary-Words Set- 4	355	355	121	5	1.40	350/(350+5)=0.98	1	0.98
Nominalized Dictionary Words Set 1	282	282	212	2	0.70	280/(280+2)=0.99	1	0.99
Oxford Dictionary-Words Set-1	177	177	29	4	2.25	173/(173+4)=0.97	1	0.98
Oxford Dictionary-Words Set-2	277	277	196	4	1.44	273/(273+4)=0.98	1	0.98

Fig 5.7 shows the bar diagram for Table 5.33.

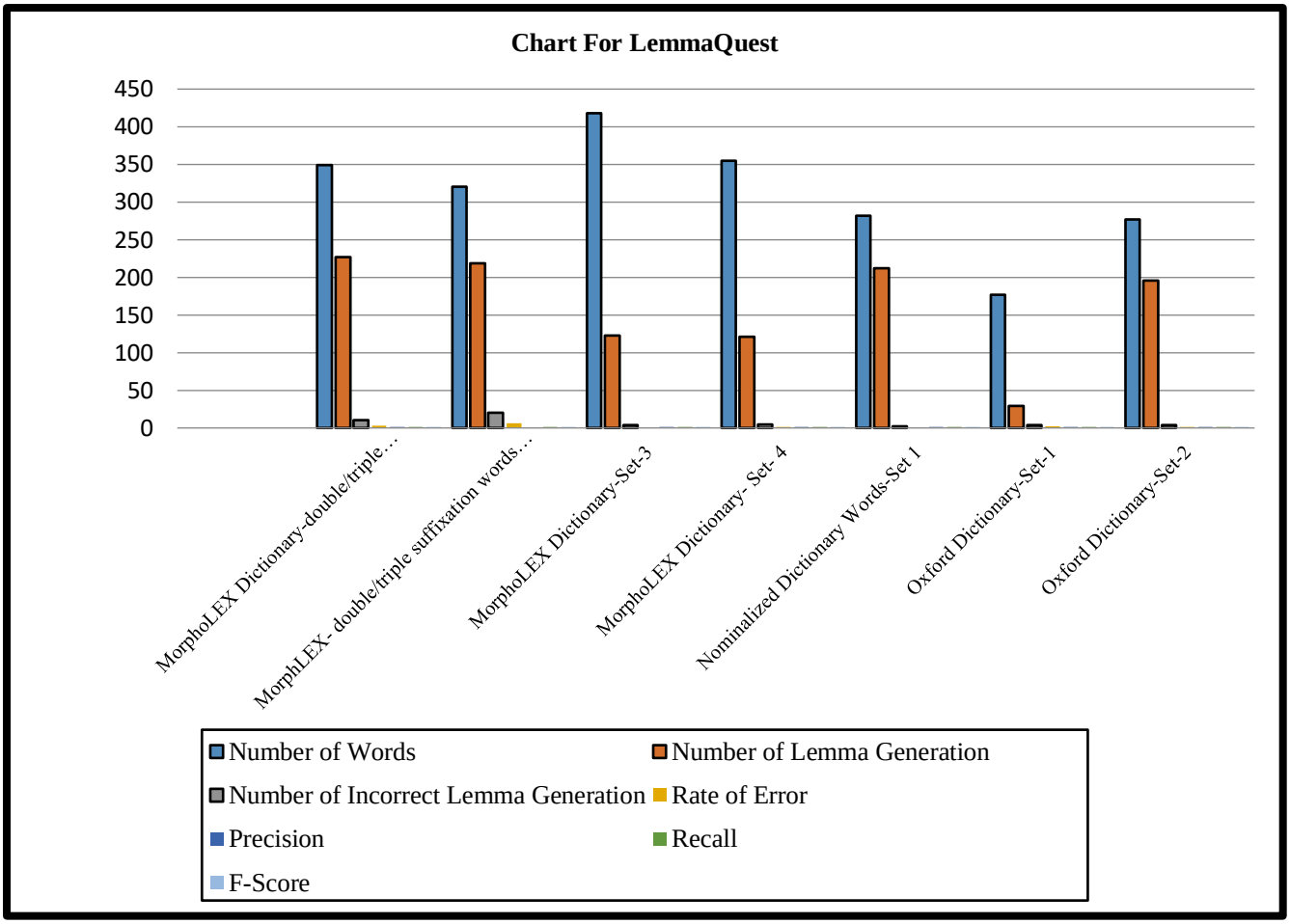

Figure 5.7 Bar Diagram for Precision, Recall and F-Score of LemmaQuest Model

5.10 Conclusion and Summary

LemmaQuest handles the maximum number of English derived morphed words and nominalized words, along with inflected words to extract their corresponding lemma. When a text file or dictionary morphed word list is taken as an input, lemma extraction processing is not required to be applied on individual words. For a group of all allied morphed words, a single run for extraction of the lemma is required. One single lemma would be representative for all the allied morphed words of a group in this model.

LemmaQuest model manages most of the semantically un-related words by clustering them in a single group but with the knowledge of POS of that word and also based on parsing of morphological structure of that word, the correct lemmas are generated. Thus the proposed model gives a finer, better and correct output.

The proposed LemmaQuest lemmatizer model would be highly efficient for applications related to Information Extraction, Text simplification, Text Summarization,

Keyword Extraction, etc. to name a few. The efficacy and precision of all these NLP and text-mining applications can be remarkably improved when the correct and minimum numbers of lemmas are fetched from the derived morphed and nominalized words.

This lemmatizer aims to reduce the number of lemmas generated from the input by correctly identifying the derived morphed and nominalized words and in turn refining the accuracy of Machine Translation System[1]. In this way the proposed model- LemmaQuest will work as a pre-requisite task for all the above mentioned applications. This model has been published as a paper titled "LemmaQuest Lemmatizer: A Morphological Analyzer Handling Nominalization" in the IETE Journal of Research by Taylor and Francis publication.

[1] Y.-L. Yeong, T.-P. Tan, and S. K. Mohammad, Using dictionary and lemmatizer to improve low resource English-Malay statistical machine translation system,