Chapter 3

Comparative Study of Stemmers and Lemmatizers

3.1 Introduction

In continuation of the literature study conducted and discussed in the previous chapter, this chapter summarizes the detailed observation and conclusion of that study. For this a comparative algorithm was developed for analyzing the output and calculating the execution time required by popular stemmers. This is followed by an exhaustive comparative analysis of the working of existing available lemmatizers. Basically this chapter too is related to literature study and discusses the analysis done which eventually led to identifying the Research Gap.

3.2 Algorithm for Comparative Study of Stemmers

The literature study on stemmers and their output was basically studied in detail after referencing a number of papers, books and related material on the internet. However to actually analyze the output of the popular stemmers and compare their output, it was very crucial and important to implement these algorithms also. This initial part of the research work hence was related to designing and implementing the stemming algorithms and generating the comparative output. This was developed in JAVA for different datasets from DUC text (Document Understanding Conferences)[1]and dictionary morphed words. The steps of the algorithm are given below:

1. Punctuation, numeric values, stop words and proper nouns are removed after tagging by Stanford Part-of-Speech software tool and then the filtered collection of words are used as an input for this comparative algorithm. For any text, duplicate words are also initially removed.

[1]DUC text. http://www-nlpir.nist.gov/projects/duc/data.html

2. Filtered sorted unique words are taken as input for the Lovins, Porter, Paice and KStem stemming techniques. For each filtered word of DUC text and dictionary text, this algorithm generates the list of output for these stemmers

Features of affix-removal stemming approaches based on execution of above algorithm:

Table 3.1 discusses the advantage and disadvantages of different affix-removal stemming algorithms.

Table 3.1 Advantage and Disadvantage of Affix-Removal Stemmers

Features of Affix Removal Stemmers			
Lovins	**Porter**	**Paice**	**KStemming**
Advantage			
Most of the irregular singular-plural morphed words are correctly stemmed here. It is observed that average computational time (in milliseconds) to generate stem token is lowest with compare to other affix removal stemmers.	Size of the vocabulary / words is reduced by one third with the help of this suffix stripping stemmer.	ICF (Index Compression factor) of Paice is quite high with compare to other affix removal stemmers. Paice algorithm is observed as a stronger stemmer with compare to others.	KStem generates root- dictionary words instead of generating meaningless stem token for morphed input word in many cases.
Disadvantage			
It fails to identify valid suffix for generating stem in some cases.	Stems generated from morphological variants are not always real words.	Stems generated from morphological variants are not always real words.	It is inefficient in case of large documents.

Note: Index Compression Factor:

"N = Number of unique words before stemming, S = Number of unique stems after stemming, ICF = Index Compression factor; Calculation of ICF = (N-S)/N; Greater the value of ICF, greater will be strength of the Stemmer." (Paice Chris D, 1994)

3.3 Comparative Output of Stemmers

Table 3.2 shows set of dictionary morphed words with their stem tokens.

Table 3.2 Number of Stem Word Generation for Set of Wordlist

List of Morphed Words	Number of Stemmed Tokens				List of Morphed Words	Number of Stemmed Tokens			
	Porter	Lovins	Paice	KStemm		Porter	Lovins	Paice	KStemmer
Abilities Ability	1	1	1	2	Abolish, Abolished Abolishes Abolishing Abolition	2	3	2	2
Abnormal Abnormally	1	1	1	1					

Table 3.3 shows number of stem tokens which are generated from different set of morphed words.

Table 3.3 Number of Stem Word Generation for Set of Wordlist (Elaboration of Table 3.2)

Sequence	Morphed Words	Porter	Number of Stem	Lovins	Number of Stem	Paice	Number of Stem	KStemmer	Number of Stem
1	abilities	abil	1	Abil	1	abl	1	abilities	2
	ability	abil		Abil		abl		ability	
2	abnormal	abnorm	1	Abnorm	1	abnorm	1	abnormal	1
	abnormally	abnorm		Abnorm		abnorm		abnormal	
3	abolish	abolish	2	Abol	3	abol	2	abolish	2
	abolished	abolish		Abol		abol		abolish	
	abolishes	abolish		Abolish		abol		abolish	
	abolishing	abolish		Abolish		abol		abolish	
	abolition	abolit		Abolit		abolit		abolition	
4	abort	abort	1	Abort	1	abort	1	abort	3
	aborts	abort		Abort		abort		abort	
	aborted	abort		Abort		abort		abort	
	aborting	abort		Abort		abort		abort	
	abortional	abort		Abort		abort		abortion	
	abortive	abort		Abort		abort		abortive	
	abortively	abort		Abort		abort		abortive	
5	abound	abound	1	Abound	1	abound	1	abound	1
	abounded	abound		Abound		abound		abound	
	abounding	abound		Abound		abound		abound	
	abounds	abound		Abound		abound		abound	
6	abrupt	abrupt	2	Abrupt	1	abrupt	1	abrupt	1
	abruptly	abruptli		Abrupt		abrupt		abrupt	
	abruptness	abrupt		Abrupt		abrupt		abrupt	
7	abscond	abscond	1	Abscons	2	abscond	1	abscond	1
	absconded	abscond		Abscons		abscond		abscond	
	absconders	abscond		Absconder		abscond		abscond	
	absconding	abscond		Abscons		abscond		abscond	
	absconds	abscond		Abscons		abscond		abscond	
8	absence	absenc	5	Abs	4	abs	2	absence	5
	absences	absenc		Abs		abs		absence	
	absent	absent		Absent		abs		absent	
	absented	absent		Absent		abs		absent	
	absentee	absente		Absente		abs		absentee	
	absentees	absente		Absentee		abs		absentee	
	absenting	absent		Absent		abs		absent	
	absently	absent		Abs		abs		absently	
9	absents	absent		Absent		abs		absent	
	absentia	absentia		Absent		absent		absentia	

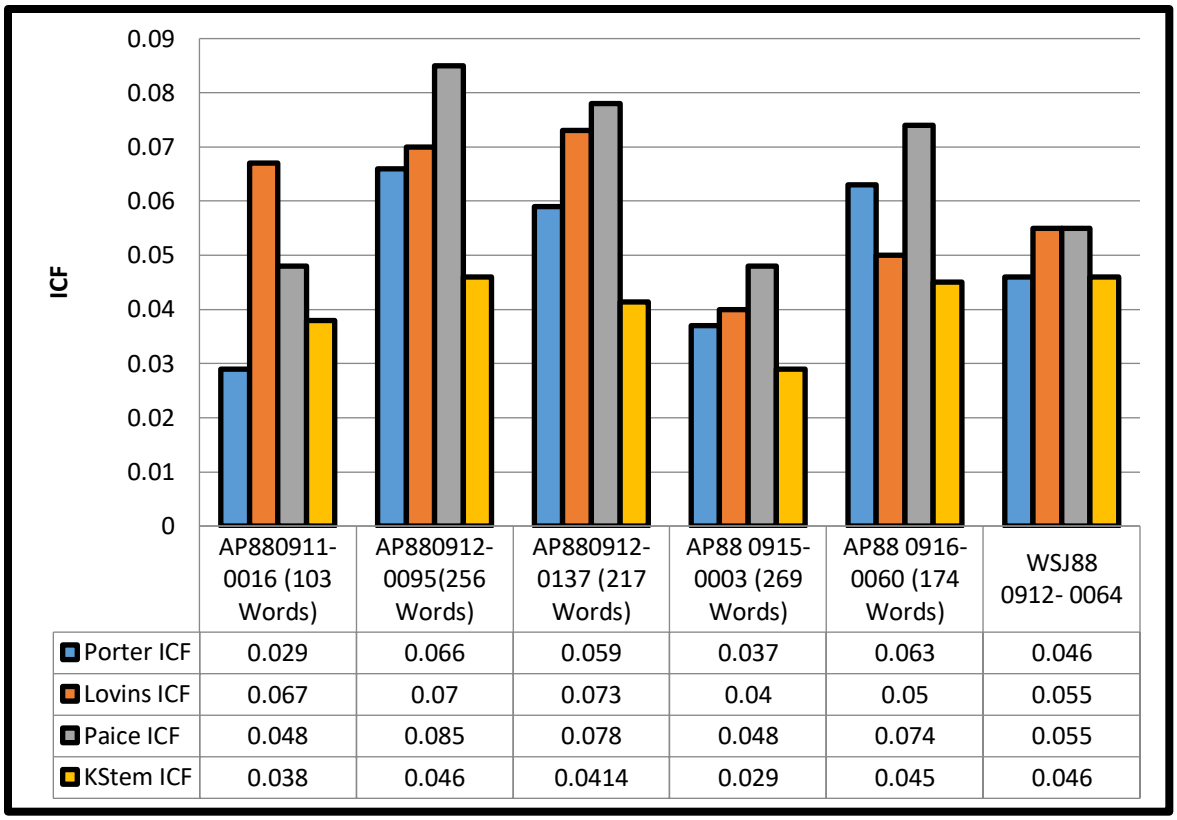
Sequence	Morphed Words	Porter	Number of Stem	Lovins	Number of Stem	Paice	Number of Stem	KStemmer	Number of Stem
10	alcohol	alcohol	1	Alcohol	1	alcohol	1	alcohol	3
	alcoholic	alcohol		Alcohol		alcohol		alcoholic	
	alcoholics	alcohol		Alcohol		alcohol		alcoholic	
	alcoholism	alcohol		Alcohol		alcohol		alcoholism	
	alcohols	alcohol		Alcohol		alcohol		alcohol	
11	alert	alert	2	Alers	1	alert	1	alert	1
	alerted	alert		Alers		alert		alert	
	alerting	alert		Alers		alert		alert	
	alertly	alertli		Alers		alert		alert	
	alertness	alert		Alers		alert		alert	
	alerts	alert		Alers		alert		alert	
12	algebra	algebra	1	Algebra	1	algebr	2	algebra	1
	algebraic	algebra		Algebra		algebra		algebra	
	algebras	algebra		Algebra		algebra		algebra	
13	Algeria	algeria	2	Alger	1	alger	2	algeria	1
	Algerian	algerian		Alger		alg		algeria	
	Algerians	algerian		Alger		alg		algeria	

Table 3.4 shows empirical result which analyses the degree of strength of popular stemmers with respect to correctness in generation of stem and computation time.

Table 3.4 Comparative Result of Affix-Removal Stemmers

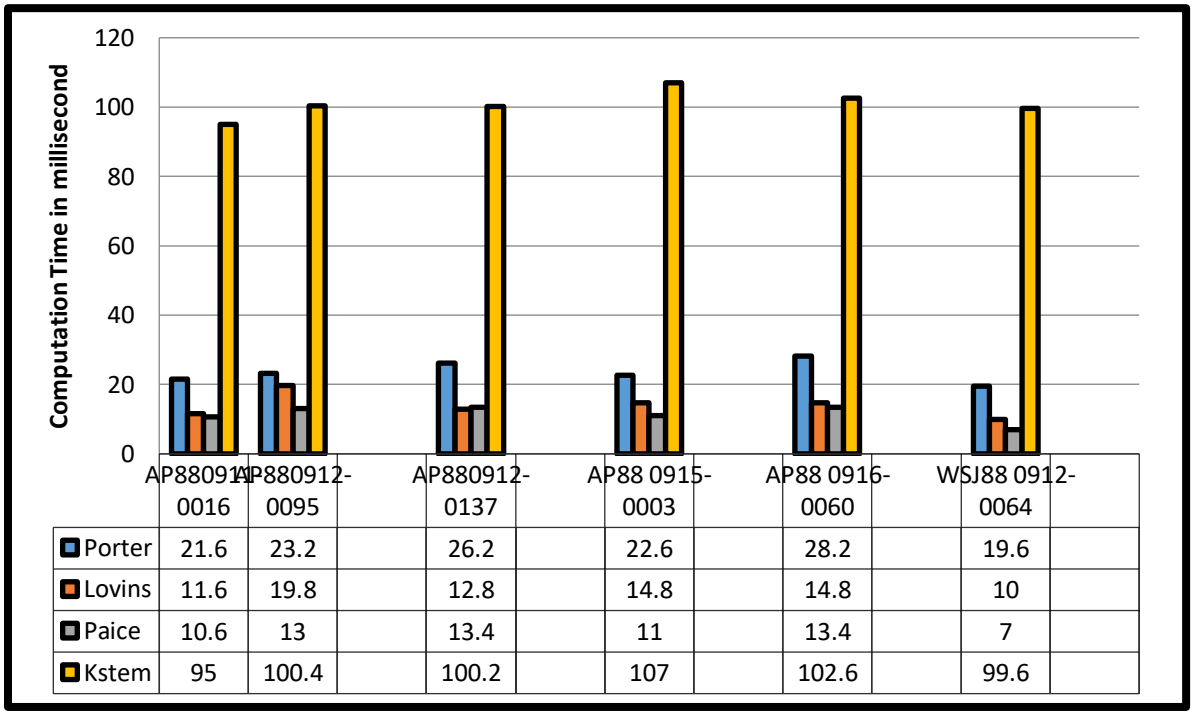
Input Text	Actual Size	Effective Size	Porter		Lovins		Paice		Krovetz (KStem)	
			ICF	Time	ICF	Time	ICF	Time	ICF	Time
AP880911-0016 [DUC]	322	103	0.029	21.6	0.067	11.6	0.048	10.6	0.038	95
AP880912-0095 [DUC]	782	256	0.066	23.2	0.07	19.8	0.085	13	0.046	100.4
AP880912-0137 [DUC]	666	217	0.059	26.2	0.073	12.8	0.078	13.4	0.0414	100.2
AP88 0915- 0003 [DUC]	1080	269	0.037	22.6	0.040	14.8	0.048	11	0.029	107
AP88 0916- 0060 [DUC]	527	174	0.063	28.2	0.05	14.8	0.074	13.4	0.045	102.6
WSJ88 0912-0064 [DUC]	362	108	0.046	19.6	0.055	10	0.055	7	0.046	99.6

Fig. 3.1 and Fig. 3.2 shows the comparative Bar diagram of ICF values which are calculated in Table 3.4 for affix-removal stemmers.



	AP880911-0016 (103 Words)	AP880912-0095(256 Words)	AP880912-0137 (217 Words)	AP88 0915-0003 (269 Words)	AP88 0916-0060 (174 Words)	WSJ88 0912- 0064
Porter ICF	0.029	0.066	0.059	0.037	0.063	0.046
Lovins ICF	0.067	0.07	0.073	0.04	0.05	0.055
Paice ICF	0.048	0.085	0.078	0.048	0.074	0.055
KStem ICF	0.038	0.046	0.0414	0.029	0.045	0.046

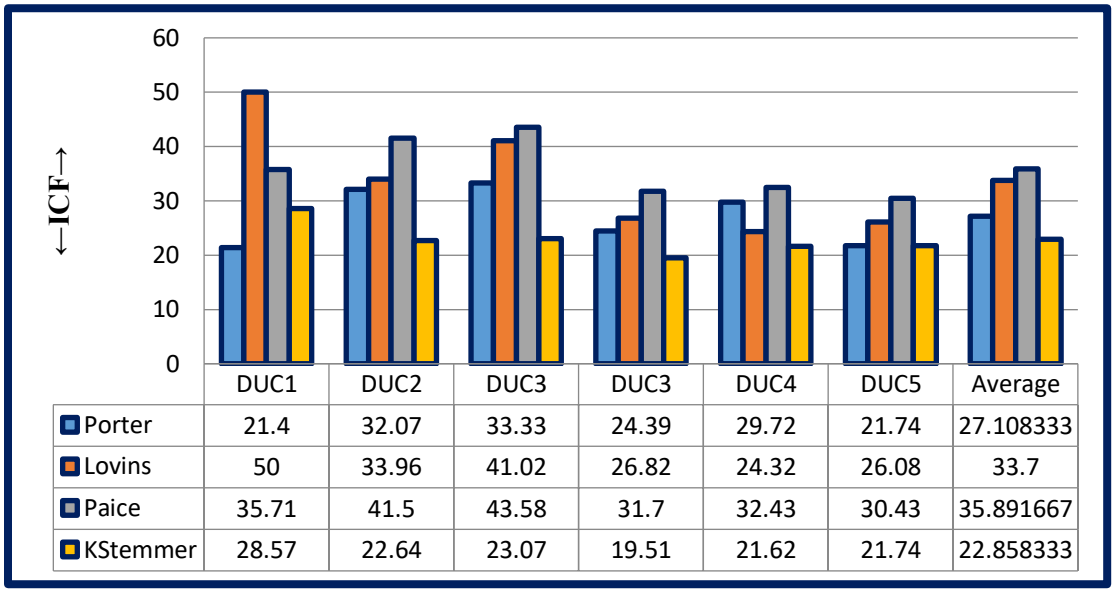
Figure 3.1 Comparative Chart of ICF values for DUC Set1



	AP880911-0016	AP880912-0095		AP880912-0137		AP88 0915-0003		AP88 0916-0060		WSJ88 0912-0064	
Porter	21.6	23.2		26.2		22.6		28.2		19.6	
Lovins	11.6	19.8		12.8		14.8		14.8		10	
Paice	10.6	13		13.4		11		13.4		7	
Kstem	95	100.4		100.2		107		102.6		99.6	

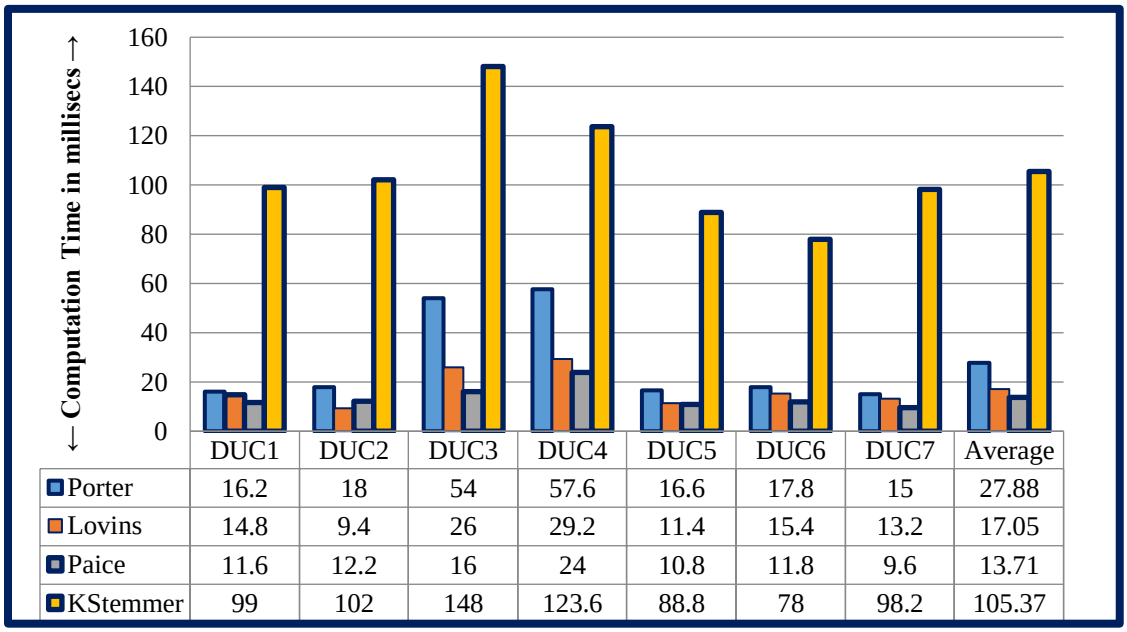
Figure 3.2 Comparative Chart of Computational Time for DUC Set1

Fig 3.3 and Fig. 3.4 shows comparative Bar diagram of strength (ICF) of each stemmer and computation time of each stemmer.



	DUC1	DUC2	DUC3	DUC3	DUC4	DUC5	Average
Porter	21.4	32.07	33.33	24.39	29.72	21.74	27.108333
Lovins	50	33.96	41.02	26.82	24.32	26.08	33.7
Paice	35.71	41.5	43.58	31.7	32.43	30.43	35.891667
KStemmer	28.57	22.64	23.07	19.51	21.62	21.74	22.858333

Figure 3.3 Comparative Chart of ICF for DUC Set2



	DUC1	DUC2	DUC3	DUC4	DUC5	DUC6	DUC7	Average
Porter	16.2	18	54	57.6	16.6	17.8	15	27.88
Lovins	14.8	9.4	26	29.2	11.4	15.4	13.2	17.05
Paice	11.6	12.2	16	24	10.8	11.8	9.6	13.71
KStemmer	99	102	148	123.6	88.8	78	98.2	105.37

Figure 3.4 Comparative Chart of Computational Time for DUC Set2

3.4 Popular Lemmatization Tools

After implementing and comparing the stemming techniques, the next implementation required was empirically comparing the output of the popular lemmatizers. Most of the lemmatizers are black box implementations and the tools to use them are freely available. Using the available tools and for the same DUC datasets, the lemmas generated as output were analyzed and summarized.

Brief description of the lemmatization tools is given below.

1. **MorphAdorner:** It is a command-line Java application which executes as a pipeline manager for processes performing morphological adornment of words in a text. It lemmatizes inflected words into their corresponding lemma[2].
2. **LemmaGen:** It was trained to generate accurate and efficient lemmatizers for twelve different languages. Its analysis on the corresponding lexicons depicts that LemmaGen outperforms the lemmatizers generated by two alternative approaches, RDR and CST, both in terms of accuracy and efficiency (MatjaˇzJurˇsiˇc, Igor Mozetiˇc, 2010). However Lemmagen only lemmatizes inflected words.
3. **spaCy Lemmatizer:** It comes as a pre-built model of NLP tool that can parse text and compute various NLP-related features through one single function call[3].
4. **UDPipe:** It is a trainable pipeline, executing for tagging, lemmatization and dependency parsing of CoNLL-U files. UDPipe is a language-agnostic tool (Milan Straka and Jana Strakova, 2017).
5. **Stanford Lemmatizer:** Stanford morphological analyzer (LemmaProcessor) was developed based on Kimmo & Goldsmith's approaches which are already discussed in the previous chapter. Stanford NLP research group developed CoreNLP for natural language processing in Java. CoreNLP facilitates users to acquire linguistic annotations for text, including token and sentence boundaries, parts of speech, named entities, lemmatization, dependency and constituency parses, coreference, sentiment, quote attributions, and relations. The centerpiece of CoreNLP is the pipelines which take raw

[2] http://morphadorner.northwestern.edu/morphadorner/
[3] http://textanalysisonline.com/spacy-word-lemmatize

text as an input. Fig. 3.5 depicts the flow of CoreNLP (Christopher D. Manning, 2005).

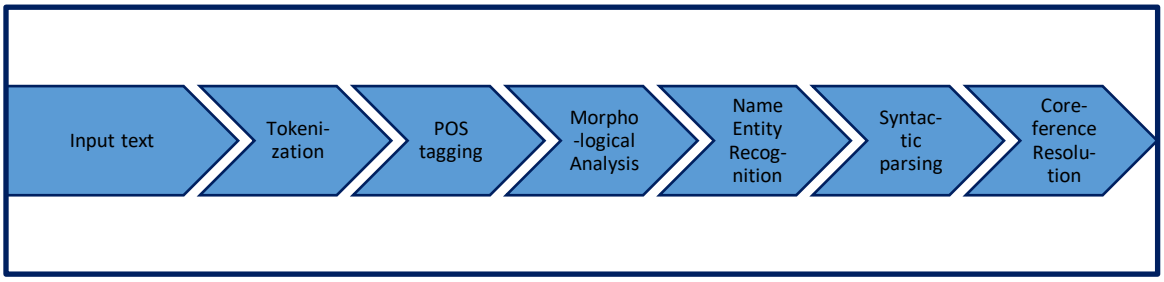


Figure 3.5 **Stanford CoreNLP Model**

6. **BioLemmatizer**: It is also a lemmatization tool which handles bio-medical terms as well as some derivational adjectives and adverbs, based on a morphological analyzer, MorphAdorner 2.0 (Haibin Liu, Tom Christiansen, 2012).
7. Other popular available online Lemmatizers are Wordnet Lemmatizer with NLTK and CST's online tools.

3.5 Comparative Output of Existing Lemmatizers

After execution of the lemmatization tools for the varied datasets, it was noticed that they are able to generate the correct lemma for all inflected words but are not able to generate lemma for derived and nominalized words. BioLemmatizer is mostly able to handle only biological terms. Table 3.6 shows the set of lemmas generated by different available lemmatizers for set of derived and nominalized words.

It became evident after actually implementing the stemmers and lemmatizers with a variety of input datasets that none of them were able to give the correct lemmas for specific words. These words were basically the derived and nominalized words. In English grammar, nominalization is a type of word construction in which a verb or an adjective-POS-word (or another part of speech) is used as (or transformed into) a noun -POS-word. The verb-POS-word is nominalize[4]. It is also called nouning. The derivational affix is added with Verb-POS-word to convert it to a noun-POS-word. e.g. the noun construction/destruction has been produced from the verb construct/destruct. These words are being treated as separate independent words in the English dictionary without connecting them to their actual root words or lemmas. Table 3.5 shows the output for nominalized/derived words by different lemmatizers.

[4] https://www.thoughtco.com/nominalization-in-grammar-1691430

Table 3.5 Output for Nominalized/Derived Words by Existing Online Lemmatizers

Existing Available Lemmatization online tool

1. LemmaGen Lemmatizer : http://lemmatise.ijs.si/Services

Input: "Acceptance achievement Action activity achievement judgment addition adjustment admiration amazement annoyance attention"

Output: "Acceptance achievement Action activity achievement judgment addition adjustment admiration amazement annoyance" [No Change]

2. BioLemmatizer

Input		Output
anlagen,	NN :	anlage
spermatogonia,	NN :	spermatogonium

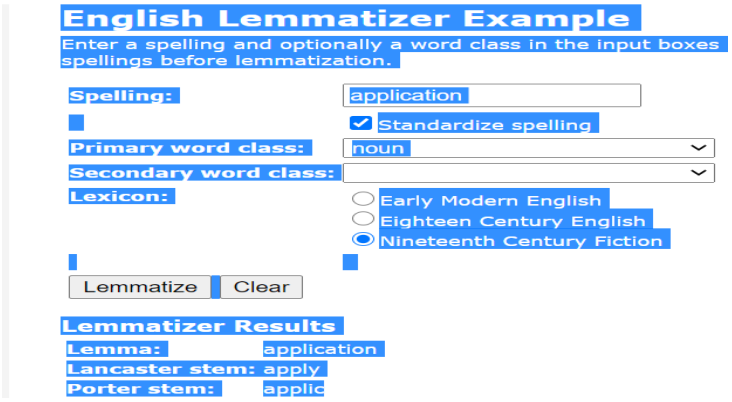
3. Stanford CoreNLP XML Output

Sentences

Id	Word	Lemma	Char begin	Char end	POS
1	Acceptance	acceptance	0	10	NN
2	Achievement	achievement	12	23	NN
3	Action	action	25	31	NN
4	Activity	activity	33	41	NN
5	Activeness	activeness	43	53	NN

Sentence #1 *Tokens*

4 MorphAdorner



Text Analysis Result

5. spaCy Word Lemmatize: [5]

Original Text	Analysis Result
act acted acting action actionable actions activate activated activates activating activation activator activators active actively activeness activism activisms activist activists activities activity actor actors actress adequacy adequate adequately adhere adhered adherence adherent adherents adheres adhering adhesion adhesional adhesions adhesive adhesively adhesives	act act act action actionable action activate activate activate activate activation activator activator active actively activeness activism activism activist activist activity activity actor actor actress adequacy adequate adequately adhere adhere adherence adherent adherent adhere adhere adhesion adhesional adhesion adhesive adhesively adhesive

6. NLTK-WordNet Lemmatizer: [6]

Original Text : actor actors actress adequacy adequate adequately adhere adhered adherence adherent adherents adheres adhering adhesion adhesional adhesions adhesive adhesively adhesives adjoin adjoined adjoining adjoins adjourn adjoumed adjourning adjourns adjust adjustable adjusted adjuster adjusters adjusting adjustment adjusts adjutant adjutants administer admirable admirably admiration admire admired admirer admirers admires admiring admiringly admiral admirals admiralty admissible admission admissions admissive admit admits admitted admittedly admitting admittance

Analysis Result

actor actor actress adequacy adequate adequately adhere adhere adherence adherent adherent adhere adhere adhesion adhesional adhesion adhesive adhesively adhesives adjoin adjoin adjoin adjoins adjoum adjourn adjourn adjourns adjust adjustable adjusted adjuster adjuster adjust adjustment adjusts adjutant adjutant administer admirable admirably admiration admire admire admirer admirer admire admire admiringly admiral admiral admiralty admissible admission admission admissive admit admits admit admittedly admit admittance

7. Seohorsesesence Lemmatizer: [7]

Analysis Result

actor actor actress adequacy adequate adequately adhere adhere adherence adherent adherent adhere adhere adhesion adhesional adhesion adhesive adhesively adhesives adjoin adjoin adjoin adjoins adjoum adjourn adjourn adjourns adjust adjustable adjusted adjuster adjuster adjust adjustment adjusts adjutant adjutant administer admirable admirably admiration admire admire admirer admirer admire admire admiringly admiral admiral admiralty admissible admission admission admissive admit admits admit admittedly admit admittance

8. CST's online tools:[8]

Analysis Result

actor/NN/actor actors/NNS/actor actress/NN/actress adequacy/NN/adequacy adequate/JJ/adequate adequately/RB/adequately adhere/VB/adhere adhered/VBN/adhere adherence/NN/adherence adherent/JJ/adherent adherents/NNS/adherent adheres/VBZ/adhere adhering/VBG/adhere adhesion/NN/adhesion adhesional/JJ/adhesional adhesions/NNS/adhesion adhesive/NN/adhesive adhesively/JJ/adhesively adhesives/NNS/adhesive adjoin/VBP/adjoin adjoined/VBD/adjoin adjoining/VBG/adjoin adjoins/VBZ/adjoin adjourn/VBN/adjourn adjourned/VBD/adjourn adjouming/NN/adjourning adjourns/VBZ/adjourn adjust/VB/adjust adjustable/JJ/adjustable adjusted/VBN/adjust adjuster/NN/adjuster adjusters/NNS/adjuster adjusting/VBG/adjust adjustment/NN/adjustment adjusts/VBZ/adjust adjutant/NN/adjutant adjutants/NNS/adjutant administer/VB/administer admirable/JJ/admirable admirably/RB/admirably admiration/NN/admiration admire/VB/admire admired/VBN/admire admirer/NN/admirer admirers/NNS/admirer admires/VBZ/admire admiring/VBG/admire admiringly/RB/admiringly admiral/JJ/admiral admirals/NNS/admiral admiralty/VBP/admiralty admissible/JJ/admissible admission/NN/admission admissions/NNS/admission admissive/JJ/admissive admit/VB/admit admits/VBZ/admit admitted/VBN/admit admittedly/RB/admittedly admitting/VBG/admit admittance/NN/admittance

9. TwinWord Lemmatizer: [9]

[5] https://textanalysisonline.com/spacy-word-lemmatize
[6] https://textanalysisonline.com/nltk-wordnet-lemmatizer
[7] https://seohorsesense.com/free/lemmatization.php
[8] https://cst.dk/tools/index.php#output

Analysis Result

"lemma": [
"actor", "actor", "actress", "adequacy", "adequate", "adequately", "adhere", "adhere", "adherence", "adherent", "adherent", "adhere", "adhere", "adhesion", "adhesion", "adhesion", "adhesive", "adhesive", "adhesive", "adjoin",
"adjoin", "adjoin", "adjoin", "adjourn", "adjourn", "adjourn", "adjourn", "adjust", "adjustable", "adjust", "adjust", "adjust", "adjustment", …. "adjust",
],
"version": "7.0.7", "author": "twinword inc.", "email": "help@twinword.com", "result_code": "200", "result_msg": "Success"
}

10. UDPipe is a trainable pipeline for tokenization, tagging, lemmatization and dependency parsing of CoNLL-U files.

Analysis Result

Id	Form	Lemma	UPosTag	XPosTag	Feats	Head	DepRel	Deps	Misc
1	actor	actor	NOUN	NN	Number=Sing	2	compound	_	TokenRange=0:5
2	actors	actor	NOUN	NNS	Number=Plur	0	root	_	TokenRange=6:12
3	actress	actress	NOUN	NN	Number=Sing	2	compound	_	TokenRange=13:20
4	adequacy	adequacy	NOUN	NN	Number=Sing	2	appos	_	TokenRange=21:29
5	adequate	adequate	ADJ	JJ	Degree=Pos	2	amod	_	TokenRange=30:38
6	adequately	adequately	ADV	RB	_	7	advmod	_	TokenRange=39:49
7	adhere	adhere	VERB	VB	VerbForm=Inf	4	parataxis	_	TokenRange=50:56
8	adhered	adhere	VERB	VBN	Tense=Past|VerbForm=Part	9	amod	_	TokenRange=57:64

3.6 Conclusion and Summary

This chapter discussed the detailed study and implementation of existing stemming and lemmatization algorithms and forms the backbone of the research work conducted. Here the generation of output of popular affix-removal stemmers and lemmatizers are compared and are summarized for their correctness. The time wise performance of all affix-removal stemmers has also been checked.

The errors and limitations in the output led to realizing the Research Gap which was that a better and more efficient lemmatizer model was required to be developed. This also inspired the idea behind designing the Research Objectives for the proposed research work.

The next two chapters discuss the proposed models–LemmaChase and LemmaQuest, which were aimed to overcome all the observed limitations of this chapter. The proposed models are the combination of stemming and lemmatization techniques. The comparative model of affix-removal stemmers and study of statistical stemming techniques has been published as a paper titled 'Analyzing the Stemming Paradigm' in the journal Springer[10] .

[9] https://www.twinword.com/api/lemmatizer.php

[10] https://link.springer.com/chapter/10.1007/978-3-319-63645-0_37