TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

PolyWordNet: A Word Sense Disambiguation Specific WordNet of Polysemy Words

Udaya Raj Dhungana

(070/PHCT/402)

A DISSERTATION SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER ENINEERING IN THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

August, 2021

Dedicated to

My mother *Khem Kumari Dhungana*

My late father *Ram Prasad Dhungana*

Copyright©

Declaration of Authorship

Dissertation entitled “**PolyWordNet: A Word Sense Disambiguation Specific WordNet of Polysemy Words**" which is being submitted to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering (IoE), Tribhuvan University, Nepal for the award of the degree of **Doctor of Philosophy in Computer Engineering**, is a research work carried out by me under the supervision of **Prof. Dr. Subarna Shakya**, Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering (IoE), Tribhuvan University between May 2013 to August 2021. I declare that this is my work and has not been previously submitted by me at any university for any academic award.

Udaya Raj Dhungana

Signed:

Date: August 9, 2021

Recommendation

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a dissertation entitled “**PolyWordNet: A Word Sense Disambiguation Specific WordNet of Polysemy Words**", submitted by **Mr. Udaya Raj Dhungana** in partial fulfillment of the requirement for the award of the degree of **Doctor of Philosophy in Computer Engineering.**

Supervisor: Prof. Dr. Subarna Shakya
Professor, Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering, Tribhuvan University

Tribhuvan University

INSTITUTE OF ENGINEERING

The dissertation “**PolyWordNet: A Word Sense Disambiguation Specific WordNet of Polysemy Words**" submitted by **Mr. Udaya Raj Dhungana** for partial fulfillment of the requirement for the degree of **Doctor of Philosophy in Computer Engineering** has been accepted by the IOE Research Committee (IERC) upon the recommendation of the supervisor and the Departmental Research Committee (DRC) with the approval by the following examiners:

External Examiner:

Prof. Dr. Vivek Kumar Singh
Head and Professor,
Department of Computer Science,
Banaras Hindu University, Varanasi, India

Internal Examiners:

Prof. Dr. Manish Pokharel
Dean, SoE, Kathmandu University, Nepal

Dr. Bal Krishna Bal
Associate Professor
DoCSE, SoE, Kathmandu University, Nepal

Institute of Engineering
Office of the Dean

S.Joshi

Prof. Dr. Shashidhar R Joshi
Committee Chairperson,
Dean,
Institute of Engineering
August 9, 2021

Tribhuvan University

INSTITUTE OF ENGINEERING

The undersigned certify that they have evaluated the dissertation entitled "**PolyWordNet: A Word Sense Disambiguation Specific WordNet of Polysemy Words**" submitted by **Mr. Udaya Raj Dhungana** and have external oral presentation for the partial fulfillment of the requirement for the degree of **Doctor of Philosophy in Computer Engineering** and recommended to the IOE for acceptance of this dissertation.

Department of Computer Science
Banaras Hindu University
Varanasi-221005

Prof. Dr. Vivek Kumar Singh
Head and Professor, Department of Computer Science,
Banaras Hindu University, Varanasi-221005, India [External Examiner]

Prof. Dr. Manish Pokharel
Dean, SoE, Kathmandu University, Dhulikhel, Nepal [Internal Examiner]

Dr. Bal Krishna Bal
Associate Professor
DoCSE, SoE, Kathmandu University, Dhulikhel, Nepal [Internal Examiner]

Departmental Acceptance

The dissertation entitled “**PolyWordNet: A Word Sense Disambiguation Specific WordNet of Polysemy Words**" submitted by **Mr. Udaya Raj Dhungana** in partial fulfillment of the requirement for the award of the degree of **Doctor of Philosophy in Computer Engineering** has been accepted as a bonafide record of work carried out by him in the department.

Institute of Engineering, Pulchowk Campus
Department of Electronics and Computer Engineering

Prof. Dr. Ram Krishna Maharjan
DRC Chairman and Head of Department
Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering, Tribhuvan University,
Nepal

Date: August 9, 2021

This page is intentionally left blank.

Abstract

This dissertation presents a new lexical resource which is named as 'PolyWordNet'. The PolyWordNet mimics the way how the senses of polysemy words and their corresponding related words are organized in a human mind. A related word of a sense of a polysemy word in a given context is a word that can disambiguate the meaning of the sense of the polysemy word in that context. The rationale behind the organization of words in PolyWordNet is that any simple sentence, which contains a polysemy word, also contains at least a related word (s). A sense of a polysemy word and its related word(s) in a sentence, therefore, have a strong semantic relation which can be used to disambiguate the sense of the polysemy word. Utilizing this semantic relation, PolyWordNet organizes the senses of polysemy words based on their corresponding related words. The organization of words in PolyWordNet is completely different as compared to the existing other popular lexical resources such as dictionary and WordNet. The words in a dictionary are organized based on the alphabetical order. Therefore, the words that spell alike come together but the words with similar meaning get scattered in the dictionary. In WordNet, the words with similar meaning are placed together based on the synonymy set. The polysemy words are the big problems in Natural Language Processing tasks since they create the ambiguity. No lexical databases deals with the organization of words based on these polysemy words. Therefore, the PolyWordNet is developed. The words in PolyWordNet are organized in such a way that the senses of polysemy words and their corresponding related words come together and form clusters. The results obtained from 63 runs of experiments performed on 3,541 words and tested by 4,105 different test sentences show the word organization of PolyWordNet is better for word sense disambiguation. These results also indicate that the word organization of PolyWordNet is acceptable and valid with reference to the popular lexical database- WordNet.

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor **Prof. Dr. Subarna Shakya** for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D. study.

I am indebted to and most sincere thanks go to **Prof. Dr. Shashidhar Ram Joshi**, Dean, Institute of Engineering, Tribhuvan University for his kind supports and encouragements throughout this research work and for helping to develop mathematical model in this research. I am also thankful to **Dr. Surendra Shrestha**, Reader, Department of Electronics and Computer Engineering at Institute of Engineering, Pulchowk Campus for his motivational support and motivation in this research.

I am equally grateful to **Prof. Dr. Ram Krishna Maharjan**, DRC Chairman and Head of Department, Department of Electronics and Computer Engineering at Institute of Engineering, Pulchowk Campus. I am also indebted to and sincere thanks go to **Dr. Bal Krishna Bal**, Associate Professor, Kathmandu University for his guidelines, heartwarming encouragements and support during this research.

Tons of most sincere thanks go to **Prof. Dr. Bettina Harriehausen-Mühlbauer**, Professor, Department of Computer Science, Darmstadt University of Applied Sciences, Darmstadt, Germany for her continuous support during my research at Darmstadt University of Applied Sciences, Darmstadt, Germany. Her guidance and suggestions helped me to shape the research towards the goal. My sincere thanks goes to **Prof Dr. Christoph Wentzel,** Department of Computer Science, Darmstadt University of Applied Sciences , Darmstadt, Germany for providing the better environment to research at Department of Computer Science and for his guidance and encouragement towards this research during my stay at Darmstadt University of Applied Sciences, Darmstadt, Germany from September, 2014 to June, 2015.

I gratefully acknowledge the funding received to study PhD from **University Grant Commission (UGC)**, Nepal under **Young fellowship** in 2070/03/23 (**Ref no: 1266 2070/071**). I am also very much thankful to **IDEAS Project** (**Erasmus Mundus Action 2 Scholarship**) for providing the great opportunity to study and research at Darmstadt University of Applied Sciences, Darmstadt, Germany. I also want to extend my sincere thanks to **Ms. Lucia Koch**, Head of International Office, Darmstadt University of Applied Sciences, Darmstadt, Germany for providing the fund for my paper presentation at Riga, Latvia.

I wish to extend my thanks to my brother, **Bharat Sharma**, for helping me through the difficult times, and for his kind support during my Ph.D. study. I also wish to extend my many thanks to **Mr. Manish Shrestha**, Department of Electronics and Computer Engineering, Pulchowk Campus for his kind support and help during my Ph.D. study. I would like to express my most sincere thanks to my friend **Shreejana KC** for her encouragement and support to shape this research. Similarly, I feel happy to express my heartily thanks to **Aastha Neupane** for her help and support during this research work. I also extend many thanks to my loving sister **Asmita Dhungana** for her encouragement and supports to complete this research. I am also thankful to all respondents for their time and valuable information while conducting the data generation for this research.

Last but not least, I wish to extend my heartily thanks to my parents **late Ram Prasad Dhungana** and **Khem Kumari Dhungana** for their several years of love and care they had given to me and sincerely dedicate this research work to them. My loving wife, **Indira Adhikari**, receives tons of thanks for her eternal affections, and heartwarming encouragements. My loving daughters and son, **Amisha, Aayusha** and **Abhi Udaya**, have always been inspirations to me, and their cheerful faces always inspired me to move ahead even in difficult situations.

Udaya Raj Dhungana

070/PHCT/402

Contents

COPYRIGHT III

DECLARATION OF AUTHORSHIP IV

RECOMMENDATION V

DEPARTMENTAL ACCEPTANCE VIII

ABSTRACT X

ACKNOWLEDGEMENT XI

LIST OF FIGURES XIX

LIST OF TABLES XXII

LIST OF ABBREVIATIONS XXIII

CHAPTER 1. INTRODUCTION **.. 2**

1.1 Natural Languages and Ambiguity 2

1.2 Lexical Ambiguity and Word Sense Disambiguation 4

1.3 WordNet: A Lexical Database 7

1.4 Word Sense Disambiguation and WordNet 7

1.5 Rationale: Purpose, Problem and Solution 8

1.5.1 Purpose of Research 8

1.5.2 Problem Statement 9

1.5.3 Solution Approach 10

1.6 Aim of Research 12

1.7 Research Objectives 12

1.8 Research Design 12

1.9 Deliverables of Research 14

1.10 Significance of Research 14

1.11 Contributions of Research 15

1.12 Scope and Limitations 16

1.12.1 Scope of Research 16

1.12.2 Limitations of Research 16

1.13 Terminologies and Definitions 16

1.14 Dissertation Structure 17

CHAPTER 2. LITERATURE REVIEW **.. 19**

2.1 A Brief History of WSD 19

2.2 WSD Approaches 21

2.2.1 Corpus Based Approaches 21

2.2.2 Knowledge Based Approaches 23

2.3 English WordNet 23

2.3.1 Motivation to WordNet 23

2.3.2 WordNet Structure 24

2.3.3 Lexical Matrix 25

2.3.4 Semantic Relations 25

2.3.5 Parts of Speech 26

2.4 Other WordNets 28

2.4.1 GermaNet: A German WordNet 28

2.4.2 EuroWordNet 29

2.4.3 Japanese WordNet 31

2.4.4 Chinese WordNet 31

2.4.5 Hindi WordNet 32

2.4.6 IndoWordNet 34

2.4.7 Nepali WordNet 35

2.5 WordNets and Their Common Features 36

2.6 Use of WordNet in Word Sense Disambiguation 37

2.7 Related Task 38

CHAPTER 3. PROBLEM STATEMENT AND SOLUTION APPROACH **40**

3.1 Knowledg-Based WSD Methods 40

3.1.1 Contextual Overlap Count Methods 40

3.1.2 Methods Based on Similarity Measures 49

3.1.3 Selectional Preferences Methods 50

3.1.4 Heuristic Based Methods 51

3.2 Algorithms Using WordNet Relations 51

3.2.1 Adapted Lesk Algorithm 52

3.2.2 WSD Using WordNet Relations 54

3.2.3 Unsupervised WSD Using WordNet Relatives 54

3.2.4 WSD Method Based on Specification Marks 55

3.2.5 WSD for Nepali Language 57

3.2.6 Word Sense Disambiguation in Queries 59

3.2.7 Word Sense Disambiguation in Hindi Language 59

3.2.8 Nepali Word Sense Disambiguation 60

3.3 Points Noted on Algorithms Using WordNet Relations 61

3.4 Illustration of Use of Hypernym Relation 62

3.4.1 The Hyponymy/Hypernymy Relation 62

3.4.2 WSD Approaches and Use of Hypernyms 64

3.4.3 Analysis of Use of Hypernyms 68

3.5 Problem Statement 73

3.5.1 Insufficient Information in WordNet for Disambiguation 73

3.5.2 Noise Information and Wrong Disambiguation 73

3.5.3 Disambiguation depends on the gloss's words 73

3.5.4 WordNet lacks dealing relations with polysemy words 74

3.5.5 Research Question 74

3.6 Solution Approach 75

3.6.1 Human Mind and Word Sense Disambiguation 75

3.6.2 Polysemy Words and Related Words 76

3.6.3 Generation of Related Words 78

3.6.4 Organization of words- *PolyWordNet* 81

3.6.5 Disambiguation Process: A New WSD Algorithm 83

3.6.6 Mathematical Model- A New WSD method using PolyWordNet 86

3.7 PolyWordNet, Dictionary and WordNet 88

CHAPTER 4. RESEARCH METHODOLOGY **.. 90**

4.1 Research Design: Experimental 90

4.1.1 Experimental Designs 90

4.1.2 PolyWordNet Statistics 98

4.1.3 System Development 99

4.1.4 Data Collection 101

4.1.5 Results Analysis and Evaluation Tool 102

4.2 Participants of the Research 103

4.2.1 Role of researcher 103
4.2.2 Respondents 104
4.3 Validation of PolyWordNet, Data and Test Data 104
4.3.1 Validation of PolyWordNet 104
4.3.2 Validation of Data that is used to build PolyWordNet and Test Data 105
4.3.3 Reference Accuracy Range 105
4.3.4 Validation Method 108

CHAPTER 5. RESULT, ANALYSIS AND COMPARISON **110**
5.1 Experiment Series 110
5.2 Series A Experiments 111
5.2.1 Results of Experiment 1- *Exp 1 Run 1* and *Exp 1 Run 2* 112
5.2.2 Results of Experiment 2- *Exp 2 Run 1* and *Exp 2 Run 2* 112
5.2.3 Results of Experiment 3- *Exp 3 Run 1* and *Exp 3 Run 2* 113
5.2.4 Results of Experiment 4- *Exp 4* 113
5.3 Series B to Series E Experiments 113
5.4 Series F Experiments 120
5.5 Result, Analysis and Comparison 126
5.5.1 Result Analysis of Series A Experiments 126
5.5.2 Result Analysis of Series B-E Experiments 127
5.5.3 Result Analysis of Series F Experiments 129
5.5.4 Findings from Result Analysis 131
5.5.5 Handled and Unhandled Cases of WSD using PolyWordNet 133
5.6 Result of Experiments and Validation 134
5.6.1 Validation of PolyWordNet 134
5.6.2 Validation of Data and Test Sentences 135

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS **136**

6.1 Conclusion 136

6.2 Recommendation 137

REFERENCES **139**

APPENDICES **148**

Appendix 1: Polysemy Words in PolyWordNet 148

Appendix 2: Related Words in PolyWordNet 152

Appendix 3: Test Data (Test Sentences) 153

Appendix 4: List of Publications 159

List of Figures

Figure 1. 1: Senses of noun “Pen” in English WordNet 2.1 4

Figure 1. 2: The noun “Pen” used in two different contexts 5

Figure 1. 3: The five senses of noun “Party” 6

Figure 1. 4: The five different contexts where the noun “Party” is used 6

Figure 2. 1: Lexical Matrix illustrating synonym words (F1, F2) and polysemy word (F2) 25

Figure 2. 2: Ontology of *school* synset 33

Figure 2. 3: Relations in Hindi WordNet 33

Figure 2. 4: Statistics of Hindi WordNet (10 Feb 2016) 34

Figure 3. 1: Lesk Algorithm 41

Figure 3. 2: Meanings of words "Pine" and "Cone" 41

Figure 3. 3: Overlaps of words between each meaning of "pine" and "cone". 42

Figure 3. 4: Simulated annealing algorithm 44

Figure 3. 5: Simplified Lesk Algorithm 45

Figure 3. 6: Pseudo-code for simplified Lesk algorithm 45

Figure 3. 7: Example of simplified Lesk algorithm 46

Figure 3. 8: Multi-token overlap counting method of adapted Lesk algorithm 48

Figure 3. 9: Corpus-based Lesk algorithm 48

Figure 3. 10: Algorithm for Selectional Preferences 50

Figure 3. 11: Zipfian distribution of word meanings in SemCor 51

Figure 3. 12: The global matching scheme of adapted Lesk 52

Figure 3. 13: Algorithm proposed by Fragos et al. (2003) 53

Figure 3. 14: Unsupervised WSD using WordNet relatives proposed by Seo et al. (2004)..........55

Figure 3. 15: WSD method based on specification marks proposed by Montoyo et al. (2005)....56

Figure 3. 16: Hypernym Heuristic defined by Montoyo et al. (2005)..56

Figure 3. 17: Gloss Hypernym Heuristic defined by Montoyo et al. (2005)................................57

Figure 3. 18: Nepali WSD method proposed by Roy et al. (2014) 58

Figure 3. 19: Word Sense Disambiguation in Queries proposed by Liu et al. (2004) 58

Figure 3. 20: Hindi Word Sense Disambiguation proposed by Sinha et al. (2003) 59

Figure 3. 21: Nepali word sense disambiguation proposed by Dhungana and Shakya (2014) 60

Figure 3. 22: Inclusion of hypernym increases overlaps of context with correct sense 63

Figure 3. 23: Glosses of five senses of word "Pen" as a noun 64

Figure 3. 24: Sense bags formed from sense's gloss of *Pen* 65

Figure 3. 25: No of overlaps between the context bag and each sense bag of *Pen* 65

Figure 3. 26: Contents of first three sense bags *pen*...66

Figure 3. 27: Overlaps of words when the hypernyms are included in the sense bags 67

Figure 3. 28: Context bag when the hypernyms of "poem" are included.....................................67

Figure 3. 29: Number of overlaps between context bag and first three sense bags of *pen* 68

Figure 3. 30: (a) Common and distinct words in the first three senses of *Pen* and (b) number of overlaps of two contexts with the first three senses of *Pen*...70

Figure 3. 31: The first three senses of noun *Pen* in WordNet 2.1 71

Figure 3. 32: Research question 74

Figure 3. 33: Related words of first three senses of noun "pen" in WordNet 2.1 77

Figure 3. 34: Algorithm for generating related words...80

Figure 3. 35: Organization of words in PolyWordNet 82

Figure 3. 36: A new WSD algorithm using PolyWordNet..84

Figure 4. 1: Setting of Experiment 1 (a) Exp 1 Run 1 and (b) Exp 1 Run 2 94

Figure 4. 2: Setting of Experiment 2 (a) Exp 2 Run 1 and (b) Exp 2 Run 2 95

Figure 4. 3: Setting of Experiment 3 (a) Exp 3 Run 1 and (b) Exp 3 Run 2 96

Figure 4. 4: Setting of Experiment 4 (Exp 4) 97

Figure 4. 5: Database structure of PolyWordNet 100

Figure 5. 1: *Recall*, *Precision* and *Coverage* of Series B to E Experiments 115

Figure 5. 2: Average *Recall*, *Precision* and *Coverage* of Series B to E Experiments.................117

Figure 5. 3: Series-wise comparison of Series B to E Experiments..118

Figure 5. 4: *Recall, Precision* and *Coverage* comparison of Series B to E Experiments............119

Figure 5. 5: *Recall, Precision* and *Coverage* obtained in Series F Experiments.........................122

Figure 5. 6: Average *Recall, Precision* and *Coverage* of Series F Experiments.........................123

Figure 5. 7: *Recall, Precision* and *Coverage* obtained in Series F Experiments.........................124

Figure 5. 8: *Recall, Precision* and *Coverage* obtained in Series F Experiments.........................125

List of Tables

Table 2. 1: Number of words, synsets and senses 27

Table 2. 2: Polysemy information 27

Table 2. 3: Relationships in GermaNet 28

Table 2. 4: WordNets in European Languages and Responsible Institutes..................................29

Table 2. 5: WordNet1.5 Relations 30

Table 2. 6: The Equivalence Relations in EuroWordNet 30

Table 2. 7: WordNets in IndoWordNet 34

Table 3. 1: Relations utilized by adapted Lesk...47

Table 3. 2 : Comparison among Dictionary, WordNet and PolyWordNet...................................88

Table 5. 1: Series A Experiments and Word Statistics in PolyWordNet 111

Table 5. 2: Results obtained in Series A Experiments 111

Table 5. 3: Number of test sentences that are correctly disambiguated in Series B to Series E Experiments 114

Table 5. 4: *Recall, Precision* and *Coverage* of Series B to E Experiments................................114

Table 5. 5: Final Word Statistics in PolyWordNet and Sample WordNet 120

Table 5. 6: Recall, Precision and Coverage observed in Series F experiments.........................121

Table 5.7: Number of sentences which are correctly disambiguated in one run but incorrectly disambiguate in other run in Series A Experiments 126

List of Abbreviations

ALPAC	Automatic Language Processing Advisory Committee
BOW	Sinica Bilingual Ontological Wordnet
CFILT	Centre for Indian Language Technology
CIIL	Central Institute of Indian Languages
CWN	Taiwan University WordNet
DARPA	Defense Advanced Research Projects Agency
ECTED	English-Chinese Translation Equivalents Database
EWN	EuroWordNet
HWN	Hindi WordNet
IA	Intersection Approach/Algorithm
IE	Information Extraction
ILI	Inter-Lingual-Index
IR	Information Retrieval
ISM	Initial Specification Mark
MDA	Minimum Distance Approach/Algorithm
MRD	Machine Readable Dictionary
MT	Machine Translation
NLP	Natural Language Processing
NWN	Nepali WordNet
OALD	Oxford Advanced Learner’s Dictionary of Current English
OL	Ontology Learning
PWN	Princeton WordNet
SA	Semantic Annotation
SEW	Southeast University WordNet
SM	Semantic Mapping
SR	Speech Recognition
SUMO	Suggested Upper Merged Ontology

TD	Test Data
TS	Test Sentence
WCA	Words Co-occurrence Approach/Algorithm
WSD	Word Sense Disambiguation

This page is intentionally left blank.

Chapter 1

Introduction

1.1 NATURAL LANGUAGES AND AMBIGUITY

Natural languages are human languages such as Nepali, English or German and is a means of communicating. The natural language uses sounds or conventional symbols to exchange the information. People of different group, region, country or continent use the different languages. The report of Ethnologue in [1] shows 7,106 different languages are used and spoken all over the World. The report published by BBC in [2] shows that the 90% of these languages are used by less than one lakh people. In addition, more than a million people use and speak 150 to 200 languages.

One common property to these languages is that they contain ambiguities while expressing the information. An ambiguity generally occurs when a single word or a single sentence gives more than one meaning. Ambiguity can be viewed from two perspectives. The ambiguity can be viewed from human perspective. In this perspective, a single sentence actually can have multiple meaning even for the human. It can also be viewed from computer/machine perspective. In this perspective, the meaning is not ambiguous for the human but ambiguous for the computers/machines. The first type of ambiguity cannot be solved since they are themselves ambiguous even for the human. Therefore, sentences with such ambiguity must be avoided to use. The second type of ambiguity is not ambiguous to human since human are enough intelligent to understand the context of the sentence to understand its correct meaning. However, it is a big problem for computers to understand the meaning of the word in a context

[3]. There are generally four types of ambiguity[1]. These include lexical, syntactic, semantic and anaphoric ambiguity.

The lexical ambiguity occurs when a same word has more than one meaning in different contexts. For instance, "She goes to bank". The word "bank", here, may have two meanings. It is ambiguous to human as well since the context is not clear. The "bank" may be a financial institution or it may be a bank of a river or lake. Such words are called homograph. Homograph words spells same, but they have different meanings. In speech recognition, ambiguity arises if two differently spelled words have same pronunciation. Such words are called homophone.

A homophone is a word that has the same pronunciation but is spelled differently and has a different meaning or different pronunciations. For example, the word pairs to/two, there/their and pray/prey have the same pronunciation but different spellings and meanings. If a sentence can have multiple parse tree, it causes to arise syntactic ambiguity. For example, in the context "Ram ate a salad with potato from Pokhara for lunch.", the phrase "with potato" can attach to "salad" or "ate" and the phrase "from Pokhara" can attach to "potato" or "salad". If a sentence gives multiple meanings, semantic ambiguity arises. For example, for the sentence "Ram and Sabita are married.", it may mean Ram and Sabita are married separately or it may mean Ram and Sabita married each other.

If a phrase or word that refers to something previously stated and there is more than one possibility to refer, anaphoric ambiguity occurs. For example, in the context "Ram told Hari that he need to buy a car.", "he" may refer to both "Ram" and "Hari". Thus the sentence may be understood either as "Ram told Hari that Ram need to buy a car." or "Ram told Hari that Hari need to buy a car.".

In these days, the natural language is not only used by the human but also being used by machines. Machines are being adapted to understand the natural languages so that people can communicate with machine and/or can use the machine to automate various natural language

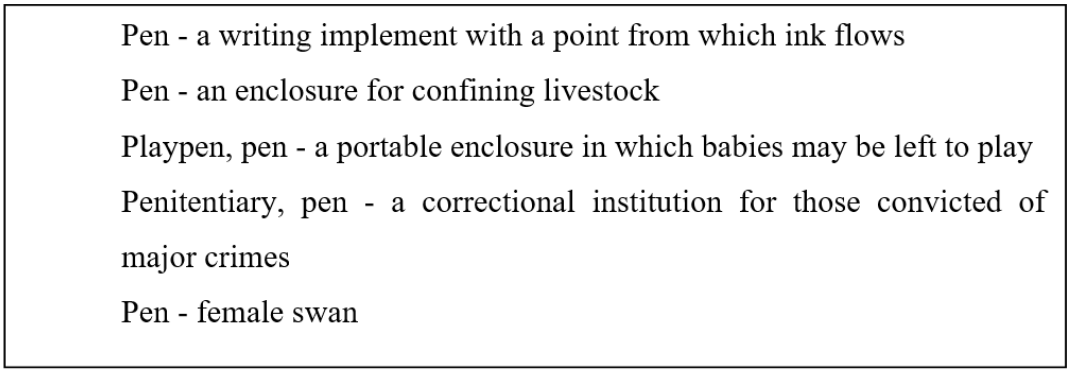
[1] http://cs.nyu.edu/faculty/davise/ai/ambiguity.html

processing tasks. The natural languages must be processed in different level to automate the systems that need to understand natural languages.

The natural language processing may include the intra-language processing like Summarization of text information, Information Extraction (IE), Information Retrieval (IR), Speech Recognition or inter-language processing like Machine Translation (MT). In all cases, ambiguity is a big problem in every natural language processing task when done by machines. It doesn't create a problem for human to get the meaning of a word. However, it is a big problem for the machines to understand the meaning of an ambiguous word in a context since the machines are not intelligent like human to understand the natural language. Many approaches have been used to make machines/computers understand the meaning of multi sense words in a context.

1.2 LEXICAL AMBIGUITY AND WORD SENSE DISAMBIGUATION

This research limits its scope and focuses only in word sense disambiguation process for lexical ambiguity in the written texts. This ambiguity occurs due to homograph word which is known as a ***polysemy word***.



Pen - a writing implement with a point from which ink flows

Pen - an enclosure for confining livestock

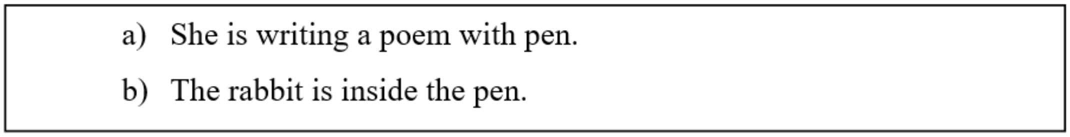
Playpen, pen - a portable enclosure in which babies may be left to play

Penitentiary, pen - a correctional institution for those convicted of major crimes

Pen - female swan

Figure 1. 1: Senses of noun "Pen" in English WordNet 2.1

Every natural language has polysemy words. They have more than one sense when used in different context. These polysemy words create big problems in Natural Language Processing (NLP). Human can easily analyze the context and understand the meaning of such polysemy word. But the machines cannot do so. It is referred to as word sense ambiguity in computation. For instance, suppose an English word “Pen”. In Princeton WordNet 2.1 [4], the noun “Pen” has five different senses as shown in Figure 1.1. This word has five different meanings for different contexts. For instance, suppose two different contexts as shown in Figure 1.2. In the first context, the meaning of the word “Pen” is *“a writing implement with a point from which ink flows”* while the meaning of “Pen” in second context is *“an enclosure for confining livestock”*. Human can easily get the meaning of the “Pen” by reading and analyzing these two sentences in which the word is used. This creates no problem for human at all to understand the meaning in different contexts. Unfortunately, machine does not have any idea to get the meaning of the word in different contexts. This situation leads to a problem called “ambiguity in word sense” while processing natural language by machines. Such type of ambiguity which occurs on finding the correct meaning of a word having multiple senses is called lexical semantic ambiguity.



a) She is writing a poem with pen.

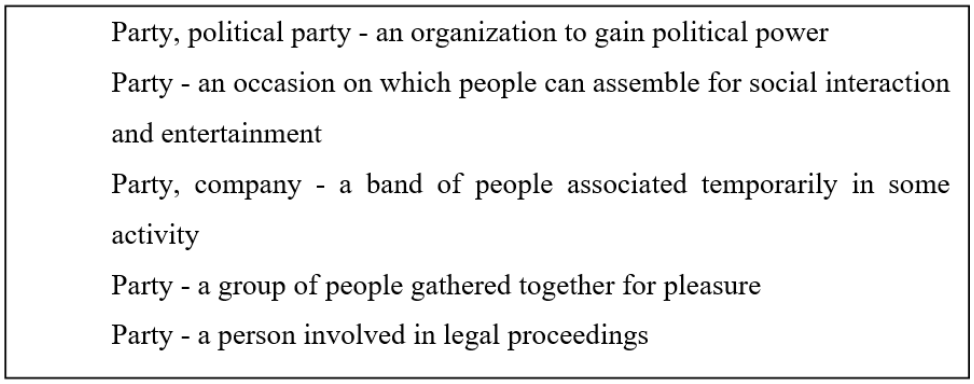
b) The rabbit is inside the pen.

Figure 1. 2: The noun “Pen” used in two different contexts

Like a human, to be able to find the correct meaning of a polysemy word, machines should first read and analyze the context. The process followed by machines to find the accurate meaning of the polysemy word is known as Word Sense Disambiguation (WSD).

Similarly in English WordNet 2.1, as a noun another word “Party” has 5 different senses: - *“a political party”, “an occasion on which people can assemble for social interaction and entertainment”, “a company”, “a group of people gathered together for pleasure” and “a person involved in legal proceedings”*. These senses of the word “Party” are shown in Figure 1.3.

Suppose different contexts where the noun “Party” has been used as shown in Figure 1.4. Although there is no ambiguity for finding the sense of word for a human, it is not hard to find the meaning of the word “Party” in the different contexts. The meaning of “Party” in sentence number 1 is “*political party*”, in sentence number 2 is “*an occasion on which people can assemble for social interaction and entertainment*”, in sentence number 3 is “*band of people associated temporarily in some activity*”, in sentence number 4 is “*a group of people gathered together for pleasure*” and in sentence number 5 is “*a person involved in legal proceedings*”.



Party, political party - an organization to gain political power

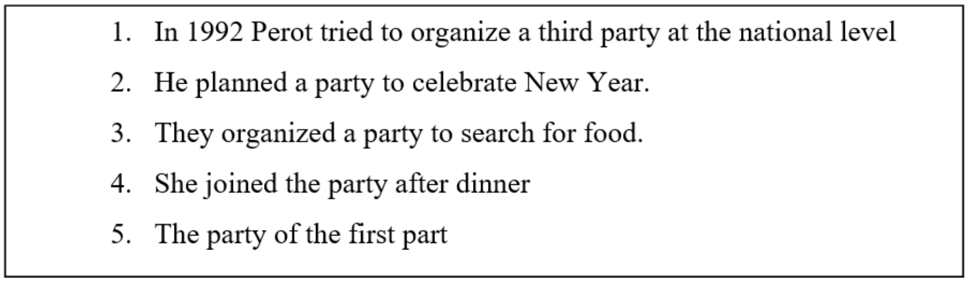
Party - an occasion on which people can assemble for social interaction and entertainment

Party, company - a band of people associated temporarily in some activity

Party - a group of people gathered together for pleasure

Party - a person involved in legal proceedings

Figure 1. 3: The five senses of noun “Party”



1. In 1992 Perot tried to organize a third party at the national level
2. He planned a party to celebrate New Year.
3. They organized a party to search for food.
4. She joined the party after dinner
5. The party of the first part

Figure 1. 4: The five different contexts where the noun “Party” is used

For machine, such situation creates ambiguities on identifying the sense of such words which give multiple meanings in different context. For language translation or information retrieving tasks, it is highly desired that the system must be provided with a module which could computationally disambiguate the meaning of polysemy words with higher accuracy. It is one of the very complex problems in NLP and is considered as AI-complete problem.

1.3 WORDNET: A LEXICAL DATABASE

There are many lexical resources such as dictionaries, thesauri, ontology, collocation etc. The information from these lexical resources are used by the knowledge-based WSD approaches to disambiguate sense of polysemy word. In early 1980s, these resources were found to be contained less information about a word. The information from these resources were noticed not to be sufficient for word sense disambiguation [5]. This situation leads to a need of a new resource that contains more information about a word. In early 1990s, a new resource-WordNet was developed at Princeton University. The WordNets organized the words based on synonym sets. Each word is connected with various semantic relations such as hypernym, hyponym, and so on [6] [7] [8]. This solved the problem of lack of sufficient information. WordNet is a lexical resource for English language [9]. After the development of WordNet, many Word Sense Disambiguation (WSD) methods used for word sense disambiguation and it became popular in NLP tasks. Following the principal idea of Princeton WordNet, WordNets for many other languages like French, German, Spanish, Chinese and Hindi etc. were built. These WordNets are now used as one important resource for word sense disambiguation.

1.4 WORD SENSE DISAMBIGUATION AND WORDNET

There are a large number of approaches for word sense disambiguation proposed till now [10]. These WSD approaches fall mainly into two groups - knowledge-based approaches and corpus-based approaches. This research work limits its scope and only focuses on the knowledge-based approaches.

Weaver in [11] had explained it is impossible to determine the sense of a single polysemy word if it is taken without any context. The WSD approach using dictionary was started when the Michel Lesk in 1986 applied the overlap count method in word sense definitions available in Oxford Advanced Learner's Dictionary (OALD) for word sense disambiguation [12]. However, dictionaries were found to have insufficient information. Such insufficient information in dictionary were unable to provide sufficient information to disambiguate the sense. It is because dictionaries only contain the short definitions for words. Due to the fact that dictionary did not sufficiently provide sufficient information for sense disambiguation, there found a need of some other lexicon resources which could sufficiently provide sufficient information for WSD. The development of WordNet was started in 1985 at Cognitive Science Lab of Princeton University. This project was handled under the direction of psychology professor George A. Miller. It was became available from the early 1990s. The WordNet provided more information about a word connecting with various relations such as hyponymy, hypernymy, holonymy etc. The WordNet solved the problem of less information for WSD approaches. Therefore, after its availability, WordNet is massively used as a resource in knowledge-based WSD approaches.

Pedersen et al (2002) used WordNet's information to adapt the Lesk algorithm for sense disambiguation [13]. The use of WordNet instead of dictionary increased Lesk algorithm's accuracy from 16% to 32%. Due to this impression, WordNets were built on other languages as well.

1.5 RATIONALE: PURPOSE, PROBLEM AND SOLUTION

The section explores the purpose of this research, the research problem and finally presents the solution approach for the stated problem.

1.5.1 Purpose of Research

The ambiguity in word sense arises due to the polysemy words. A context along with a polysemy word contain at least a word (called related word of the sense of the polysemy word) that clearly disambiguate the meaning of the polysemy word in the context. If the senses of polysemy word and their corresponding related words are connected, it will help to disambiguate the sense of the polysemy word more accurately, efficiently and easily. There are no lexical databases that deal with and organize the senses of polysemy words till now. This is the main motivation towards this research with a purpose to develop a new lexical database that organizes words based on senses of polysemy words. This new lexical database can be used to find the correct sense of polysemy word in the context.

1.5.2 Problem Statement

This research analyzed many other research works on WSD which uses WordNet. Some of these include the WSD methods by [13], [14], [15], [16], [17], [18], and [19]. From these works, it was noticed that the WordNet is an important resource which can be used for word sense disambiguation. However, it doesn’t exactly fit for knowledge-based, contextual overlap count approaches although it contains more information than a dictionary.

In knowledge-based approaches, the contextual overlap count WSD methods use the WordNet to utilize a large amount of information for sense disambiguation of polysemy word. However, these WSD methods can utilize only very few words from the WordNet for sense disambiguation. Therefore, it is waste of processing time and space required to store that huge amount of unused information during processing period for sense disambiguation. In addition, same words/information in WordNet are connected with many senses of the same polysemy word. This creates again another ambiguity while using the information from WordNet in knowledge-based, contextual overlap count WSD methods. From the work [20], it is noticed that the correctly disambiguated polysemy words start to be incorrectly disambiguated when information from deeper levels of the hypernyms are used for disambiguation. It is because probably from the second and/or third levels of Hypernyms of most of senses of polysemy words (for example "Pen") are found to be common. The same hypernyms are expressing the different senses of a polysemy word in WordNet. The information from such common hypernyms cannot be used to disambiguate the different senses of the same polysemy word.

The information taken from the WordNet including hypernyms increases the information. However, this increase in information does not relate the context with the correct sense of the polysemy word if the senses have the common hypernym hierarchies. Even though WordNet contains more information, only very few distinct information among the different senses of the polysemy word can be used for disambiguation purpose. Therefore, the information taken from WordNet is still insufficient to disambiguate the senses. The common hypernyms from WordNet induced a noise information during disambiguation process. This noise information leads to the incorrect disambiguation. The causes of these problems are explained in detail in Chapter 3.

There is, till now, no especial and dedicated lexical database that deals with and organizes the polysemy words which are the main cause of word sense ambiguity in every natural languages. Even WordNet does not have any relation that deals with polysemy words. Furthermore, the current organization of words in popular lexical resource WordNet is inadequate to address the problem of word sense disambiguation of polysemy words. This is because the current organization of words in WordNet causes the noise information for overlap count knowledge based WSD systems due to the common information for two or more than two senses of the same polysemy word. These are the main problems this research work intends to address.

1.5.3 Solution Approach

WordNet is most used and popular lexical resource in natural language processing. However, the current organization of words in WordNet is inadequate to address the problem of word sense disambiguation of polysemy words as discussed in 1.5.2 subsection. The main limitation of all existing lexical database is that they don't deal with polysemy words. That is they don't define any relation that connects the polysemy words and corresponding related words. Therefore, it is motivated to develop a new lexical resource/database which deals with and organizes the polysemy words based on the related words.

In dictionary, words are organized in alphabetical order. Due to this arrangement, the words which spell similar come together. However the words with similar meaning get scattered. This doesn't deal with polysemy words. The arrangement of words in WordNet is different. The words with similar meaning are put together based on the synonymy set. The WordNet organizes the words in different semantic relations like synonyms, hypernymy, hyponymy, meronymy etc. However, this also doesn't have any provision to deal with any relation for connecting polysemy words and corresponding related words. The motivation towards this research arises right from this point. In the field of natural language, many relations such synonyms, antonyms, hypernymy, hyponymy, meronymy etc. have been found and used since long. Till now, no single research work is thinking of there exists some strong relation between polysemy words and the related words in a context. If this relation is formulated properly in some lexical resource, then it can be used to find out the correct sense of the polysemy word.

The solution approach has utilized the fact that for a given context if it contains a polysemy word (say P), the context also contains at least another word (called related word, say R_w) which gives correct sense (say S_k) of P. The word R_w is a related word for the sense S_k of the polysemy word P. Since R_w helps to get the meaning S_k of polysemy word P, these R_w and S_k are semantically related. Therefore, R_w and S_k are interconnected/linked each other so that S_k can be reached/visited from R_w through a connection path and vice versa.

For every context or sentence that exist in any natural language, if it contains a polysemy word, it simultaneously contains a related word that determines the correct meaning of the polysemy word in that context/sentence. This relation between polysemy words and related words that come together in a context should be find out and are very significant for sense disambiguation. The collections of such relations between polysemy words and related words, when put together, forms the new lexical database called **PolyWordNet**.

To disambiguate a sense is simple using PolyWordNet. For a given context, polysemy word (P) and it's all senses (S_1, S_2 … S_k … S_n) are determined and stored in an array. Similarly, context words ($R_{w1,}$ R_{w2} … R_{wn}) from the given context are collected and stored in another array. For each related word R_{wk}, WSD method will explore all the possible paths originated from R_{wk} and try to find all possible connecting paths that lead to a sense of P.

Suppose while finding a connection path, a connecting path is found between R_{wk} and S_k, then S_k is the correct meaning/sense of P in the given context. Since the same context word is linked with only one sense of the same polysemy word in PolyWordNet, the context word does not lead to two or more senses of the polysemy word. In addition, since this research is focusing in simple sentences only the two context word doesn't lead to two or more senses of the same polysemy word.

1.6 AIM OF RESEARCH

The aim of this research is to develop a new lexical database that deals with polysemy words and organizes the senses of polysemy word based on their corresponding related words. This utilizes the fact that every context if it contains polysemy word, it also contains the related words which disambiguate the sense of that polysemy word in that context. The new developed lexical resources contains the relations between senses of polysemy word and their corresponding related words. These relations can be then used for sense disambiguation more accurately.

1.7 RESEARCH OBJECTIVES

This research strongly believes that there exist a novel relation between senses of polysemy word and their corresponding related words that come together in a context/sentence. This relation is a principal evidence which can be used for word sense disambiguation in a context.

The objectives of this research includes: - to study the structure of WordNets of various languages, how the words in the WordNets are organized, how the WordNets are being used in WSD and to find out the issues in WordNet that are causing to get low accuracy in WSD approaches. The objectives are listed as:

1. To study structures and word's organization of existing WordNets of various languages.
2. To investigate factors, issues and problems in WordNet that are causing WSD methods to obtain high accuracy.
3. To design and develop a new lexical resource/database that deals with the polysemy words and organizes the senses of polysemy words based on their related words.

1.8 RESEARCH DESIGN

Based on the objectives of this research, experimental research strategy is used in this research. Various data collection methods such as telephone interview, questionnaire and documents are adapted in this research to collect data set to test data. The participants of this research include the researcher and the respondents who are interviewed and asked to answer questionnaires to collect the data required for this research.

The researcher prepared the questionnaires, selected the respondents and distributed the questionnaires to the respondents. In this research, the data are collected in three stages. In first stage of data collection, an English lecturer is selected from the Central College of Pokhara University to find out 10 polysemy words. In second stage of data collection, 15 graduate students are randomly chosen. The 10 out of 15 were selected from Pulchowk Campus, IoE, Tribhuvan University, Nepal and remaining 5 are selected from Darmstadt University of Applied Sciences, Darmstadt, Germany. These respondents are contacted via telephone and email. The questionnaires are distributed and finally collected via email. In the third stage of data collection, 5 students studying at last year of Computer Engineering at Pokhara Engineering College, Pokhara University are selected. Each of these students collected 100 plus polysemy words and their related words. They also collected 2500 plus test sentences. After the respondents returned the questionnaires, researcher sent the answered questionnaires to English lecturer to check the information collected in questionnaires are correct semantically and grammatically. After this, researcher prepared the final information to build the lexical databases and test data.

Altogether six series of experiments are set up. In each series except in Series F, the number of data are increased to observe the effect on increasing the numbers of words in PolyWordNet and WordNet. Each of these six series of experiments has 7 different experimental settings. The first 6 experiments (named as Exp 1 Run 1, Exp 1 Run 2, Exp 2 Run 1, Exp 2 Run 2, Exp 3 Run 1 and Exp 3 Run 2) in each series use the WordNet and simplified Lesk. The seventh experiment (named as Exp 4) uses the PolyWordNet. The difference in the first 6 experiments in each series is the amount of information used for disambiguation. As we go from experiment 1 to 6, the amount of information is increased. The seventh experiment uses a simple WSD algorithm and PolyWordNet developed in this research. For series F experiments, 1200 test sentences are randomly collected from ***news*** category of popular Brown Corpus. The amount of data in PolyWordNet and sample WordNet throughout the Series F experiments are same (i.e. 3541 words) as in Series E experiments. The 7 different experiments are repeated 4 times and tested with different numbers of these test sentences. The experiments are tested by 100 sentences in first, then by 400 sentences in second repetition, 800 hundred sentences in third repetition and finally by 1200 sentences in fourth repetition. Thus, altogether 63 experiments are run and obtained results are compared and analyzed to check whether the organization of words in PolyWordNet is acceptable for sense disambiguation.

The results from experiments are evaluated using standard WSD evaluation metrics- ***recall, precision*** and ***coverage***. These are then presented using line diagrams to compare and analyze the patterns of results in order to draw conclusions.

1.9 DELIVERABLES OF RESEARCH

The first deliverable of this research is a new lexical resource called **PolyWordNet.** The first deliverable- the PolyWordNet organizes the senses of polysemy words based on their corresponding related words. In this word organization, the senses of a polysemy word and their related words come together and form clusters.

A simple **WSD algorithm** that uses the relations of words from PolyWordNet for word sense disambiguation is another deliverable of this research. This algorithm is not an optimized WSD algorithm for word sense disambiguation. However, it can be used to access the relations in PolyWordNet for word sense disambiguation. The third deliverable[2] of this research is **WSD Evaluation Exercise** which is available in Kaggle. This contains 4105 sentences with 320 occurrences of polysemy words.

1.10 SIGNIFICANCE OF RESEARCH

This research has explored the principal relationship among the senses of polysemy word and their corresponding related words that come together in context. Based on this relation, a new lexical resource that deals with and organizes the senses of polysemy words is built. Till now,

[2] https://www.kaggle.com/udayarajdhungana/test-data-for-word-sense-disambiguation

no lexical resources are dealing with the relationship among the senses of polysemy word and their corresponding related words. This relationship is a principal evidence for word sense disambiguation in a given context. In this sense, this research work has a potential significance since it has developed a new lexical database PolyWordNet that deals with the polysemy words.

The PolyWordNet can be used in every NLP tasks like Machine Translation, Information Retrieval, and Text Summarization etc. where there is a need of word sense disambiguation. In addition, PolyWordNet can be used as a dictionary to find the meaning of words and can be used to find the words which are related to a particular word.

1.11 CONTRIBUTIONS OF RESEARCH

There are no any lexical resource that deals with the organization of polysemy words. This research work has considered and explored the principal relationship between the polysemy words and related words. This research work has utilized this relation to build a new lexical resource that deals with and organizes polysemy words and related words. In addition, a new WSD algorithm is developed. This WSD algorithm uses the PolyWordNet for word sense disambiguation. These are significant contributions of this research in the field of NLP tasks. These contributions are listed below:

1. This research work has **explored the principal relationship** between polysemy words and related words.
2. A **new** lexical database **PolyWordNet** is developed. The PolyWordNet deals with and organizes the polysemy words which are the root cause of word ambiguity in any NLP tasks.
3. The **WSD algorithm** which uses PolyWordNet for sense disambiguation is a new WSD method.
4. **A WSD Evaluation Exercise** is prepared from the Test Sentences used in this research. It is available in Kaggle. This contains 4105 sentences with 320 occurrences of polysemy words.

1.12 SCOPE AND LIMITATIONS

Scope refers to boundaries of research that a researcher sets to focus on their research of interest. On the other hand, limitations are the possible weaknesses of research which are generally out of control.

1.12.1 Scope of Research

This research limits its scope to study the structure of WordNets in different languages and the contextual overlap-count knowledge-based WSD approaches. This research work studied the structure of the WordNets, how the information from these WordNets are being used for the word's sense disambiguation, investigated the factors/issues that are causing for low accuracy and finally developed a new lexical resource that organizes the polysemy words. This research work does not concern about any other WSD approaches such as supervised, unsupervised, semi-supervised, statistical WSD approaches or any other approaches.

1.12.2 Limitations of Research

The developed PolyWordNet contains only 3541 words. The experiments are tested in these data set that only contains 3541 words. The Test Data (Test Sentences) generated in this research are only 2905 different sentences. In addition to these test sentences, only 1200 sentences are taken from Brown corpus to test the experiments. Therefore, the test data that is used to evaluate the new lexical database PolyWordNet and WSD algorithm contains only 4,105 English sentences.

1.13 TERMINOLOGIES AND DEFINITIONS

In this section, some terms are defined and are used throughout this dissertation to present this research work.

Definition 1: ***Polysemy word*****: -** It is a word which gives more than one meaning when used in different contexts. For example, let us take two different sentences: 1) *He deposited money in bank* and 2) *He is walking on river bank*. The "bank" has two meanings in these two contexts.

It means a financial institution in sentence 1. It means a slopping land alongside water sources. Therefore, "bank" is a polysemy word.

Definition 2: ***Context***: - It is a phrase or a sentence or a paragraph which expresses a clear meaning to a polysemy word. For example, the first and second sentences in the definition 1 are the two context for "bank".

Definition 3: ***Target Word***: - It is a polysemy word whose meaning is to be disambiguated. For instant, "bank" is called a target word in the context "*He deposited money in bank.*"

Definition 4: ***Related Words***: - They are the words that come with polysemy word in a context/sentence. These words disambiguate the sense of the polysemy word in that context. For instant, in sentence "*a person is sitting on a pan and writing poem with a pen*", the words "***writing***" "***poem***" are the related words for the sense – "*a writing implement*" of the polysemy word "***pen***".

Definition 5: ***Context bag***: - It is a set of words taken from the given context. It may contain only the words from the given context or it may contains the words that are collected from the various relationships like glosses or hypernyms etc. in WordNet for every word in the context.

Definition 6: ***Sense bag***: - It is a set of words collected for every sense of a polysemy word. Like context bag, it may contain only the words from the sense definition or it may contains the words that are taken from various relationships like glosses or hypernyms etc. in WordNet.

1.14 DISSERTATION STRUCTURE

To describe the research work in detail, this dissertation has been organized into six chapters: Introduction, Literature Review, Statement of Research Problem and Solution Approach, Research Methodology, Experiment and Results and Conclusion.

The Chapter 1 is introductory. The chapter discusses on the natural languages and the types of ambiguities that found in Natural Languages. It then introduces the lexical ambiguity. It provides the information of the lexical database WordNet, its use in the WSD. The remaining sections provides the problems found in the use of WordNet for WSD approaches and the solution approach to the stated problem. **Chapter 2** explores the literature review in detail. It briefly discusses about the history of Word Sense Disambiguation Approaches and WordNet. It provides the detail information about the structure of the WordNet and then insights into the use of WordNet and hypernymy in WSD approaches. Finally, it discusses the limitations of using the hypernymy from the WordNet in WSD.

Chapter 3 analyses and defines the research problem in detail and presents a novel solution approach to the stated problem in detail. This chapter first presents the various WSD algorithms which are contextual overlap count knowledge-based methods, investigate the problems in those algorithms and finally present the problem and solution approach to the stated problem. **Chapter 4** discusses the research methodology that is followed to achieve the stated objectives. It also discusses on the methods for data collection. Validation methods that are used to validate the new lexical database, data set and test sentences are also discussed.

Chapter 5 is devoted to analyze and discuss the results from the various experiments. This also points out the findings noticed from the results of the experiments. Finally, the **chapter 6** concludes the findings based on the result obtained from experiments and also discusses about recommendations for future works.

Chapter 2

Literature Review

2.1 A BRIEF HISTORY OF WSD

Word Sense Disambiguation (WSD), is a very old problems in computational linguistics. It was originated from late 1940s along with the beginning of concept of Machine Translation (MT). Weaver had already introduced a view that provided with some context surrounding the word where it is used, has sufficient evidence to find the meaning of the polysemy word [21]. However, it is impossible to find the correct meaning of a word if it is examined one at a time in a book. In addition to this, Weaver also noticed that 1) context play vital role and 2) necessity of the statistical semantic studies as primary step. Following Weaver's principle, many WSD works has been done to find the meaning of polysemy words during 1950s. Much works are done for bilingual dictionary and for applying simple statistical models. It is claimed in [22] that the words on either sides of a polysemy word has the resolving power equivalent to whole context. In [23], in different domains, word's sense frequencies was calculated and then Bayes formula was applied to choose the most probable sense for a given context.

Difficulty for WSD and the argument of Bar-Hillel in [24] created a big threaten for WSD research during 1960s. When Bar-Hillel argued that there is no existing or imaginary program that will enable an electronic machine to determine the meaning of the word "pen" as an "enclosure" in a context like "*Little John was looking for his toy box. Finally he found it. The box was in the pen*". These causes the unfavorable ALPAC report [25] for WSD research and most of researches in MT in US are halted in 1966. Applying the statistical semantic proposed by Weaver [26], explained about the three words "in the pen" is the strong indicative of the sense "enclosure" since it is obvious that what may be inside a writing pen rather than ink.

During 1970s, WSD was included in Artificial Intelligence research [27]. Preference semantic approach was developed by Wilks in [28]. However, there was a lack of proper knowledge representation and it was mandatory for the AI WSD approaches. There was lack of such knowledge sources. Therefore, WSD systems were facing the problem of knowledge acquisition bottleneck since knowledge sources at this time were mostly hand-coded and hence there was lack of large machine-readable knowledge sources [29]. After the availability of large scale lexical resources like machine-readable knowledge sources and corpora in 1980s, the situation made a U-turn for WSD research.

During 1980s, dictionary-based WSD was started with Lesk's algorithm which used *Oxford Advanced Learner's Dictionary's word definitions* to determine the overlap in word definitions to disambiguate senses. Afterward, dictionary-based approaches such as [30] and [31] are tried. However, due to the fact that dictionary did not sufficiently provide complete information for a word sense, there found a need of some other lexicon resources which could sufficiently provide complete information for WSD purpose.

This desire in WSD was fulfilled in 1990s when three major developments were in existence in WSD environment. The first achievement was the availability of WordNet, next achievement was the statistical revolution in NLP field and the third one is development of SENSEVAL.

WordNet is developed in Princeton University. It is a lexical database and organizes the words based on synonyms. It hierarchically presents the semantic relationships among the word senses [32]. Using the same concept of WordNet, it is developed in many other languages such as Spanish, German etc. are available at present. Similarly, statistical and machine learning methods are being used for WSD purpose. Although Weaver was first to recognize the statistical possibility in WSD [33] first used corpus based WSD in statistical machine translation. After the development of SENSEVAL, the task of comparing and evaluating WSD system became very easier since SENSEVAL defeated inconsistencies in test words, annotators, sense inventories, and corpora.

2.2 WSD APPROACHES

The task of analyzing the context of word for machine is really very complex and challenging task [34, 35]. Moreover, a highly accurate WSD is extremely desired in many real world computational applications such as Semantic Mapping (SM), Machine Translation, Ontology Learning (OL), Speech Recognition (SR), Semantic Annotation (SA), Information Extraction (IE) and Information Retrieval (IR). To fulfill this desire, since 1950s, many approaches had been used to address the WSD problem. Depending upon the main sources of knowledge that the WSD approaches are using for the sense disambiguation, they are categorized into 1) dictionary-based or knowledge-based, 2) unsupervised approaches and 3) supervised approaches. There are also 4) semi-supervised approaches that takes the advantages of both supervised and unsupervised approaches. Those approaches which primarily uses the dictionary, thesauri, and lexical knowledge bases without using any corpus evidence for the disambiguation fall in the knowledge-based approaches. The remaining approaches such as supervised, unsupervised approaches utilizes on the corpus evidence for the sense disambiguation. The unsupervised approaches uses the completely un-annotated corpora without using any external information to collect the cross-linguistic evidence for sense disambiguation. The supervised approaches uses the sense annotated corpora for sense disambiguation. The semi-supervised approaches uses the annotated corpora as only a seed data which then help to gather the evidence information from the un-annotated corpora for the further sense disambiguation.

2.2.1 Corpus Based Approaches

The corpus based WSD approaches uses the information evidence from corpus sense disambiguation. The corpus may be sense-tagged (annotated) or raw corpora [35]. Depending on whether the corpus used is sense-tagged or not, a corpus-based approach can be a supervised or unsupervised.

Supervised method uses the corpus that is sense-tagged. The rationale behind the supervised method is that a context itself can provide enough evidence to disambiguate a word in that context [36]. In this approach, using an already available disambiguated corpus where each ambiguous word is sense-tagged for training, it correctly disambiguates new ambiguous word [37]. These approaches collect examples, classify events, determine the patterns in the classifications and generalize the patterns using some rules. Then the derived rules are then used to classify where a new event belongs to.

Thus, these methods learn from the corpus which is previously sense-tagged. This method is most accurate but the main problem of this approach is that the machine-learning classifiers are trained examples which are manually annotated. This is very expensive to develop in terms of time and effort as well [38] and this leads to knowledge acquisition bottleneck[3]. Although the supervised methods are found to be more accurate, it is extremely costly and time consuming as they need manual sense-tagging. In addition, only very few sense-tagged corpora are in existence. To defeat the difficulty in supervised method, unsupervised methods that can automatically obtain sense-tagged training corpus are being proposed and are supposed to remove knowledge acquisition bottleneck, since these unsupervised methods do not need manual effort [39].

Unsupervised methods do not require sense-tagged corpora. These methods presume the similar contexts contain similar senses. Therefore, using context similarity measure, they cluster word occurrences from corpora and classify then the new word's occurrences to the most appropriate clusters (or senses).

In [38], author had discussed the various reasons for lower accuracy of unsupervised methods with respect to supervised one. To the opposition of [38], in [37], author has concluded that his unsupervised method achieved nearly same performance (95.5% vs. 96.1%) with compared to supervised one. His algorithm in [37], presumes one word exhibits one sense in a given context.

[3] The knowledge acquisition bottleneck can be defined as the problem which occurs when one could not be able to obtain sufficient knowledge as requirement.

From [3], it is known that only very few sense-tagged corpora are in existence. Due to this fact, there is obviously lack of training data which are already sense-tagged. Therefore, many WSD algorithms have used semi-supervised learning method utilizing both sense-tagged and raw data/corpora. This learning method first bootstrap starting with few data which are manually sense-tagged. These are used to train classifier. The classifier then used to train remaining part of untagged corpus. This process is repeated until the whole corpus is trained. The *bootstrapping algorithm* described in [37] is an example of a semi-supervised approach.

2.2.2 Knowledge Based Approaches

The manually tagged corpuses which are used by supervised and semi-supervised methods are very costly as it require much of time and effort. Moreover, all these requires considerable amount of time to develop a classifier to train the raw corpora and only very few sense-tagged corpora are in existence. These problems make knowledge based methods re-emerge as an alternative solution for WSD problem [40]. The rationale behind the knowledge-based approaches is to take advantage of lexicon resources for word sense disambiguation.

2.3 ENGLISH WORDNET

WordNet is a lexical resource of English language. It groups words together based on sets of synonyms called synsets. Each of these synsets expresses a different concept. The words which express the same or similar concepts and can be interchangeably used in many contexts form a synset. A dictionary organized the words based on alphabetical order. That is the dictionary groups the words that are spelled alike. WordNet, in contrast, organizes words based on word’s meaning. That is the words that express similar meaning are grouped into a synset [41] [42]. The development of WordNet was started in 1985 at Cognitive Science Lab of Princeton University. The project leader was George A. Miller [41].

2.3.1 Motivation to WordNet

The dictionaries put the words (lexical information) using alphabetical order. The words which have similar spelling, come together. But this cause to scatter the words which have similar meanings [43]. For instance, the synonym words “put” and “arrange” are scattered in the dictionary. However, the words “pustule” and “put” are arranged together. Therefore, in dictionary, the task of searching the words which have similar meaning is difficult and time consuming.

One alternative to make fast search of words is to implement the dictionary in computer database. This will obviously decrease the searching time by greatly: a word can be found while user is typing a word. However, the problem is still unsolved: - the words which spell alike are grouped together and the words with similar or same meanings are still scattered in the dictionary list.

To resolve these problems, a project containing a group of psychologist and linguistics in Princeton University was formed in 1985. The aim of the project was to develop a lexical database so that the dictionary can be searched conceptually but not alphabetically [41]. The output of the project is the WordNet.

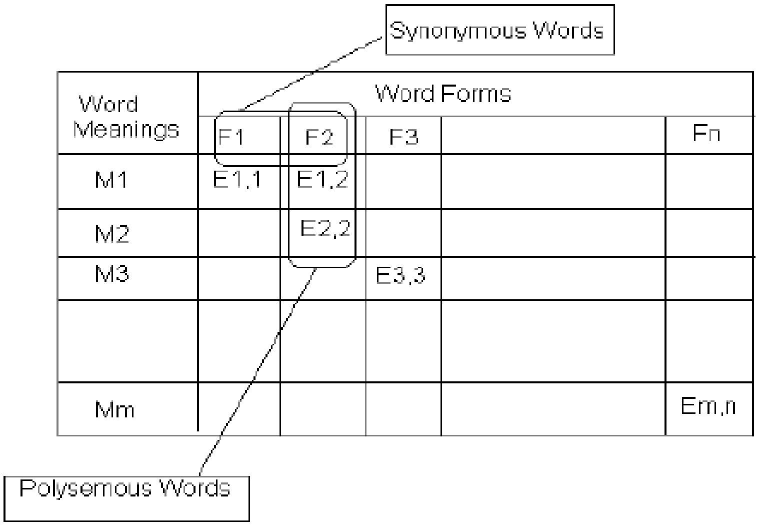
Since the English WordNet contains more information about a word, it is massively used in knowledge based WSD approaches and became popular. Due to its popularity, many WordNets on other languages like Italian, German, Spanish, and Hindi were built. The knowledge based WSD approaches uses the information of various relations (such as hypernyms, hyponyms, meronyms etc.) in WordNet.

2.3.2 WordNet Structure

The WordNet arranges the words based on synset concept. It divided the words into the nouns, verbs, adjectives, adverbs and function words and then arranged together into sets of synonym [44]. The category function words contain relatively few words and therefore this category is kept separately. The synsets are interlinked by using conceptual semantics and various lexical relations. The nouns are organized as hierarchies. The verbs are arranged as entailment relations. The adverbs and adjectives are arranged as n-dimensional hyperspaces. This type of categorization introduce a redundancy in the WordNet since a word “put”, for example, is found in more than one category.

2.3.3 Lexical Matrix

Synsets are used to represent lexical concepts in lexical matrix. In lexical matrix, words and their senses are bound together. It is shown in Figure 2.1 [45]. In the table, column head denotes the forms of word. The row head denotes meanings of word. A value E1.1 indicates the word form F1 has meaning M1. A word form is polysemy if a column has multiple values for that word form. Word are synonymous if a row has multiple values.



Word Meanings	Word Forms				
	F1	F2	F3		Fn
M1	E1,1	E1,2			
M2		E2,2			
M3			E3,3		
Mm					Em,n

Synonymous Words
Polysemous Words

Figure 2. 1: Lexical Matrix illustrating synonym words (F1, F2) and polysemy word (F2)

The words F1 and F2 are synonyms since the meanings (E1,1 and E1, 2) of the two word forms F1 and F2 conclude the same meaning M1. In the other hand, the same word F2 which has meanings E1,2 and E2,2 corresponds to different meanings M1 and M2 respectively. Therefore F1 is Polysemy word.

2.3.4 Semantic Relations

WordNet arranges the words based on synsets. Each word in a synset expresses a concept. These synsets are connected each other using various semantic relations such as hypernym, hyponymy, holonymy, meronymy, and entailment etc. [46]. The semantic oppositions are connected with antonym relations. WordNet also deals with morphological level information of words.

2.3.5 Parts of Speech

The words in WordNet are first divided into four groups based on parts of speech- nouns, verbs, adverbs and adjectives.

Nouns

Nouns are hierarchically organized into different levels from specific to generic. The top most, or most generic level of the hierarchy is almost empty semantically. The inheritance hierarchies seldom go more than ten levels deep.

The distinguishing features are normally found somewhere in the middle level called the base level of the noun. The noun mostly contain the relations such as hypernyms, hyponyms, meronyms, holonyms and antonyms. The most common relation is hyponymy/hypernymy. Another major noun's organizer is meronymy/holonymy relation.

Verbs

The number of verbs is very low compared to the nouns. The verbs are found to be more polysemous as compared to nouns. The WordNet has arranged verbs using fourteen 14 groups such as change, communication, consumption, creation, emotion, weather verbs etc. [47]. Verbs cannot be easily organized into hierarchy structures. The hierarchical levels cannot more than four. In addition, all verbs cannot be arranged under a single node [47].

The most commonly found relation in WordNet is troponymy. Any acceptable statement about part-relation among verbs always involves the temporal relation between the activities that the two verbs denote. One activity or event is part of another activity or event only when it is part of, or a stage in, its temporal realization.

The simultaneous activities like *fatten* and *feed*, including activities like *snore* and *sleep* or preceding activities like *try* and *succeed* are organized under lexical entailments. Another variation of entailment relation is causation and it is asymmetrical.

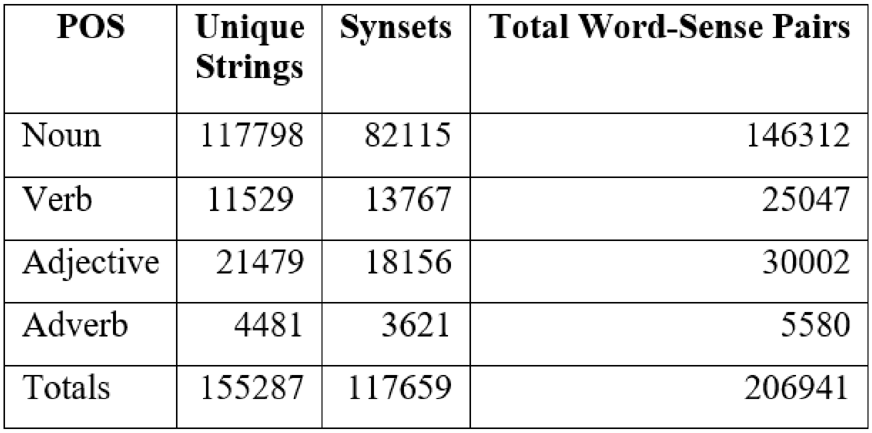
Adjectives

Adjectives are arranged in WordNet by dividing into two groups- descriptive and relational adjectives. The descriptive adjectives arranged as binary oppositions as Antonymy while the similarity of meaning arranged as synonymy. Relational adjectives are arranged by cross-referencing to the files of noun [48].

Current Statistics of WordNet

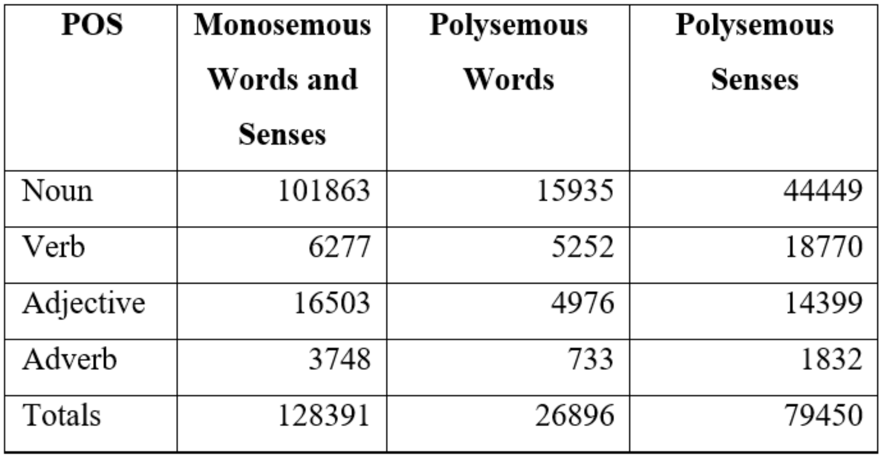
The statistics of WordNet 3.0 database [5]:

Table 2. 1: Number of words, synsets and senses



POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Table 2. 2: Polysemy information



POS	Monosemous Words and Senses	Polysemous Words	Polysemous Senses
Noun	101863	15935	44449
Verb	6277	5252	18770
Adjective	16503	4976	14399
Adverb	3748	733	1832
Totals	128391	26896	79450

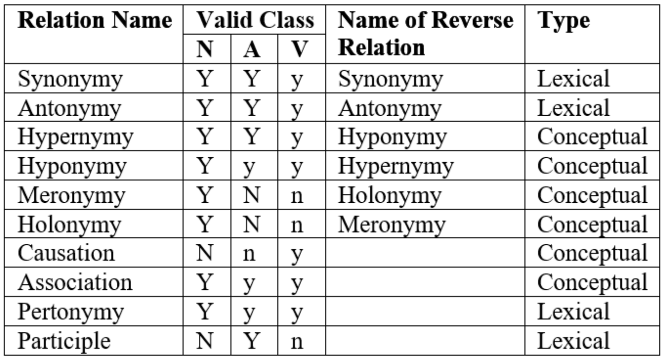
2.4 OTHER WORDNETS

This section presents a brief study of various WordNets such as German WordNet (GermaNet), EuroWordNet, Japanese WordNet, Chinese WordNet, Hindi WordNet, IndoWordNet and Nepali WordNet.

2.4.1 GermaNet: A German WordNet

GermaNet is a lexical resource developed for German language based on the principle of English WordNet. GermaNet arranges the words based on synset concepts dividing words into different parts of speech- noun, verb and adjective. Adverbs are not included in the current work phase in the GermaNet. These synsets are connected each other by using various semantic relations [49]. The words (noun, adjective and verb) are hierarchically structured by the hypernymy relation of synsets. The development of the GermaNet was started in 1997 University of Tübingen under the Division of Computational Linguistics of the Linguistics Department. **EuroWordNet (EWN)** has included the GermaNet**.** The GermaNet provides a facility to use as on-line thesaurus.

Table 2. 3: Relationships in GermaNet

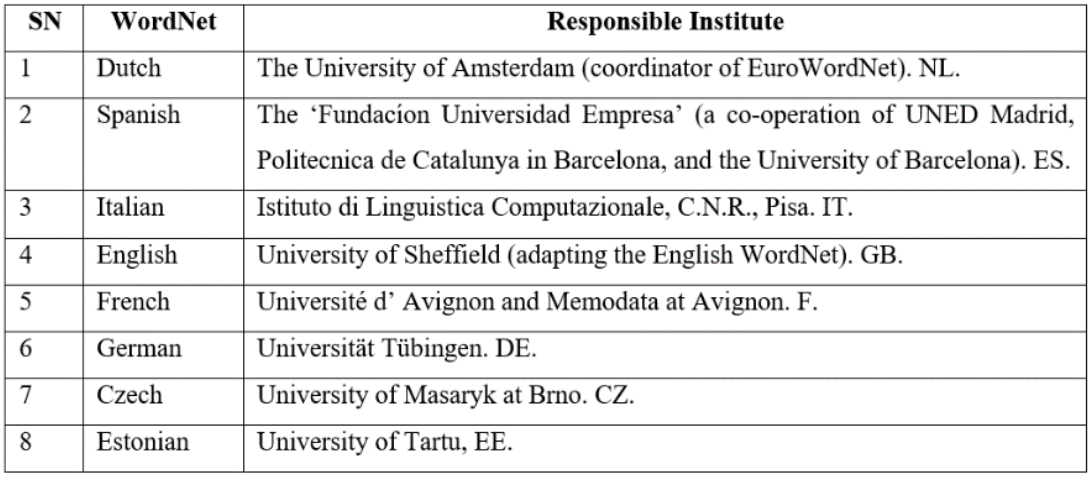


Relation Name	Valid Class			Name of Reverse Relation	Type
	N	A	V		
Synonymy	Y	Y	y	Synonymy	Lexical
Antonymy	Y	Y	y	Antonymy	Lexical
Hypernymy	Y	Y	y	Hyponymy	Conceptual
Hyponymy	Y	y	y	Hypernymy	Conceptual
Meronymy	Y	N	n	Holonymy	Conceptual
Holonymy	Y	N	n	Meronymy	Conceptual
Causation	N	n	y		Conceptual
Association	Y	y	y		Conceptual
Pertonymy	Y	y	y		Lexical
Participle	N	Y	n		Lexical

GermaNet organizes the words using various relationships like the WordNet do. The GermaNet contains all relations which are defined in WordNet to connect the words semantically except the adjective’s relation **'similar to'. This relation** in GermaNet is substituted by the **hyperonymy/hyponymy**. Multiple inheritance is allowed in GermaNet while it is not found in the WordNet. This multiple inheritance provides the cross-classification among the words in the GermaNet. The Table 2.3 lists all the relations that exists in GermaNet. The table also shows whether the relations are valid for the valid classes where N denotes Noun, A denotes Adjective and V denotes Verb.

2.4.2 EuroWordNet

Table 2. 4: WordNets in European Languages and Responsible Institutes

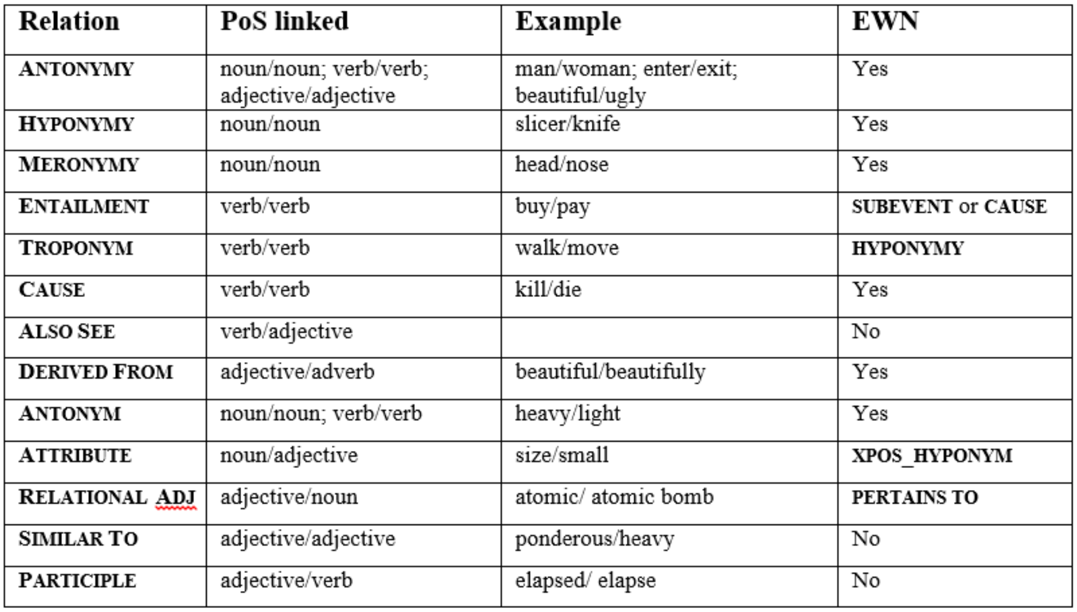


SN	WordNet	Responsible Institute
1	Dutch	The University of Amsterdam (coordinator of EuroWordNet). NL.
2	Spanish	The 'Fundacíon Universidad Empresa' (a co-operation of UNED Madrid, Politecnica de Catalunya in Barcelona, and the University of Barcelona). ES.
3	Italian	Istituto di Linguistica Computazionale, C.N.R., Pisa. IT.
4	English	University of Sheffield (adapting the English WordNet). GB.
5	French	Université d' Avignon and Memodata at Avignon. F.
6	German	Universität Tübingen. DE.
7	Czech	University of Masaryk at Brno. CZ.
8	Estonian	University of Tartu, EE.

The EuroWordNet (abbreviated as EWN) is a multilingual lexical database which stores and links the WordNets of eight European languages [50]. These eight languages includes English, German, Dutch, Spanish, Italian, French, Estonian and Czech. EWN was developed under the EuroWordNet project which was 3 years project. Initially, the EWN was developed for Italian, Spanish, Dutch and English. Later the project is extended and included French, German, Czech and Estonian. Individual WordNet in these languages are built in the similar way like the English WordNet developed at Princeton University. Each WordNet are slightly different in their structure depending upon the specific nature of the particular language. However, each WordNet has the synset expressing a distinct concept. A distinct concept in one language is interconnected with a semantically equivalent distinct concept in other language by using Inter-Lingual-Index (ILI). The individual WordNets in 8 European languages are being developed and maintained by the different institutes (see Table 2.4) [51].

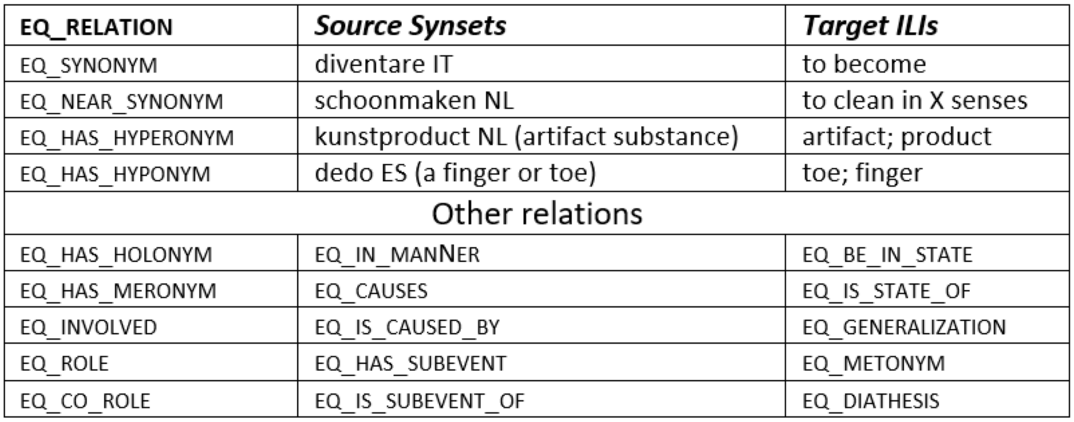
The main difference between EuroWordNet and the English WordNet is that the EuroWordNet is multilingual while the English WordNet is monolingual. The EuroWordNet has adapted the multi-linguality.

Table 2. 5: WordNet1.5 Relations



Relation	PoS linked	Example	EWN
ANTONYMY	noun/noun; verb/verb; adjective/adjective	man/woman; enter/exit; beautiful/ugly	Yes
HYPONYMY	noun/noun	slicer/knife	Yes
MERONYMY	noun/noun	head/nose	Yes
ENTAILMENT	verb/verb	buy/pay	SUBEVENT or CAUSE
TROPONYM	verb/verb	walk/move	HYPONYMY
CAUSE	verb/verb	kill/die	Yes
ALSO SEE	verb/adjective		No
DERIVED FROM	adjective/adverb	beautiful/beautifully	Yes
ANTONYM	noun/noun; verb/verb	heavy/light	Yes
ATTRIBUTE	noun/adjective	size/small	XPOS_HYPONYM
RELATIONAL ADJ	adjective/noun	atomic/ atomic bomb	PERTAINS TO
SIMILAR TO	adjective/adjective	ponderous/heavy	No
PARTICIPLE	adjective/verb	elapsed/ elapse	No

Table 2. 6: The Equivalence Relations in EuroWordNet



EQ_RELATION	*Source Synsets*	*Target ILIs*
EQ_SYNONYM	diventare IT	to become
EQ_NEAR_SYNONYM	schoonmaken NL	to clean in X senses
EQ_HAS_HYPERONYM	kunstproduct NL (artifact substance)	artifact; product
EQ_HAS_HYPONYM	dedo ES (a finger or toe)	toe; finger
Other relations		
EQ_HAS_HOLONYM	EQ_IN_MANNER	EQ_BE_IN_STATE
EQ_HAS_MERONYM	EQ_CAUSES	EQ_IS_STATE_OF
EQ_INVOLVED	EQ_IS_CAUSED_BY	EQ_GENERALIZATION
EQ_ROLE	EQ_HAS_SUBEVENT	EQ_METONYM
EQ_CO_ROLE	EQ_IS_SUBEVENT_OF	EQ_DIATHESIS

The Table 2.5 shows relations that are used in the English WordNet 1.5. This table compares the relations with EuroWordNet (EWN) along with the PoS and the example. To link one concept with the equivalent concept in another language, the EuroWordNet uses various equivalence relations. These are listed in the Table 2.6.

2.4.3 Japanese WordNet

Inspired with the English WordNet and the Global WordNet, the development of Japanese WordNet was started in 2006 by NICT (National Institute of Information and Communications Technology) [52]. It is built in the similar way the English WordNet at Princeton University. Japanese WordNet contains the Japanese equivalents to English synsets. The first version was released in February 2009. All relations in Japanese WordNet are borrowed from English WordNet 3.0 [53]. The synsets are further enriched with Japanese translations of the definitions, examples and lemmas. Wn-Ja 1.1 is released in 22 October 2010. The contributors of this WordNet provided the illustrations for each concept by linking the synset with images from Open ClipArt Library.

After the development of the Japanese WordNet, it is being used in various applications such as Weblio Online Japanese/English Dictionary, Japanese Reading Practice, Japanese Thesaurus Android and Japanese-English Thesaurus iPhone application App.

The contributors[4] of Japanese WordNet are Hitoshi Isahara, Francis Bond, Kow Kuroda, Kyoko Kanzaki, Kiyotaka Uchimoto, Takayuki Kuribayashi, Darren Cook, Masao Utiyama, Kentaro Torisawa and Asuka Sumida.

2.4.4 Chinese WordNet

There are three Chinese WordNets that have been developed based on the principles of English WordNet. These three Chinese WordNets includes Southeast University WordNet (SEW), Taiwan University WordNet (CWN) and Sinica Bilingual Ontological Wordnet (BOW). The Chinese WordNet SEW is in simplified Chinese, while the other Chinese WordNets BOW and CWN are in traditional Chinese. Bootstrapping method is used to create

[4] http://nlpwww.nict.go.jp/wn-ja/index.en.html

BOW [54]. SEW was created automatically by using Intersection, Words Co-occurrence and Minimum Distance approaches [55] [56]. English WordNet 3.0 is translated into Chinese WordNet using these three approaches. A Chinese WordNet- CWN is developed by Taiwan University and Academia Sinica [57].

The Academia Sinica Bilingual Ontological Wordnet (Sinica BOW) was built by integrating the SUMO (Suggested Upper Merged Ontology), the English-Chinese Translation Equivalents Database (ECTED) and WordNet. It is built at the Institute of Linguistics and the Institute of Information Science of Academia Sinica. These were first linked in two pairs. The WordNet 1.6 was first manually linked/mapped to SUMO in first pair [58] and it was again manually linked/mapped to ECTED (the English lemmas in WordNet were mapped to their Chinese lexical equivalents) in second pair. Therefore, it can be used as an English-Chinese bilingual WordNet. In addition, it can be used to access SUMO as a bilingual lexical resource. The Sinica BOW, however, has many un-lexicalized entries in Chinese. This problem is resolved in Chinese WordNet (CWN) created at Taiwan University. The Chinese WordNet has only entries for Chinese words [56]. All these Chinese WordNets are developed just as like the English WordNet.

2.4.5 Hindi WordNet

Hindi WordNet (HWN) is a lexical resource developed for Hindi language at Centre for Indian Language Technology (CFILT), IIT Bombay. This project was led by Pushpak Bhattacharyya [59]. Hindi WordNet is also inspired with and was built following the principle of the English WordNet. The development of Hindi WordNet was started from 2000 and was publicly available in 2006 [60]. Hindi WordNet also divided the words into four PoS categories and the words are semantically related with the synset as in the English WordNet.

In the ontology, each concept contains synset, word’s gloss and position. The words in the synset are organized based on used frequency. The gloss describes the concept with example. In addition, the synset concept is described by the position in ontology as well by mapping into some place in the ontology. An ontology is a hierarchical organization of concepts. Each syntactic category- noun, or verb or adjective or adverb has a separate ontological hierarchy. The ontology of a concept school synset is shown in Figure 2.3.

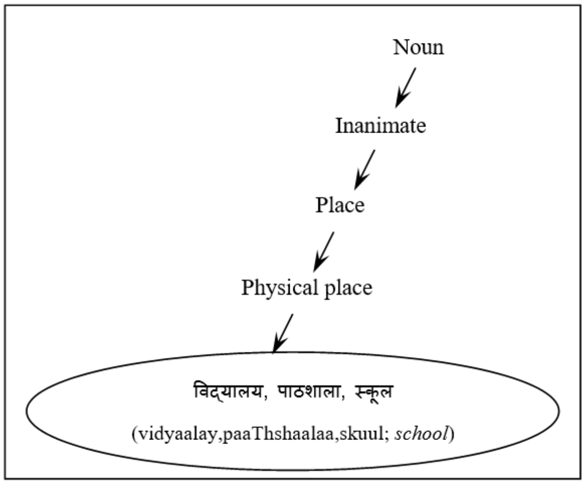
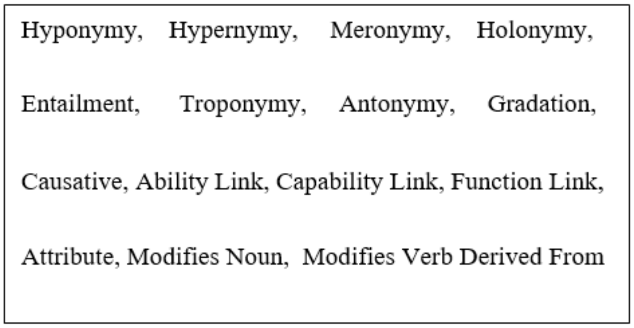


Figure 2. 2: Ontology of *school* synset

Relations in Hindi WordNet

Like in English WordNet, synset is the basic element of the HWN and it expresses a distinct concept. These synsets are connected by the various relations such as hypernymy, holonym, hyponymy, meronymy, troponymy, antonyms or entailment. The HWN altogether contains sixteen different relations to connect the different concepts in the hierarchy as shown in Figure 2.3.



Hyponymy, Hypernymy, Meronymy, Holonymy,

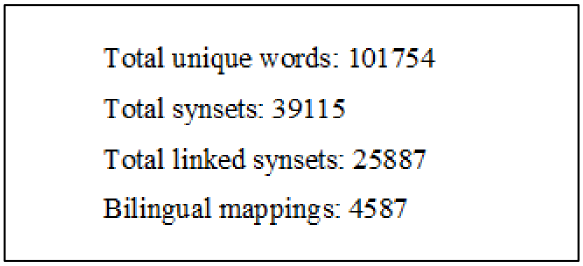
Entailment, Troponymy, Antonymy, Gradation,

Causative, Ability Link, Capability Link, Function Link,

Attribute, Modifies Noun, Modifies Verb Derived From

Figure 2. 3: Relations in Hindi WordNet

The Figure 2.4 shows the statistics of Hindi WordNet when it is accessed on 10th of Feb, 2016.



Total unique words: 101754

Total synsets: 39115

Total linked synsets: 25887

Bilingual mappings: 4587

Figure 2. 4: Statistics of Hindi WordNet (10 Feb 2016)

2.4.6 IndoWordNet

IndoWordNet is a multilingual resource which connects WordNets in 18 Indian languages [61]. These languages are 1) Assamese, 2) Bangla, 3) Bodo, 4) Gujarati, 5) Hindi, 6) Kannada, 7) Kashmiri, 8) Konkani, 9) Malayalam, 10) Manipuri, 11) Marathi, 12) Nepali, 13) Oriya, 14) Punjabi, 15) Sanskrit, 16) Tamil, 17) Telugu and 18) Urdu. The development of Hindi WordNet was started from 2000 and is first publicly available from 2006.

Table 2. 7: WordNets in IndoWordNet

SN	Name of WordNet	Number of Synsets	Name of Institute
1	Assamese	14958	Guwahati University, Guwahati, Assam
2	Bengali	36346	Indian Statistical Institute, Kolkata, West Bengal
3	Bodo	15785	Guwahati University, Guwahati, Assam
4	Gujarati	35599	Dharamsinh Desai University, Nadiad, Gujarat
5	Hindi	38607	IIT Bombay, Mumbai, Maharashtra
6	Kannada	20033	Mysore University, Mysore, Karnataka
7	Kashmiri	29469	Kashmir University, Srinagar, Jammu and Kashmir
8	Konkani	32370	Goa University, Taleigao, Goa
9	Malayalam	30060	Amrita University, Coimbatore, Tamil Nadu
10	Manipuri	16351	Manipur University, Imphal, Manipur
11	Marathi	29674	IIT Bombay, Mumbai, Maharashtra
12	Nepali	11713	Assam University, Silchar, Assam
13	Oriya	35284	Hyderabad Central University, Hyderabad, Andhra Pradesh
14	Punjabi	32364	Thapar University and Punjabi University, Patiala, Punjab
15	Sanskrit	23140	IIT Bombay, Mumbai, Maharashtra
16	Tamil	25431	Tamil University, Thanjavur, Tamil Nadu
17	Telugu	21925	Dravidian University, Kuppam, Andhra Pradesh
18	Urdu	34280	Jawaharlal Nehru University, New Delhi

After the development of the Hindi WordNet, WordNet in other languages of India were built by expanding approach with the Hindi WordNet and are linked each other to form IndoWordNet. IndoWordNet is similar to the EuroWordNet. The one difference is that in IndoWordNet, Hindi provides the Interlingual Index (ILI) while in EuroWordNet, English provides the ILI.

The WordNets in various languages of India are developed in different institutions [60]. The Table 2.7 shows the list of wordnets included in IndoWordNet along with the number of synsets (As of August 2014) and creating institutes' name.

2.4.7 Nepali WordNet

Nepali WordNet was built at Assam University, India [62, 63]. It was developed as a part of a Consortium Project led by IIT Bombay. The Nepali WordNet was created based on the Hindi and English WordNet using expansion approach. In Expansion approach, the lexicographer is known about the synset in the preexisting WordNet and the lexicographer creates the equivalent synset for the new WordNet to be created in new language. Due to high similarity between the Hindi and Nepali, Hindi WordNet was taken as pivot for building of the Nepali WordNet. Like English and Hindi WordNet, Nepali WordNet (NWN) organizes the Nepali words into synset each expressing the distinct concept. The Nepali WordNet has also included all the four parts of speech. The synsets concepts are inter-linked with the various relations like Hypernymy, meronymy etc.

The Nepali WordNet is included in the IndoWordNet multilingual database since Nepali[5] language is the official language of India. The NWN contains 5802 synsets and 10278 unique words in its database as on March 1, 2010.

[5] **Nepali** is an Indo-Aryan part of the Indo-European language family and is official language of Nepal. Nepali language is spoken by a significant amount of people also in Bhutan, Burma and India. Nepali language is also an official of Sikkim and Darjeeling, India. It is also used and spoken in Uttaranchal and Assam of India. As per

2.5 WORDNETS AND THEIR COMMON FEATURES

In section 2.4, the WordNets in various languages are presented. These WordNets are being used as a key resource in various natural language processing tasks. The English WordNet at Princeton University, is also known as Princeton WordNet (PWN).

The PWN organizes the words dividing them into four parts of speech into synonymy set called synset. A synset contains the words which have the similar meaning and can be used by interchanging each other without changing the meanings. These synsets in the PWN provide the distinct concepts. These distinct concepts i.e. synsets in PWN are connected via various relationships such as hypernymy, meronymy, antonymy and more. The synset are the means of expressing the semantic relations among the words in the WordNet. These semantic relations are mainly of two types. These include lexical relation such as antonymy and conceptual relations such as hypernymy, meronymy or entailment.

After the development of the WordNet, it is massively used as a key resource in the Machine Translation and Word Sense Disambiguation. With the success of PWN in Natural Languages Processing of English language, other WordNets were developed by following the same principle of the PWN with a little or no modification to adapt to the particular natural language. The development of the various WordNets such as GermaNet, EuroWordNet, Japanese WordNet, Chinese WordNet, Hindi WordNet, IndoWordNet and Nepali WordNet are described in previous section 2.4. After the brief review on these WordNets, it is noticed that all of these are developed using the underlying principle of Princeton WordNet. It is found that all WordNets organize the words into synset to express a distinct concept in the WordNet. It is also found that these synsets in all WordNets are interconnected by the various relations such as hypernymy, hyponym, holonym, meronymy, antonymy, entailment etc. as in Princeton WordNet. The only difference found from this study is that each WordNet are using some less

census 2001, there are 17 million speakers of Nepali within Nepal. The website of Ethnologue shows more than forty two million (2012) worldwide speakers.

or additional relations in their WordNet to better fit and adapt in their particular languages like in Hindi WordNet where some extra relations such as ability-link and capability-link are defined to better fit for Hindi language. It is also found that some WordNets are developed by just translating each synset in the English WordNet into other languages like the development of the Chinese WordNet.

Machine Translation uses multilingual WordNets such as EuroWordNet [50], IndoWordNet [60], Asian WordNet [64] and MultiWordNet [65]. The Multilingual WordNets links synsets in one language with equivalent synsets in other language. This features of Multilingual WordNet makes easier in translating one language into another. The most important point here is that these multilingual WordNets are also being built in the concept of the English WordNet. Finally, it is concluded that all the WordNets in the World are developed based on the same principle, concept and the structure of the English WordNet. They share the common structure and common relations to organize the words in the WordNets copying from the English WordNet. The only difference is that the other WordNets are being built with a little modification in structure and more or less modification in relations to organize the word in their WordNet to adapt and to better fit in their WordNet.

2.6 USE OF WORDNET IN WORD SENSE DISAMBIGUATION

During 1980s, dictionary-based WSD was started with Lesk’s algorithm which used *Oxford Advanced Learner’s Dictionary’s word definitions* to determine the overlap in word definitions to disambiguate senses. Afterward, dictionary-based approaches such as [30] and [31] are tried.

In 2002, Banerjee and Pedersen used the original Lesk algorithm with some modification and used information from WordNet instead from Dictionary to count the overlaps with context [13]. They utilized information from hypernymy, hyponymy, torponymy, holonymy, meronymy of verbs and nouns. They observed 32% of accuracy when tested their system by Senseval-2.

In 2003, Patwardhan generalized Adapted Lesk algorithm using approach of semantic relatedness [66] [67]. They utilized "is-a" relation from WordNet for measuring semantic relatedness. They calculated overlaps on extended gloss. The extended gloss means they looked overlaps not only on glosses of synsets but also looked on glosses of synsets, hypernyms, meronym, hyponym, tropony and holony [68].

Sinha et al. (2003) developed automatic WSD which used Hindi WordNet for Hindi word sense disambiguation [69]. To determine the sense of noun words, they used statistical method using simple overlap count method. They observed the accuracy from 40% to 70%.

In 2003, Fragos et al. formed sense bags from the definition of all the hypernyms of nouns and verbs which are in the sense definition [14]. In the same way, they formed context bags using same relations from WordNet for all words present in context. They also tested the effect of inclusion of information from hyponym in their system and observed no improvement in the accuracy with this inclusion.

In 2005, Shuang Liu et al. utilized information from WordNet’s relations like synsets, hyponyms and hypernyms to find the correct sense [17]. Various WSD approaches including [15], [16], [18], [63], [70], [71], [72], [73] utilized information of WordNet’s synonym, hyponym and hypernym for sense disambiguation.

2.7 RELATED TASK

This research work is motivated from the problems raised due to the common information used from the hypernymy relation in WordNet for overlap-count knowledge-based approaches. When the similar cases are analyzed, it is found that the common information in sense bags are causing formation of noise information. This noise information is found to produce more overlaps for wrong sense causing the overlap-count knowledge-based WSD approaches to fail.

From this problem, an assumption is developed to use only that words is sufficient for sense disambiguation by avoiding the common information for senses and this modification will increase the accuracy of knowledge-based overlap-count WSD approaches. To examine this assumption, sample Nepali WordNet developed by Dhungana and Shakya is modified [18]. The Nepali WordNet is modified in such a way that new modified version arranged polysemy word's senses connected with only clue words. The clue words of a sense of polysemy word are the related words for the sense. The settings of all experiment of [18] are constant as they are. The results of experiments on 209 Nepali words when tested by 201 Nepali test sentences show the accuracy of 91.54% which was better than that of the accuracy found in [18] by 3.49%.

Chapter 3

Problem Statement and Solution Approach

In this chapter, the research problem and solution approach are discussed in detail. Before to describe the problem statement and the solution approach in detail, the use of WordNet in the knowledge-based WSD approaches are presented in detail, the problems in those WSD approaches are investigated and finally a novel solution approach is presented.

3.1 KNOWLEDG-BASED WSD METHODS

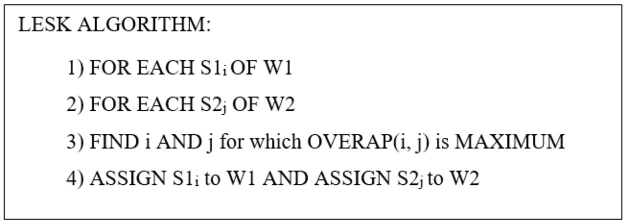
Knowledge-based approaches take advantage of dictionaries, collocation, ontology, thesauri etc. for disambiguating the sense of a polysemy. All the word sense disambiguation methods which primarily rely on these lexical resources, instead of using any corpus evidence are known as knowledge-based methods. The knowledge-based methods can be mainly categorized into four groups. The first group includes the WSD methods which use contextual overlap count method. The second group utilizes the similarity measures calculated on semantic networks. The third group WSD method uses the selectional preferences for sense disambiguation. The fourth group of WSD methods use the heuristic-based methods.

3.1.1 Contextual Overlap Count Methods

The contextual overlap methods count the contextual overlap of context words with respect to dictionary definitions. The idea behind this method is that the sense of a polysemy word which have the highest count of overlaps is the correct sense. The Lesk algorithm is one example of WSD method which uses contextual overlap count.

Lesk Algorithm

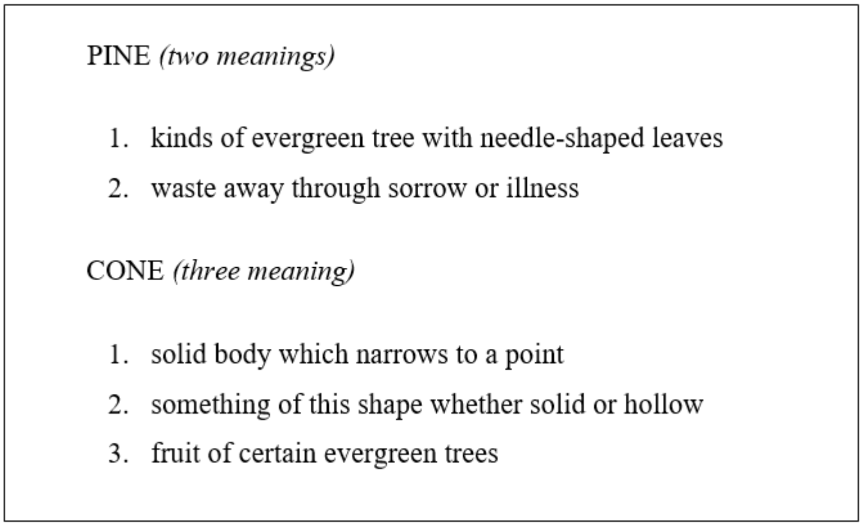
Lesk algorithm was developed by Michael Lesk. He used *Oxford Advanced Learner’s Dictionary* for contextual overlap count of word definition to disambiguate the word senses. During 1980s, dictionary-based WSD was started with the use of Lesk algorithm. The rationale of his algorithm is to count the numbers of overlaps between context words and with the definition of each sense of polysemy word. The sense of the polysemy word which shows highest number of overlap is the correct meaning.



LESK ALGORITHM:

1) FOR EACH $S1_i$ OF W1
2) FOR EACH $S2_j$ OF W2
3) FIND i AND j for which OVERAP(i, j) is MAXIMUM
4) ASSIGN $S1_i$ to W1 AND ASSIGN $S2_j$ to W2

Figure 3. 1: Lesk Algorithm



PINE *(two meanings)*

1. kinds of evergreen tree with needle-shaped leaves
2. waste away through sorrow or illness

CONE *(three meaning)*

1. solid body which narrows to a point
2. something of this shape whether solid or hollow
3. fruit of certain evergreen trees

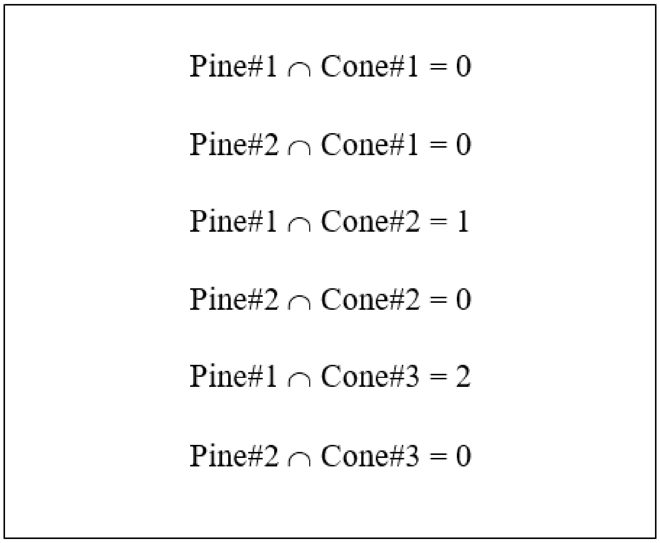
Figure 3. 2: Meanings of words "Pine" and "Cone"

The Figure 3.1 shows the original Lesk Algorithm. For the two words W1 and W2, let $S1_i$ is the ith sense of word W1 and $S2_j$ is the jth sense of the word W2 where i = 1, 2, 3, 4, ... n, j = 1, 2, 3, 4, ... m. The n and m are the number of senses of the words W1 and W2 respectively.

Then, for each $S1_i$ of word W1 and $S2_j$ of word W2, the OVERLAP(i, j) is calculated. If the OVERLAP(i, j) is maximum, then the sense $S1_i$ is assigned to word W1 and $S2_j$ is assigned to the word W2.

To illustrate this algorithm, let us take an example which shows the sense disambiguation of two words "cone" and "pine". The definitions for the words "pine" and "cone" taken from the dictionary are shown in Figure 3.2. The word "pine" has two meanings and the word "cone has three meanings.

In Figure 3.2, the word "pine" has two meanings: 1) "*kinds of evergreen tree with needle-shaped leaves*" (say Pine#1) and 2) *"waste away through sorrow or illness"* (say Pine#2). Similarly, "cone" has three meanings: 1) *"solid body which narrows to a point"* (say Cone#1), 2) *"something of this shape whether solid or hollow"* (say Cone#2) and 3) *"fruit of certain evergreen trees"* (say Cone#3). Now, the overlaps of words between each meaning of words "pine" and "cone" are calculated as shown in Figure 3.3. The pair (Pine#1, Cone#2) has one overlap and the pair (Pine#2, Cone#3) has two overlaps. The overlaps between other pairs are zero.



Pine#1 $\cap$ Cone#1 = 0

Pine#2 $\cap$ Cone#1 = 0

Pine#1 $\cap$ Cone#2 = 1

Pine#2 $\cap$ Cone#2 = 0

Pine#1 $\cap$ Cone#3 = 2

Pine#2 $\cap$ Cone#3 = 0

Figure 3. 3: Overlaps of words between each meaning of "pine" and "cone".

In Figure 3.3, there is the highest Overlaps of words between the definitions of the meaning no 1 of PINE (Pine#1) and meaning no 3 of CONE (Cone#3) and it is two. Therefore, the correct meaning of CONE when it is used with PINE is meaning no 3 and the correct meaning of PINE when it is used with CONE is the meaning no 1.

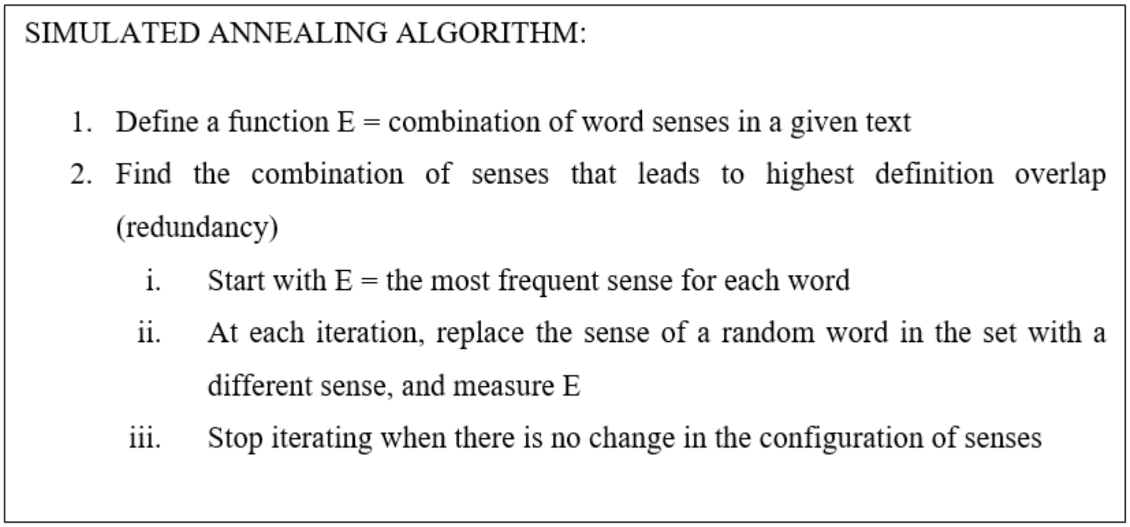
Variations on Lesk Algorithms

After the original Lesk algorithm, various modifications on this algorithm gave rise to several modified Lesk algorithms. Some of these includes simulated annealing, simplified Lesk algorithm and adapted Lesk.

Simulated Annealing

The original Lesk algorithm disambiguates the meanings of two polysemy words at a time. A sentence or context can have more than two polysemy words. Let us see what will happen if a sentence has more than two polysemy words whose meanings are to be disambiguated. Suppose a compound sentence- *"I saw a man who is 98 years old and can still walk and tell joke."* from [2]. In this sentence, the words- see (26), man (11), year (4), old (8), can (5), still (4), walk (10), tell (8), joke (3) have multiple senses. The number enclosed inside the parenthesis after each word shows the number of multiple senses or meanings of the word. The possible combinations of the senses of these nine words will be 43,929,600 in total and this huge number of combinations are very difficult to keep record of overlaps count of each pair. To overcome this problem by finding the optimal number of sense combinations is the simulated annealing by Cowie [74].

The idea is to define a function E as inverse of redundancy and to minimize the value of this function. The simulated annealing works as follows: for a given context containing multiple polysemy words, the possible sense combinations are found. The most frequent sense of each word is taken. Iterations are then performed to test sense of each word becomes constant to it reduce E’s value. The Figure 3.4 shows the simulated annealing version of Lesk algorithm which works for the context containing more than two polysemy words. This algorithm takes the advantage of using most frequent senses of words so that the possible numbers of sense combinations can be reduced. At each iteration, the sense of a random word in the set are replaced with a different sense and E whose value must be optimum is measured. The iteration stops only when there is no change in the combinations of senses.



SIMULATED ANNEALING ALGORITHM:

1. Define a function E = combination of word senses in a given text
2. Find the combination of senses that leads to highest definition overlap (redundancy)
   i. Start with E = the most frequent sense for each word
   ii. At each iteration, replace the sense of a random word in the set with a different sense, and measure E
   iii. Stop iterating when there is no change in the configuration of senses

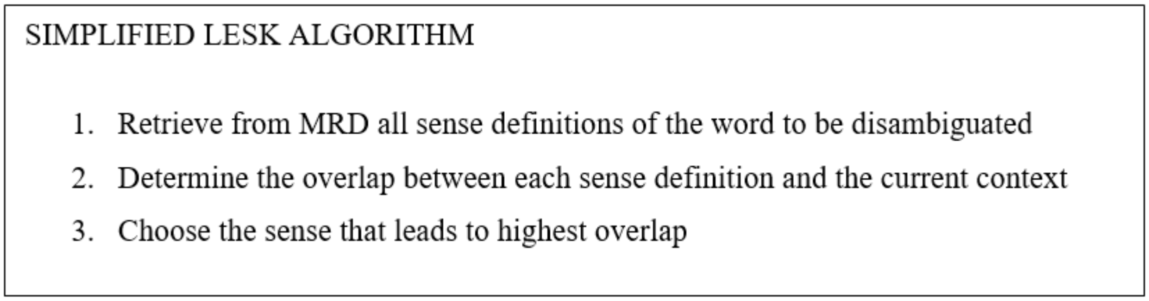
Figure 3. 4: Simulated annealing algorithm

Simplified Lesk

The original Lesk algorithm and the simulated annealing version of the Lesk algorithm simultaneously determine the senses of all the polysemy words in a sentence. That is, all the possible combinations of polysemy words in a sentence are considered. This combinations may became very huge in number if the context contain many polysemy words in the context and creates problem to keep track of the overlap counts of each combination. The simulated annealing tried to resolve the problem of huge number of possible combination.

Another version is simplified Lesk algorithm [75]. In contrast to original Lesk algorithm and its simulated annealing version, the simplified Lesk algorithm determine the sense of one polysemy word at a time.

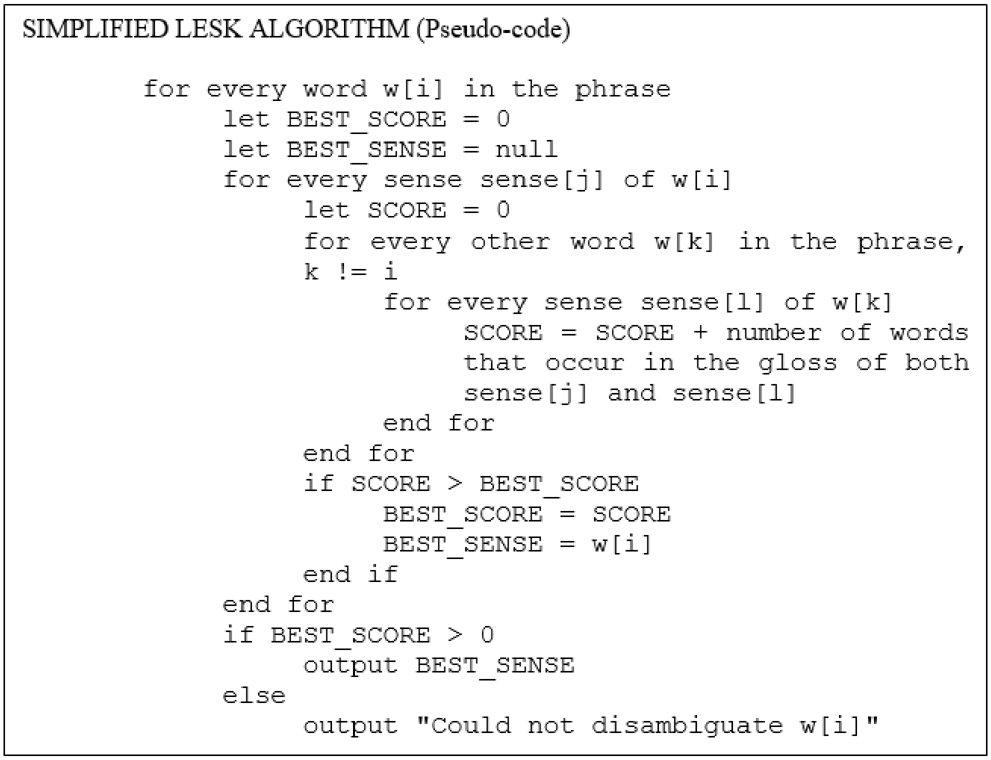
The simplified Lesk algorithm counts the number of overlaps between the words in context and definition of senses of polysemy word. The sense with maximum overlaps is the correct meaning of the polysemy word. Since this algorithm disambiguate a polysemy word at a time and the senses of remaining polysemy words are not considered, this significantly reduces the search space.



SIMPLIFIED LESK ALGORITHM

1. Retrieve from MRD all sense definitions of the word to be disambiguated
2. Determine the overlap between each sense definition and the current context
3. Choose the sense that leads to highest overlap

Figure 3. 5: Simplified Lesk Algorithm

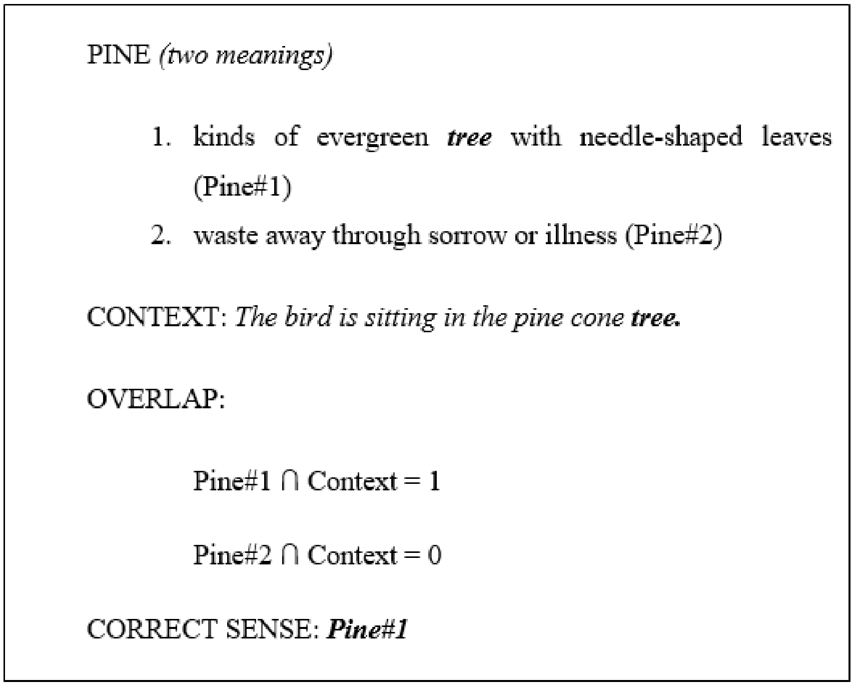


SIMPLIFIED LESK ALGORITHM (Pseudo-code)

```
for every word w[i] in the phrase
     let BEST_SCORE = 0
     let BEST_SENSE = null
     for every sense sense[j] of w[i]
          let SCORE = 0
          for every other word w[k] in the phrase,
          k != i
               for every sense sense[l] of w[k]
                    SCORE = SCORE + number of words
                    that occur in the gloss of both
                    sense[j] and sense[l]
               end for
          end for
          if SCORE > BEST_SCORE
               BEST_SCORE = SCORE
               BEST_SENSE = w[i]
          end if
     end for
     if BEST_SCORE > 0
          output BEST_SENSE
     else
          output "Could not disambiguate w[i]"
```

Figure 3. 6: Pseudo-code for simplified Lesk algorithm

The Figure 3.5 shows the simplified Lesk algorithm and it was proposed by Kilgarriff and Rosensweig. The algorithm first retrieves all sense definitions of a single polysemy word from MRD. The overlaps between context and each sense's definition is determined. The sense with highest overlap is chosen as the correct sense. The Figure 3.6 shows the simplified Lesk algorithm.



PINE *(two meanings)*

1. kinds of evergreen ***tree*** with needle-shaped leaves (Pine#1)
2. waste away through sorrow or illness (Pine#2)

CONTEXT: *The bird is sitting in the pine cone* ***tree.***

OVERLAP:

Pine#1 ∩ Context = 1

Pine#2 ∩ Context = 0

CORRECT SENSE: ***Pine#1***

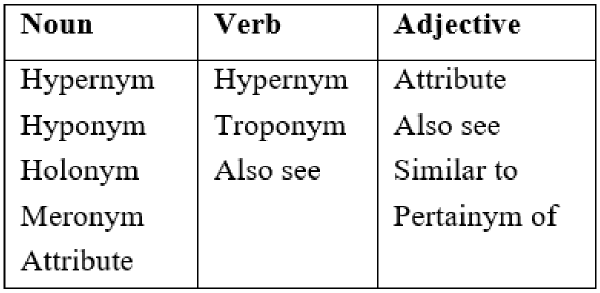
Figure 3. 7: Example of simplified Lesk algorithm

Suppose a context "The bird is sitting in the pine cone tree". Here, let the target word be "pine". There are two senses of word "pine" as shown in Figure 3.7. There is only one overlap of context with the definition of sense 1 (Pine#1) of the "pine" and no overlap of words with the sense 2 (Pine#2). The sense 1 of "Pine" has the maximum overlap counts. The sense Pine#1 is chosen as the correct sense for the context.

Adapted Lesk

Original Lesk algorithm uses definitions from dictionary. These definitions are very short. In addition, the dictionary does not provide the other relationships of words like the WordNet provides. The WordNet does not only provide the glosses of the words but also provide the other relations such as hypernymy, hyponymy etc. by which the words are interconnected semantically in the sunset. Banerjee and Pedersen (2002) adjusted the Lesk algorithm and used information from various relations in WordNet. They used WordNet instead of dictionary [13]. They also used a different overlap count method. They adjusted Lesk algorithm in various ways. They used the shorter context window of three words due to the computational reasons as in the simulated annealing version of Lesk algorithm. Their context window is centered on the target word. They used lexical database WordNet instead of using the dictionary. Another important modification is that they used the glosses of synsets that the word belongs to plus the glosses of synsets which are related by the relations such as attribute relation, Hypernyms, Hyponym, Troponym, Holonym, Meronym relations etc for both nouns and verbs. They avoided the use of the relations cause and entailment since there are less than 2% of links for the verbs. Similarly they did not use the antonymy relation. The Table 3.1 shows the relations that they use to disambiguate the senses for different parts of speech.

Table 3. 1: Relations utilized by adapted Lesk



Noun	**Verb**	**Adjective**
Hypernym	Hypernym	Attribute
Hyponym	Troponym	Also see
Holonym	Also see	Similar to
Meronym		Pertainym of
Attribute		

The original Lesk algorithm uses a single-token overlap counting method. For example, for the two sentences: 1) "Maria is writing poem with pen." and 2) "Writing poem is his hobby.", the original Lesk algorithm counts three overlaps for words "is", "writing" and "poem". However, the adapted Lesk used a multi-token overlap counting method. A gloss pair with an n-token overlap is given a higher score.

The score is obtained by getting sum of squared value of the no of tokens. For example, let us take the same two sentences. For these two sentences, the overlap is counted as: the overlapped single word token "is" is assigned a value 1^2 and the overlapped two word token "writing poem" is assigned a value 2^2. Thus, the overall score for this overlap counting is $1^2 + 2^2 = 1 + 4 = 5$. This is illustrated in Figure 3.7.

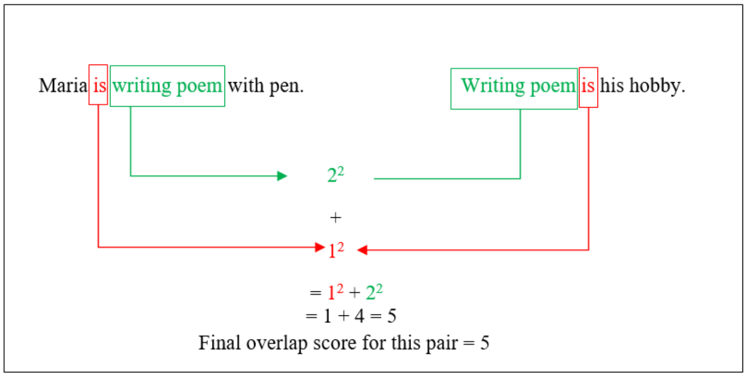
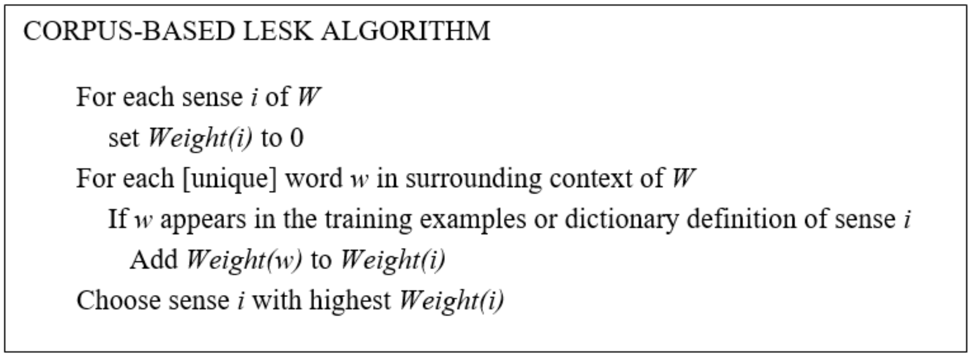

Maria is writing poem with pen.
Writing poem is his hobby.
2^2
+
1^2
$= 1^2 + 2^2$
$= 1 + 4 = 5$
Final overlap score for this pair = 5

Figure 3. 8: Multi-token overlap counting method of adapted Lesk algorithm

This method of overlap counting gives the higher value for the longer token phrase and preserve the semantic relatedness of words that occur together in a context. For the two sentences 1) "Hunter saw birds sitting in a tree." and 2) "There are many birds sitting in a tree.", the original Lesk algorithm counts 5 overlaps and the adapted Lesk algorithm counts 25 (square of 5) which is significantly higher. Thus, this scoring method magnifies the score for the longer token phrase.

Corpus-based Lesk Algorithm



```
CORPUS-BASED LESK ALGORITHM

    For each sense $i$ of $W$
        set $Weight(i)$ to 0
    For each [unique] word $w$ in surrounding context of $W$
        If $w$ appears in the training examples or dictionary definition of sense $i$
            Add $Weight(w)$ to $Weight(i)$
    Choose sense $i$ with highest $Weight(i)$
```

Figure 3. 9: Corpus-based Lesk algorithm

The corpus-based Lesk algorithm is a supervised WSD method and one of the best performing algorithm [76]. The corpus-based algorithm uses annotated training examples and dictionary definitions. Word’s weight is calculated using the method of inverse document frequency (IDF) method. The sense having the highest weight is chosen as correct meaning. This algorithm is shown in Figure 3.9.

3.1.2 Methods Based on Similarity Measures

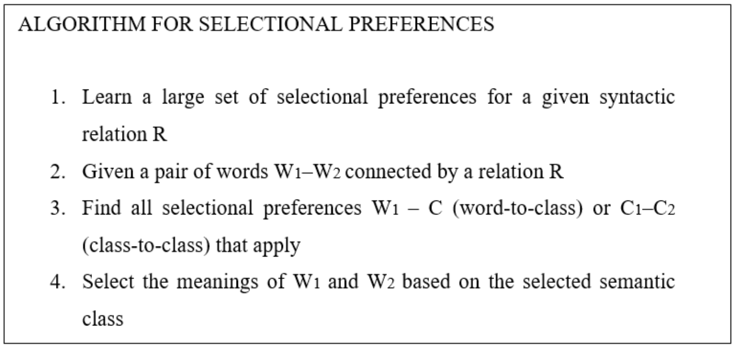
This subsection describes the knowledge-based WSD methods which use semantic similarity over the words in a given context. For these approaches, the main important task is to determine a metric to measure semantic similarity among context words using semantic networks like WordNet. It is measured by calculating semantic density and/or distance between two concepts on such semantic networks. The similarity measure can be based on the local or global context.

The local context for similarity measures refers to the similarity between pair of words or surrounding words within the context window size while the global context refers to similarity measured based on the entire text beyond the small window centered on target word. To which extent words are semantically related to each other defines the semantic similarity. Some of the similarity measures includes the approaches proposed by Resnik [77], Leacock [78] and Hirst [79].

Resnik defines the semantic similarity using information content (IC) which is a specificity measure of a concept, C [77]. The IC is defined as probability of occurrence of C. In a huge amount of corpus, IC(C) = -log (P(C)). The lowest common subsumer (LCS) of two concepts-C1 and C2, the semantic similarity between words is defined as Similarity (C1, C2) = IC(LCS(C1, C2)). Leacock and Chodorow uses the minimum path value between concepts to determine semantic similarity [78]. They defined it as Similarity (C1,C2) = - log [(path(C1,C2) /2D] where path(C1, C2) is the path connecting C1 and C2 and D is the depth of taxonomy. There are many other methods for similarity measure.

3.1.3 Selectional Preferences Methods

The selectional preference methods of WSD approaches use the commonsense rules for finding the meaning of a word in a context for disambiguation [80]. These methods that rely on selectional preference or constrain, represents commonsense knowledge about the concepts by capturing the information of possible relations between the word classes or concept classes. Then, these methods use the semantic constrains to select only the senses which agree with commonsense rules. Thus, these approaches of WSD first collect the commonsense knowledge about the word classes, then uses the semantic constrains to disprove the occurrence of classes of incorrect meanings that disagree the commonsense rules and finally chooses the correct word meanings that agree with the commonsense rules. For example, let us consider a context "Ram ate orange." In this case, a semantic constrain EAT-FOOD can be used as commonsense rule to disagree the "color" meaning of "orange” since verb "ate" needs a food that can be eaten.



ALGORITHM FOR SELECTIONAL PREFERENCES

1. Learn a large set of selectional preferences for a given syntactic relation R
2. Given a pair of words W_1–W_2 connected by a relation R
3. Find all selectional preferences W_1 – C (word-to-class) or C_1–C_2 (class-to-class) that apply
4. Select the meanings of W_1 and W_2 based on the selected semantic class

Figure 3. 10: Algorithm for Selectional Preferences

These methods can use annotated corpora or raw corpora to obtain the selectional preferences for disambiguation. To obtain the selectional preference or constrains from raw corpora, there are methods such as information theory measures, frequency counts and class to class relation measures.

3.1.4 Heuristic Based Methods

The heuristics based WSD methods uses the linguistic properties of words or concepts that are found on the large texts. Such property or fact is referred to as a heuristic. The most-frequent-sense heuristic is one example of such heuristic. The meanings of a polysemy word have different occurrence frequencies of [81]. Figure 3.11 shows the Zipfian distribution of word meanings. Based on the heuristic- most-frequent-sense, the WSD method assigns most frequent sense to each polysemy word. For example, since the meaning "plant/flora" is found more often as compared with "plant/factory", therefore the WSD method using the heuristic "most-frequent-sense", annotates any instance of PLANT as "plant/flora".

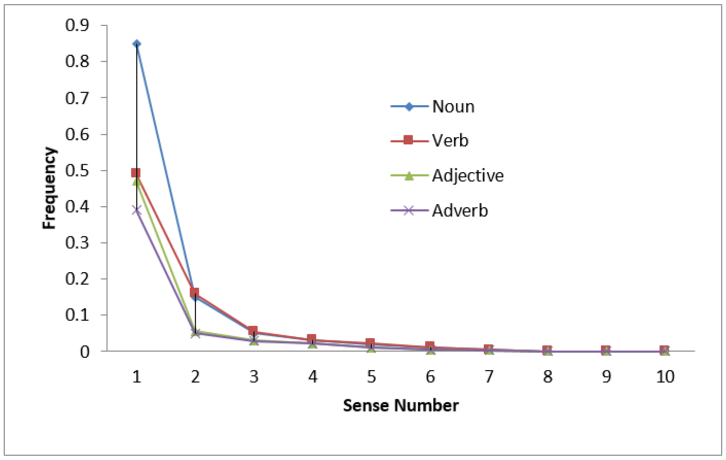


Figure 3. 11: Zipfian distribution of word meanings in SemCor

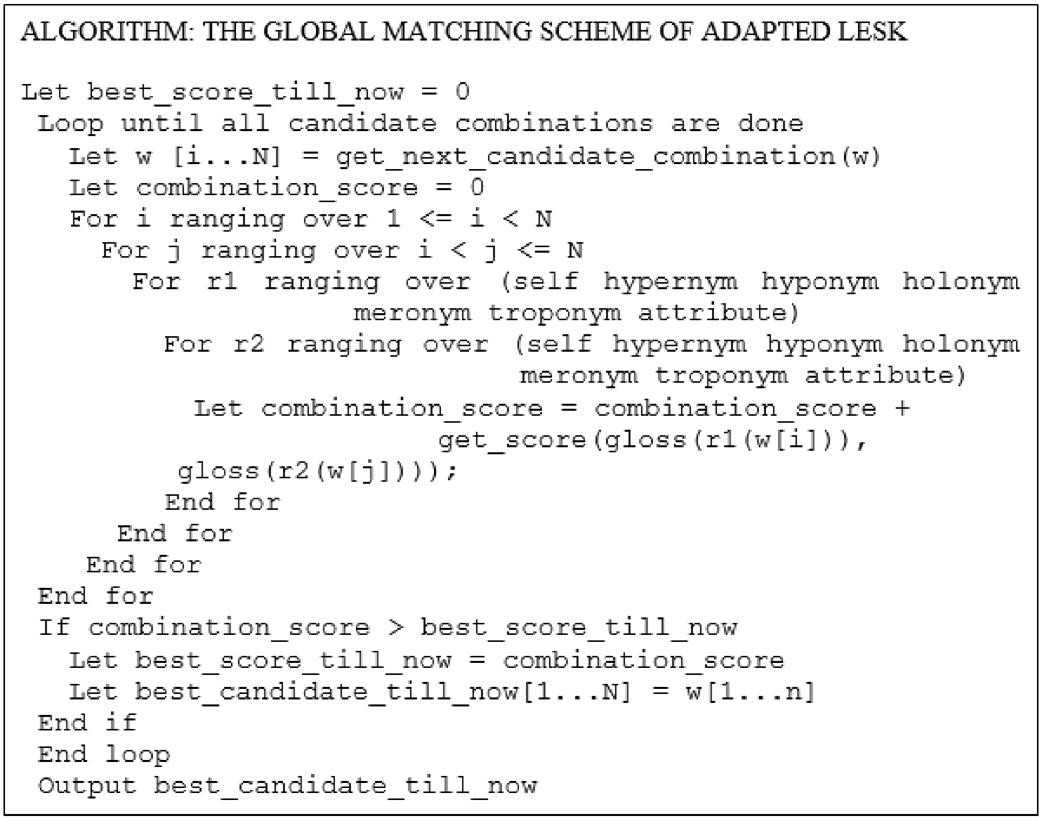
There are many other heuristics that are proposed for the sense disambiguation. However, this research limits the discussion of these heuristic method with this heuristic "most-frequent-sense".

3.2 ALGORITHMS USING WORDNET RELATIONS

This section describes various overlap count WSD algorithms that utilize information from WordNet. The purpose to discuss on these algorithm is to find out how and which relations from the WordNet are being used for sense disambiguation.

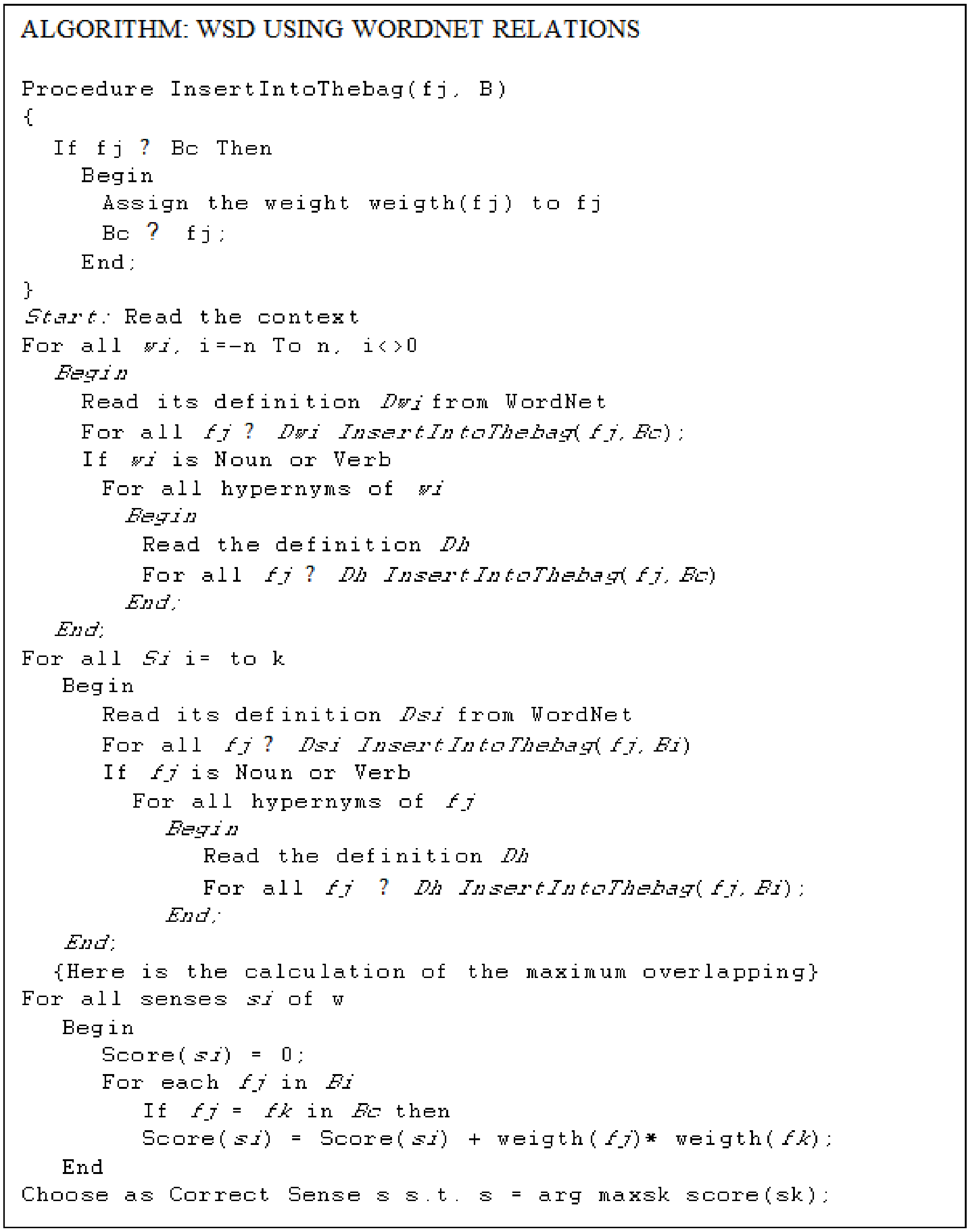
3.2.1 Adapted Lesk Algorithm

In 2002, Banerjee and Pedersen adjusted the original Lesk algorithm by taking advantage of more information provided by the lexical database WordNet [13]. They modified the Lesk algorithm in various ways. They used the shorter context window of three words due to the computational reasons as in the simulated annealing version of Lesk algorithm. Their context window is centered on the target word. They used WordNet instead of using the Oxford Advanced Learner's Dictionary. Another important modification is that they used the information from words' synsets and also from those synsets which are connected to these words from the relations- "Hypernyms", "Hyponym", "Troponym", "Holonym", "Meronym", "attribute" relation, "Also see" relation, "Pertainym" relation and "Similar to" relation etc for nouns, verbs and adjectives as shown in Table 3.1.



```
ALGORITHM: THE GLOBAL MATCHING SCHEME OF ADAPTED LESK

Let best_score_till_now = 0
 Loop until all candidate combinations are done
   Let w [i...N] = get_next_candidate_combination(w)
   Let combination_score = 0
   For i ranging over 1 <= i < N
     For j ranging over i < j <= N
       For r1 ranging over (self hypernym hyponym holonym
                    meronym troponym attribute)
         For r2 ranging over (self hypernym hyponym holonym
                           meronym troponym attribute)
           Let combination_score = combination_score +
                        get_score(gloss(r1(w[i])),
          gloss(r2(w[j])));
         End for
      End for
    End for
 End for
 If combination_score > best_score_till_now
   Let best_score_till_now = combination_score
   Let best_candidate_till_now[1...N] = w[1...n]
 End if
 End loop
 Output best_candidate_till_now
```

Figure 3. 12: The global matching scheme of adapted Lesk



ALGORITHM: WSD USING WORDNET RELATIONS

```
Procedure InsertIntoThebag(fj, B)
{
  If fj ? Bc Then
    Begin
      Assign the weight weigth(fj) to fj
      Bc ? fj;
    End;
}
Start: Read the context
For all wi, i=-n To n, i<>0
  Begin
    Read its definition Dwi from WordNet
    For all fj ? Dwi InsertIntoThebag(fj, Bc);
    If wi is Noun or Verb
      For all hypernyms of wi
        Begin
          Read the definition Dh
          For all fj ? Dh InsertIntoThebag(fj, Bc)
        End;
  End;
For all Si i= to k
   Begin
      Read its definition Dsi from WordNet
      For all fj ? Dsi InsertIntoThebag(fj, Bi)
      If fj is Noun or Verb
         For all hypernyms of fj
           Begin
              Read the definition Dh
              For all fj  ?  Dh InsertIntoThebag(fj, Bi);
           End;
   End;
  {Here is the calculation of the maximum overlapping}
For all senses si of w
   Begin
      Score(si) = 0;
      For each fj in Bi
         If fj = fk in Bc then
         Score(si) = Score(si) + weigth(fj)* weigth(fk);
   End
Choose as Correct Sense s s.t. s = arg maxsk score(sk);
```

Figure 3. 13: Algorithm proposed by Fragos et al. (2003)

The results of the experiments on their system when tested by Senseval-2 showed an overall accuracy of 32%. This accuracy was 2 times of that original Lesk algorithm. The Adapted Lesk Algorithm is generalized by Patwardhan using "is-a" sematic relationship that exist in the WordNet as a semantic relatedness measure [66]. In 2003, Banerjee and Pedersen utilized extended gloss overlap method as a semantic measure method [68]. They used this extended measure method in order to enlarge the overlaps among glosses of various relations- hypernyms, hyponym, meronym, holonym and troponym, synsets but not only between the glosses of the synsets as it was before.

3.2.2 WSD Using WordNet Relations

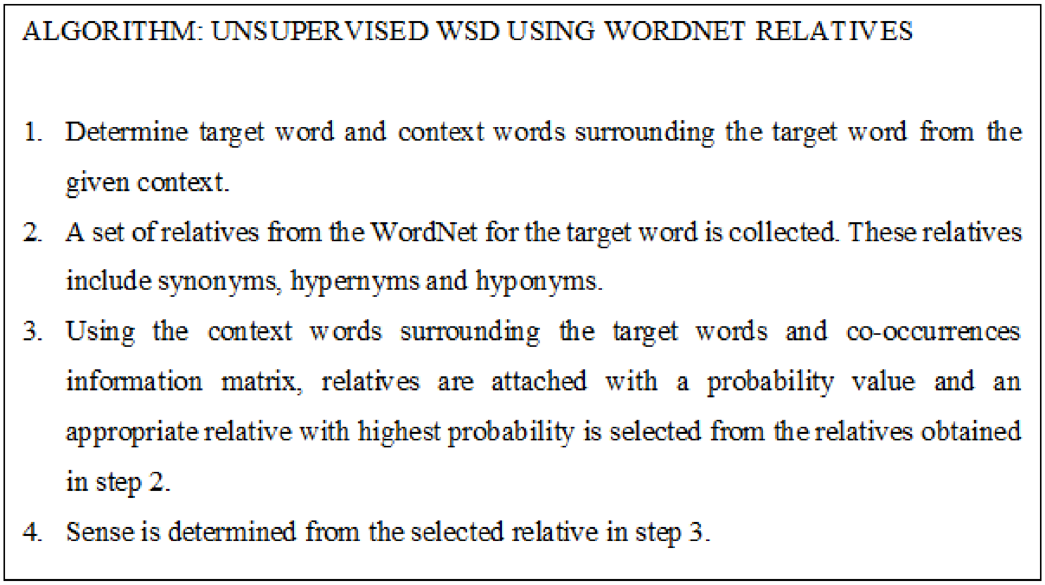
Fragos proposed the "Weighted Overlapping" disambiguation method [14]. They used the definitions of sense of word, synset definitions and hypernymy relations to form both sense bag and context bag for disambiguation. They used the definitions from all noun’s and verb’s hypernyms. In addition, they assigned to hypernym synset level a weight. The assigned weight was inversely proportional to the WordNet’s depth of the hierarchy. Their algorithm is shown is Figure 3.13.

They have concluded that the use of hypernymy improves the result of Lesk algorithm. From the results obtained from their experiments, they found there is no contribution of the hyponymy in disambiguation as the inclusion of information from hyponymy found to decrease accuracy by 6%. They evaluated their algorithm using Brown corpus data and observed 52.5% accuracy. With the use of heuristic, they attained the accuracy of 66.2%.

3.2.3 Unsupervised WSD Using WordNet Relatives

Seo et al. developed an unsupervised WSD method. They utilized a set of relations- synonyms, hyponyms and hypernyms from WordNet [15]. They have also used the co-occurrences matrix to assign probability value to a relative. Finally, they chose the sense with highest probability as a correct sense.

From the results they obtained they observed various important facts: - 1) higher the hierarchy level lesser semantic bond and therefore higher level hypernyms/hyponyms are not appropriate for sense disambiguation and 2) synonym relatives share similar context. They also concluded that the WordNet still don’t hold sufficient info that is needed for sense disambiguation.



ALGORITHM: UNSUPERVISED WSD USING WORDNET RELATIVES

1. Determine target word and context words surrounding the target word from the given context.
2. A set of relatives from the WordNet for the target word is collected. These relatives include synonyms, hypernyms and hyponyms.
3. Using the context words surrounding the target words and co-occurrences information matrix, relatives are attached with a probability value and an appropriate relative with highest probability is selected from the relatives obtained in step 2.
4. Sense is determined from the selected relative in step 3.

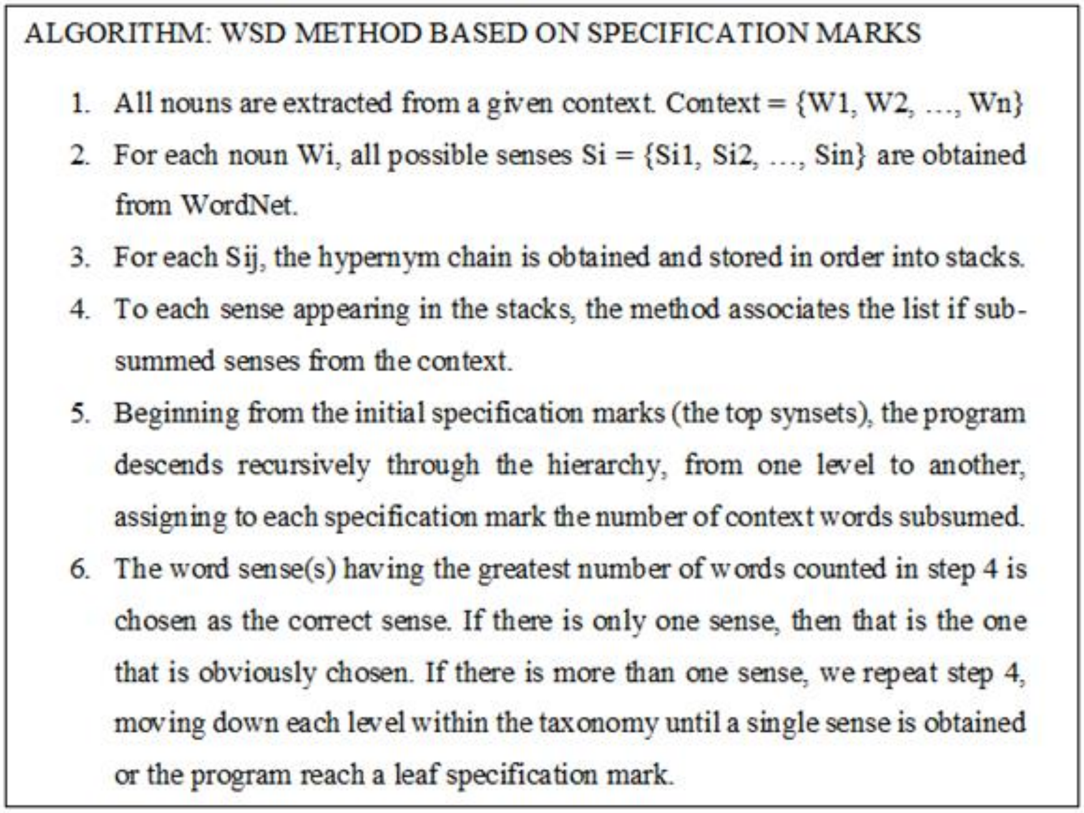
Figure 3. 14: Unsupervised WSD using WordNet relatives proposed by Seo et al. (2004)

3.2.4 WSD Method Based on Specification Marks

Montoyo et al. proposed a knowledge based WSD algorithm. They utilized the fact that two words shares a large amount of information if the two words have high similarity [16]. A group of noun is formed from the context.

Word’s definition (gloss) and hypernymy/hyponymy are collected for each word in context. This collection is referred to as initial specification mark (ISM). This ISM is used to find the sense of polysemy word. If it is not sufficient for disambiguation, then new specification marks are determined by descending through the hierarchy level. The sense with highest number of words in specification mark is returned as correct sense.

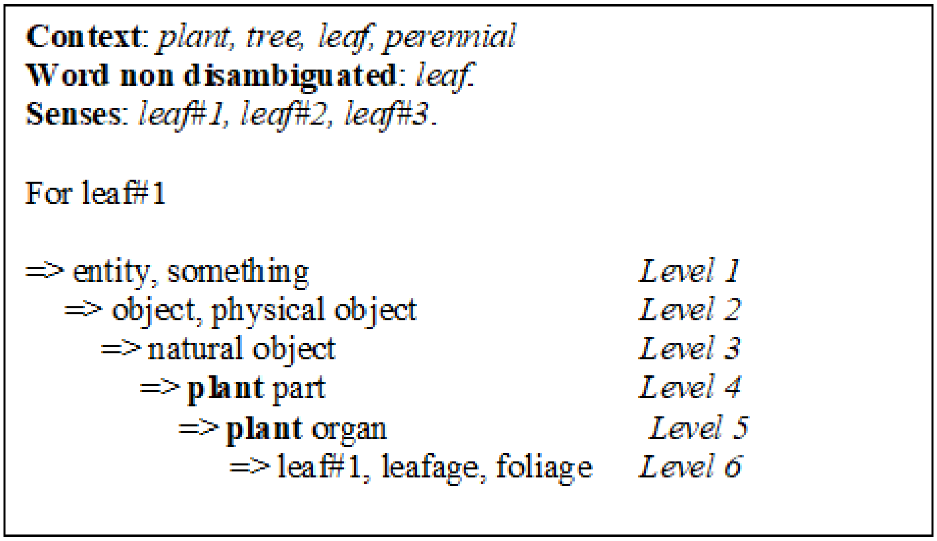
They have obtained the specification mark by using the five heuristics which are knowledge-based. These heuristics include definition, hypernym/hyponym, gloss hypernym/hyponym, domain and common specification mark heuristics. For example, the Figure 3.16 shows the hypernym heuristic for context {plant, tree, leaf, perennial} where the target word is {leaf} with 3 senses.



ALGORITHM: WSD METHOD BASED ON SPECIFICATION MARKS

1. All nouns are extracted from a given context. Context = {W1, W2, …, Wn}
2. For each noun Wi, all possible senses Si = {Si1, Si2, …, Sin} are obtained from WordNet.
3. For each Sij, the hypernym chain is obtained and stored in order into stacks.
4. To each sense appearing in the stacks, the method associates the list if subsummed senses from the context.
5. Beginning from the initial specification marks (the top synsets), the program descends recursively through the hierarchy, from one level to another, assigning to each specification mark the number of context words subsumed.
6. The word sense(s) having the greatest number of words counted in step 4 is chosen as the correct sense. If there is only one sense, then that is the one that is obviously chosen. If there is more than one sense, we repeat step 4, moving down each level within the taxonomy until a single sense is obtained or the program reach a leaf specification mark.

Figure 3. 15: WSD method based on specification marks proposed by Montoyo et al. (2005)



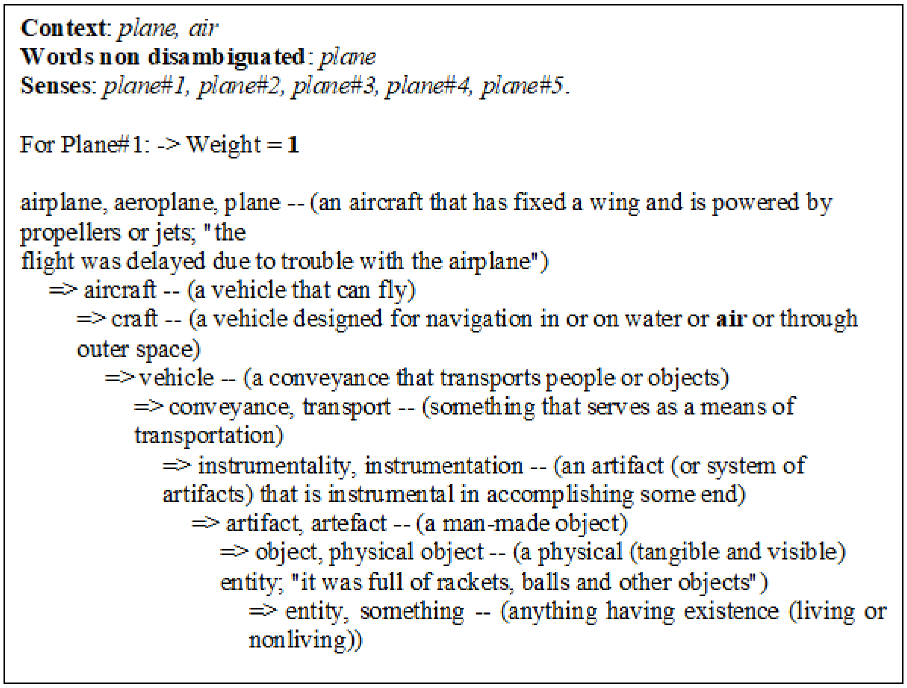
Context: *plant, tree, leaf, perennial*
Word non disambiguated: *leaf.*
Senses: *leaf#1, leaf#2, leaf#3*.

For leaf#1

=> entity, something *Level 1*
=> object, physical object *Level 2*
=> natural object *Level 3*
=> **plant** part *Level 4*
=> **plant** organ *Level 5*
=> leaf#1, leafage, foliage *Level 6*

Figure 3. 16: Hypernym Heuristic defined by Montoyo et al. (2005)

This heuristic show the hypernymy heuristic for sense leaf#1. In the Figure 3.16, it is seen that there is two overlaps in the hypernymy level 4 and 5 with the context word "plant". Similarly, let us see on the Gloss Hypernym Heuristic as shown in Figure 3.17 used in their method for the context {plane, air} where the target word is {plane}. There is only one overlap count which is the highest overlap. The plane#1 is returned as the correct meaning.



Context: *plane, air*
Words non disambiguated: *plane*
Senses: *plane#1, plane#2, plane#3, plane#4, plane#5*.

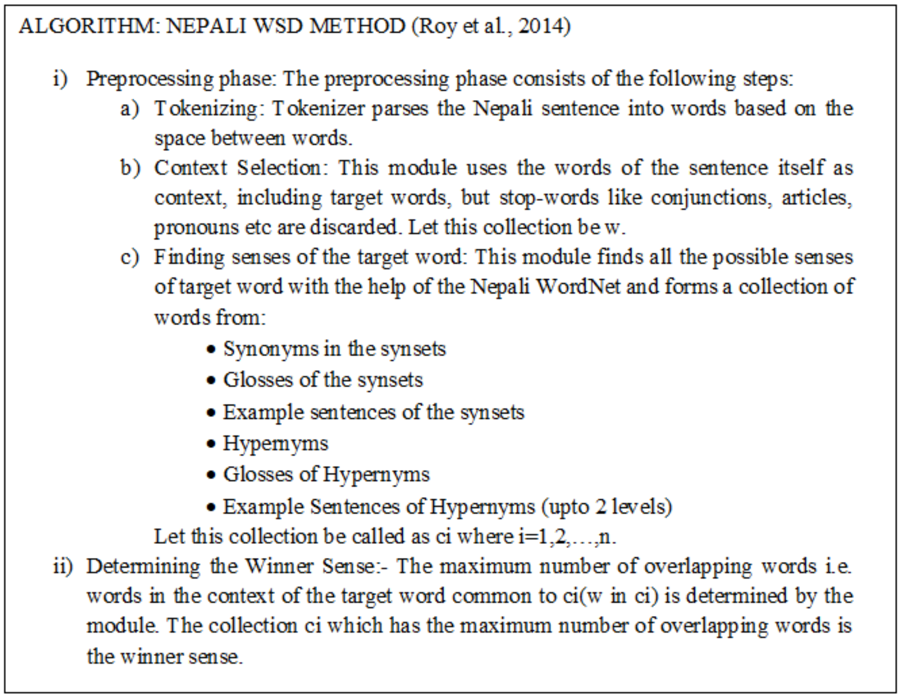
For Plane#1: -> Weight = **1**

airplane, aeroplane, plane -- (an aircraft that has fixed a wing and is powered by propellers or jets; "the
flight was delayed due to trouble with the airplane")
=> aircraft -- (a vehicle that can fly)
=> craft -- (a vehicle designed for navigation in or on water or **air** or through outer space)
=> vehicle -- (a conveyance that transports people or objects)
=> conveyance, transport -- (something that serves as a means of transportation)
=> instrumentality, instrumentation -- (an artifact (or system of artifacts) that is instrumental in accomplishing some end)
=> artifact, artefact -- (a man-made object)
=> object, physical object -- (a physical (tangible and visible) entity; "it was full of rackets, balls and other objects")
=> entity, something -- (anything having existence (living or nonliving))

Figure 3. 17: Gloss Hypernym Heuristic defined by Montoyo et al. (2005)

3.2.5 WSD for Nepali Language

In 2014, Roy et al. proposed an overlap-based WSD algorithm for Nepali word sense disambiguation using Nepali WordNet built at Assam University [63]. They collected the information from synonyms, synsets' glosses and examples, hypernyms, examples and glosses of hypernyms up to level two. They have noticed that their overlap based approach is suffered from the sparse overlap.



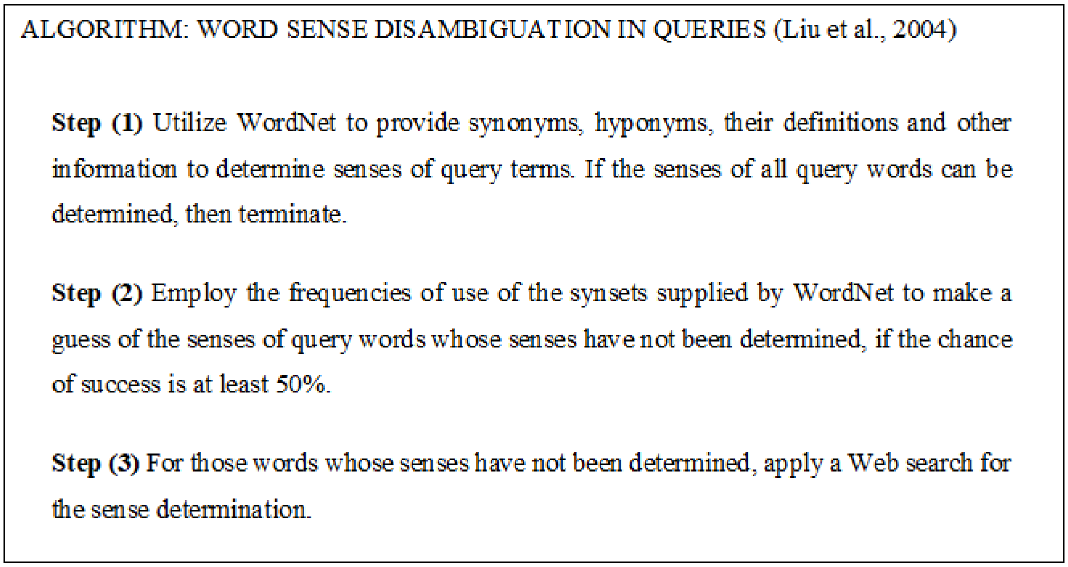
ALGORITHM: NEPALI WSD METHOD (Roy et al., 2014)

i) Preprocessing phase: The preprocessing phase consists of the following steps:
   a) Tokenizing: Tokenizer parses the Nepali sentence into words based on the space between words.
   b) Context Selection: This module uses the words of the sentence itself as context, including target words, but stop-words like conjunctions, articles, pronouns etc are discarded. Let this collection be w.
   c) Finding senses of the target word: This module finds all the possible senses of target word with the help of the Nepali WordNet and forms a collection of words from:
      - Synonyms in the synsets
      - Glosses of the synsets
      - Example sentences of the synsets
      - Hypernyms
      - Glosses of Hypernyms
      - Example Sentences of Hypernyms (upto 2 levels)

      Let this collection be called as ci where i=1,2,…,n.

ii) Determining the Winner Sense:- The maximum number of overlapping words i.e. words in the context of the target word common to ci(w in ci) is determined by the module. The collection ci which has the maximum number of overlapping words is the winner sense.

Figure 3. 18: Nepali WSD method proposed by Roy et al. (2014)



ALGORITHM: WORD SENSE DISAMBIGUATION IN QUERIES (Liu et al., 2004)

Step (1) Utilize WordNet to provide synonyms, hyponyms, their definitions and other information to determine senses of query terms. If the senses of all query words can be determined, then terminate.

Step (2) Employ the frequencies of use of the synsets supplied by WordNet to make a guess of the senses of query words whose senses have not been determined, if the chance of success is at least 50%.

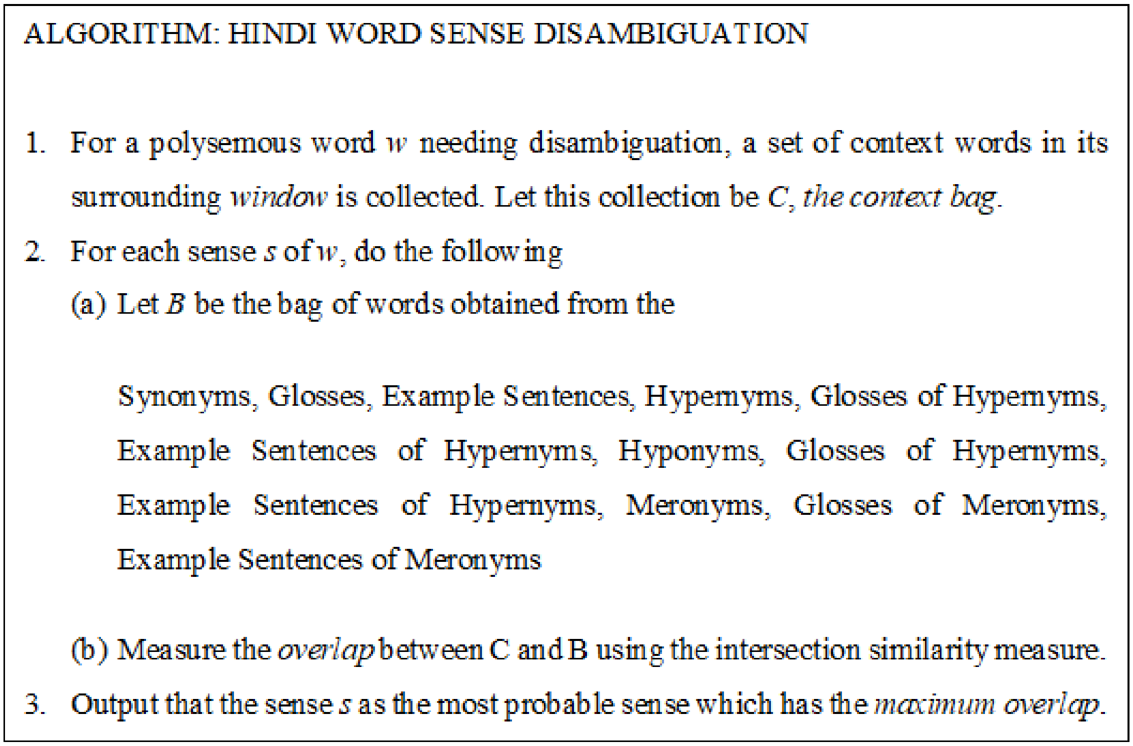
Step (3) For those words whose senses have not been determined, apply a Web search for the sense determination.

Figure 3. 19: Word Sense Disambiguation in Queries proposed by Liu et al. (2004)

3.2.6 Word Sense Disambiguation in Queries

Liu et al. proposed word sense disambiguation in queries [17]. They used WordNet to collect information associated with each word in context formed by noun phrase in a given query. The information for a word is collected from its domains, its synonyms, its synonyms' definitions, its hyponyms and its hypernyms. If the information could not disambiguate, then the meaning is guessed by using highest frequent sense and even if this fails, then they take assist from a web search.

3.2.7 Word Sense Disambiguation in Hindi Language



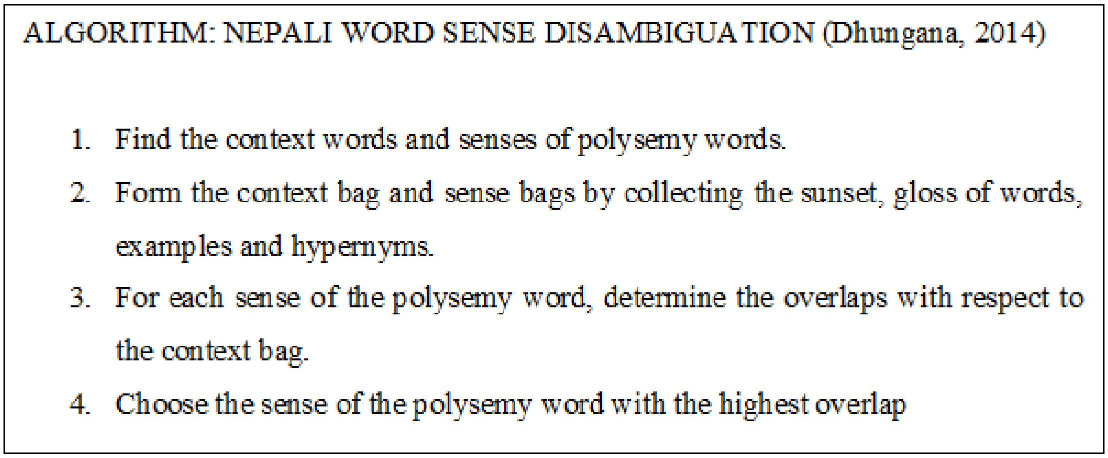
```
ALGORITHM: HINDI WORD SENSE DISAMBIGUATION

1. For a polysemous word w needing disambiguation, a set of context words in its
   surrounding window is collected. Let this collection be C, the context bag.
2. For each sense s of w, do the following
   (a) Let B be the bag of words obtained from the

       Synonyms, Glosses, Example Sentences, Hypernyms, Glosses of Hypernyms,
       Example Sentences of Hypernyms, Hyponyms, Glosses of Hypernyms,
       Example Sentences of Hypernyms, Meronyms, Glosses of Meronyms,
       Example Sentences of Meronyms

   (b) Measure the overlap between C and B using the intersection similarity measure.
3. Output that the sense s as the most probable sense which has the maximum overlap.
```

Figure 3. 20: Hindi Word Sense Disambiguation proposed by Sinha et al. (2003)

Sinha et al. proposed a WSD method for Hindi word sense disambiguation [69]. The propose method was for nouns only and it utilized statistical technique for finding the sense. The words in context bag are compared with words sense bags formed. The bags are formed by collecting the information from many relations in Hindi WordNet. These includes 1) word's synonyms, glosses and examples, 2) Meronyms and their glosses and examples, 3) Hypernyms and their glosses and examples and 4) Hyponyms and their glosses and examples. A simple overlap count technique is used in their system.

3.2.8 Nepali Word Sense Disambiguation

Dhungana and Shakya proposed a WSD method for Nepali language with some modification on adapted Lesk algorithm [18]. They used the synset, glosses of words, examples and hypernyms to build the sense bags and the context bags. In their experiment, they found that when the number of examples for the context and sense bags are increased, accuracy increases. However, when lower level hypernyms are considered in disambiguation, the accuracy is found to be decreased. From this, they concluded 1) the only first level hypernymy has less information and 2) if all hypernyms are considered, it induces common information and is not appropriate to use for disambiguation. Rather, the common information cause noise information and this results in wrong disambiguation by increasing the number of overlaps of context with wrong sense.



ALGORITHM: NEPALI WORD SENSE DISAMBIGUATION (Dhungana, 2014)

1. Find the context words and senses of polysemy words.
2. Form the context bag and sense bags by collecting the sunset, gloss of words, examples and hypernyms.
3. For each sense of the polysemy word, determine the overlaps with respect to the context bag.
4. Choose the sense of the polysemy word with the highest overlap

Figure 3. 21: Nepali word sense disambiguation proposed by Dhungana and Shakya (2014)

3.3 POINTS NOTED ON ALGORITHMS USING WORDNET RELATIONS

In section 3.2, eight knowledge based WSD methods which use overlap count method for sense disambiguation are discussed in detail. The detail investigation on these algorithms summarizes the following important points:

1. These WSD methods are found to use information from relations of WordNet. The relations used in those methods are “synsets”, “glosses”, “examples”, “hypernymy”, “holonymy”, “hyponymy”, “troponymy”, “meronymy”, “attribute” relation, “also see” relation, “similar to”, “pertainym” and “domain” relations for nouns, verbs and adjectives [10] [13] [14] [15] [16] [17] [18] [36] [63] [69] [72].
2. The info from hyponymy relation of WordNet are not appropriate. Rather, its inclusion causes to decrease the accuracy [14].
3. The higher level (here higher level means deeper one) hypernyms in the WordNet are common to all senses of a polysemy word and therefore are not appropriate. The top level distinct hypernyms are few. Therefore, WordNet still doesn’t contain information which are sufficient for sense disambiguation [15].
4. Even if full hypernymy hierarchy is considered, it contains only few distinct words that match with context. Only one word is overlapped in 130 words from hypernymy hierarchy for a sense of target word as shown in fig 3.17 [16]. Therefore, WordNet is still lacking necessary information which are sufficient for finding meaning of polysemy words. [17] [18].
5. If full hypernymy hierarchy is considered, Sense and context bags will have huge amount of information which are common to all senses. Such common information are not appropriate for distinguishing the senses. Rather, it introduces noise information. This noise information increases the number of overlaps of context with wrong sense. Therefore, it causes wrong disambiguation. [18].

These facts clearly indicate the current organization of words in WordNet is inadequate to address the problem of word sense disambiguation of polysemy words. This is because the current organization of words in WordNet induces the noise information for overlap count knowledge based WSD systems due to the common information for two or more than two senses of the same polysemy word. In addition, WordNet organizes the words based on synonyms and connected the words using many relations. However, it does not deal with the relationship of words with polysemy words.

3.4 ILLUSTRATION OF USE OF HYPERNYM RELATION

In this section, the use of hypernym relation is illustrated in detail for word sense disambiguation with examples. For simplicity, only the hypernym relation is chosen for illustration since it is the one of the most used relation from WordNet for sense disambiguation. In this illustration, it is explained how the hypernym can used to contribute in word sense disambiguation, how the inclusion of deeper level hypernym induce the common information and it results in wrong disambiguation.

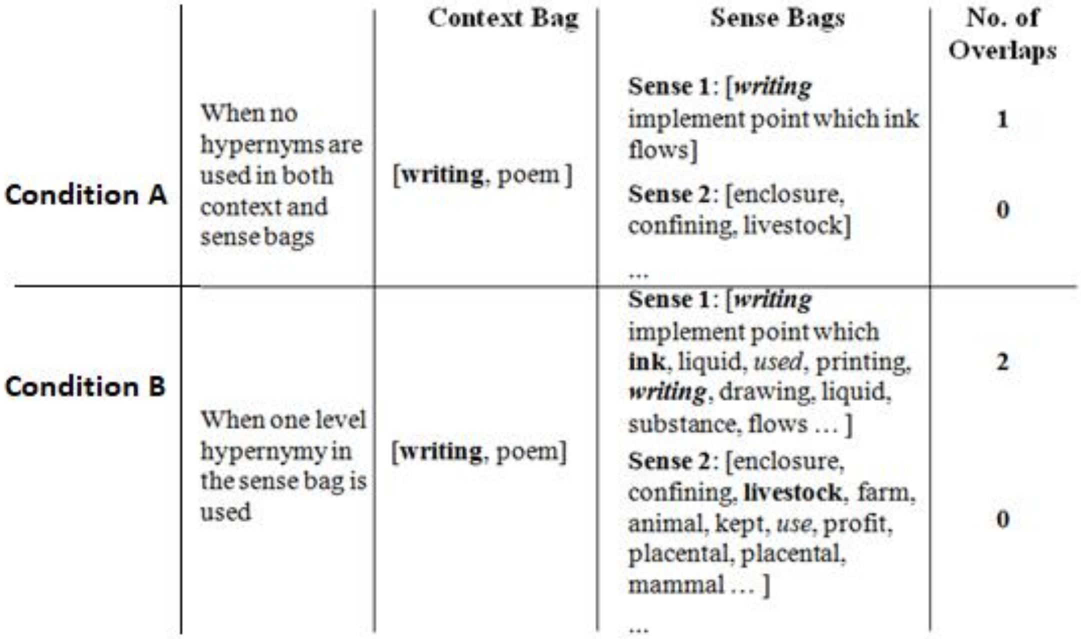
3.4.1 The Hyponymy/Hypernymy Relation

The both synonym and the antonym are lexical relations between two words. In contrast, hyponymy and hypernymy are semantic relations and express a subordinate/superordinate relation or "is a" relation. A set of words say X is hyponym of another set of words say Y if a word $x_1 \in X$ is a kind of $y_1 \in Y$. Then Y is hypernym of X. For instance {cauliflower} is hyponym of {vegetables}. The {cauliflower} receives all characteristics from {vegetables} plus has its own characteristics that differentiates from its super ordinate. Hyponymy is an asymmetrical but transitive relation [82]. The hypernym relation is a frequently used relation of WordNet for sense disambiguation.

Suppose a context say C and a polysemy word say P in the C. Suppose P has *n* different senses. Suppose a condition A in which sense bags are formed by considering only sense's glosses. Suppose another condition B in which sense bag contains info from both sense's glosses plus sense's level 1 hypernym.

It is then supposed that the inclusion of hypernym increases the number of overlaps between context and correct sense of a polysemy words. Therefore, it suggest that the number of overlaps between C and the correct sense of P is greater in condition B as compared to condition A. Mathematically,

$$Overlaps(C\ and\ correctSense(P)\ in\ Condition\ B\ ) \\ > overlaps(C\ and\ correct(P)\ in\ Condition\ A)$$



		Context Bag	Sense Bags	No. of Overlaps
Condition A	When no hypernyms are used in both context and sense bags	[**writing**, poem]	**Sense 1**: [***writing*** implement point which ink flows]	1
			Sense 2: [enclosure, confining, livestock] ...	0
Condition B	When one level hypernymy in the sense bag is used	[**writing**, poem]	**Sense 1**: [***writing*** implement point which **ink**, liquid, *used*, printing, ***writing***, drawing, liquid, substance, flows …]	2
			Sense 2: [enclosure, confining, **livestock**, farm, animal, kept, *use*, profit, placental, placental, mammal …] ...	0

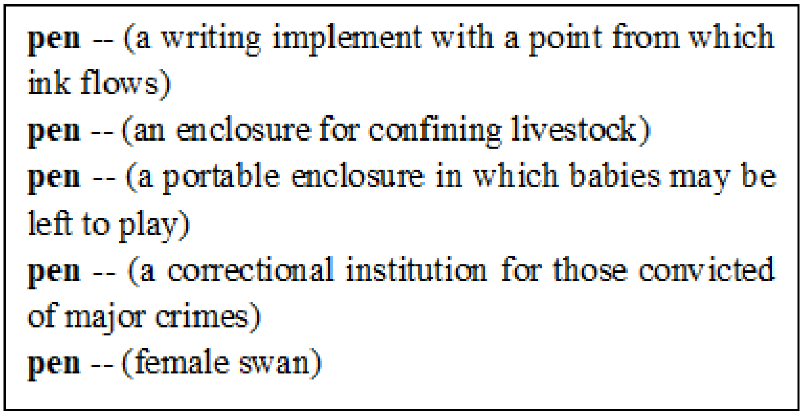
Figure 3. 22: Inclusion of hypernym increases overlaps of context with correct sense

Suppose a context *".... writing a poem with a **pen**",* target word- ***Pen***. Therefore, Context bag, C = {*writing, poem*}. The Figure 3.22 shows two conditions A and B. The condition A has sense bag with only sense's gloss while condition B has sense bag with sense's gloss plus its first level hypernym. The number of overlaps of context with correct sense is 1 in Condition A while it is double (i.e. 2) in condition B. This indicates that the inclusion of hypernym in sense bag increase the number of overlaps for correct sense. This is the reason for massively using info from hypernym relation of WordNet for sense disambiguation. But, it is not true always. The inclusion of info from hypernymy is also found to increase the overlaps for incorrect sense of polysemy words. It is because of induction of noise information from common hypernyms. These cases are illustrated in the following subsections.

3.4.2 WSD Approaches and Use of Hypernyms

The noun ***pen*** has 5 senses in WordNet 2.1 and are shown in **Figure 3.23**. Suppose the same context *".... writing a poem with a **pen**",* target word- ***Pen***. Therefore, Context bag, C = {*writing, poem*}. Now, let us examine the following three cases:

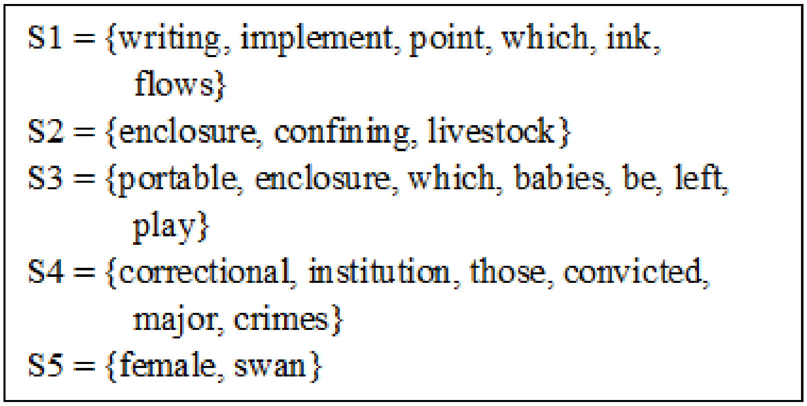
Case I: The sense bag is formed by including only. It is a set of words from sense's gloss after removing articles, prepositions and pronouns. Figure 3.24 shows five sense bags named as S1, S2, S3, S4 and S5. These sense bags are formed from the sense's gloss of five sense in Figure 3.23 respectively.



pen -- (a writing implement with a point from which ink flows)
pen -- (an enclosure for confining livestock)
pen -- (a portable enclosure in which babies may be left to play)
pen -- (a correctional institution for those convicted of major crimes)
pen -- (female swan)

Figure 3. 23: Glosses of five senses of word "Pen" as a noun

The number of overlaps between Context bag, C = {*writing, poem*} with each sense bag are calculated and shown in Figure 3.25. It shows maximum number of overlap 1 with sense bag S_1. The number of overlaps with all other sense bags is 0. Therefore, sense 1 is returned as a correct sense.



S1 = {writing, implement, point, which, ink, flows}
S2 = {enclosure, confining, livestock}
S3 = {portable, enclosure, which, babies, be, left, play}
S4 = {correctional, institution, those, convicted, major, crimes}
S5 = {female, swan}

Figure 3. 24: Sense bags formed from sense's gloss of *Pen*

Case II: In this case, the setting of Case I is slightly modified with update in sense bag. Sense bags now contains all hypernyms of each word in the sense's gloss. It is shown in Figure 3.26 for the first three senses of ***pen***. The number of overlaps between the context bag and sense bags after the update in sense is found to be increased by one for the correct sense and is shown in Figure 3.27. The number of overlaps for two next senses are found to be still zero.

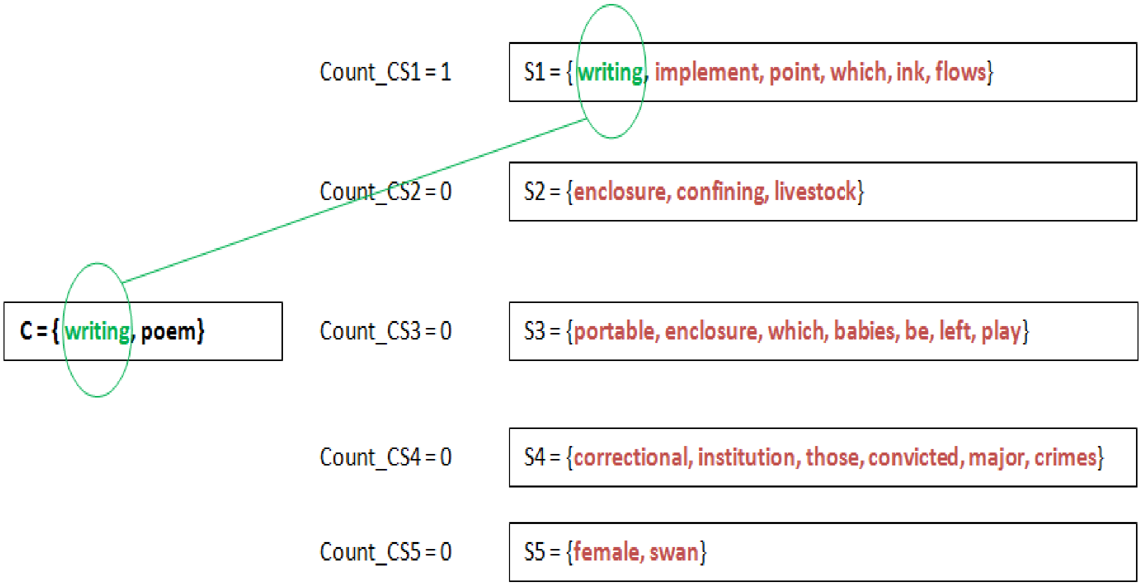
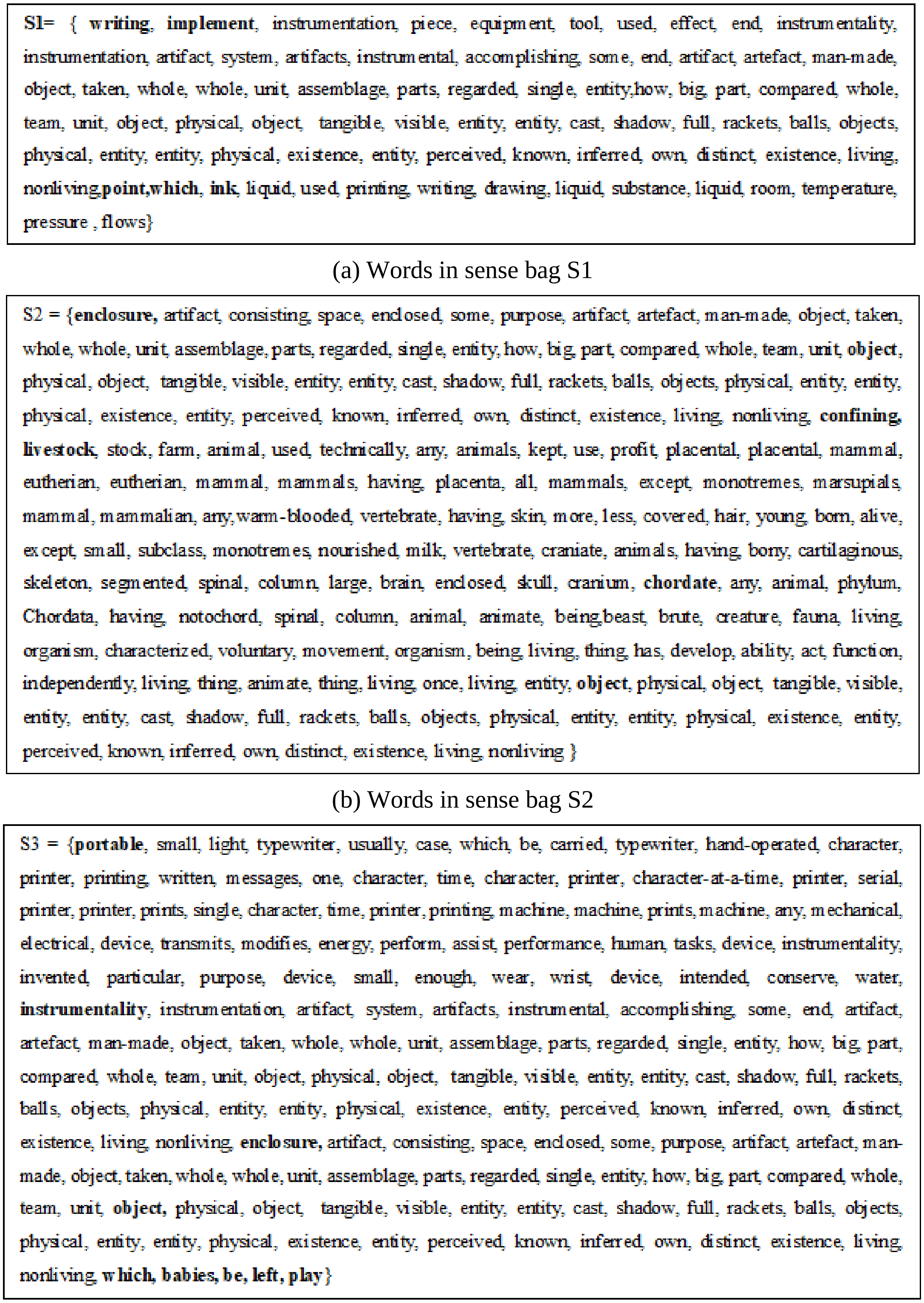

Count_CS1 = 1
S1 = {writing, implement, point, which, ink, flows}
Count_CS2 = 0
S2 = {enclosure, confining, livestock}
C = {writing, poem}
Count_CS3 = 0
S3 = {portable, enclosure, which, babies, be, left, play}
Count_CS4 = 0
S4 = {correctional, institution, those, convicted, major, crimes}
Count_CS5 = 0
S5 = {female, swan}

Figure 3. 25: No of overlaps between the context bag and each sense bag of *Pen*



S1= { **writing**, **implement**, instrumentation, piece, equipment, tool, used, effect, end, instrumentality, instrumentation, artifact, system, artifacts, instrumental, accomplishing, some, end, artifact, artefact, man-made, object, taken, whole, whole, unit, assemblage, parts, regarded, single, entity,how, big, part, compared, whole, team, unit, object, physical, object, tangible, visible, entity, entity, cast, shadow, full, rackets, balls, objects, physical, entity, entity, physical, existence, entity, perceived, known, inferred, own, distinct, existence, living, nonliving,**point,which**, **ink**, liquid, used, printing, writing, drawing, liquid, substance, liquid, room, temperature, pressure , flows}

(a) Words in sense bag S1

S2 = {**enclosure,** artifact, consisting, space, enclosed, some, purpose, artifact, artefact, man-made, object, taken, whole, whole, unit, assemblage, parts, regarded, single, entity, how, big, part, compared, whole, team, unit, **object**, physical, object, tangible, visible, entity, entity, cast, shadow, full, rackets, balls, objects, physical, entity, entity, physical, existence, entity, perceived, known, inferred, own, distinct, existence, living, nonliving, **confining, livestock**, stock, farm, animal, used, technically, any, animals, kept, use, profit, placental, placental, mammal, eutherian, eutherian, mammal, mammals, having, placenta, all, mammals, except, monotremes, marsupials, mammal, mammalian, any,warm-blooded, vertebrate, having, skin, more, less, covered, hair, young, born, alive, except, small, subclass, monotremes, nourished, milk, vertebrate, craniate, animals, having, bony, cartilaginous, skeleton, segmented, spinal, column, large, brain, enclosed, skull, cranium, **chordate**, any, animal, phylum, Chordata, having, notochord, spinal, column, animal, animate, being,beast, brute, creature, fauna, living, organism, characterized, voluntary, movement, organism, being, living, thing, has, develop, ability, act, function, independently, living, thing, animate, thing, living, once, living, entity, **object**, physical, object, tangible, visible, entity, entity, cast, shadow, full, rackets, balls, objects, physical, entity, entity, physical, existence, entity, perceived, known, inferred, own, distinct, existence, living, nonliving }

(b) Words in sense bag S2

S3 = {**portable**, small, light, typewriter, usually, case, which, be, carried, typewriter, hand-operated, character, printer, printing, written, messages, one, character, time, character, printer, character-at-a-time, printer, serial, printer, printer, prints, single, character, time, printer, printing, machine, machine, prints, machine, any, mechanical, electrical, device, transmits, modifies, energy, perform, assist, performance, human, tasks, device, instrumentality, invented, particular, purpose, device, small, enough, wear, wrist, device, intended, conserve, water, **instrumentality**, instrumentation, artifact, system, artifacts, instrumental, accomplishing, some, end, artifact, artefact, man-made, object, taken, whole, whole, unit, assemblage, parts, regarded, single, entity, how, big, part, compared, whole, team, unit, object, physical, object, tangible, visible, entity, entity, cast, shadow, full, rackets, balls, objects, physical, entity, entity, physical, existence, entity, perceived, known, inferred, own, distinct, existence, living, nonliving, **enclosure,** artifact, consisting, space, enclosed, some, purpose, artifact, artefact, man-made, object, taken, whole, whole, unit, assemblage, parts, regarded, single, entity, how, big, part, compared, whole, team, unit, **object,** physical, object, tangible, visible, entity, entity, cast, shadow, full, rackets, balls, objects, physical, entity, entity, physical, existence, entity, perceived, known, inferred, own, distinct, existence, living, nonliving, **which, babies, be, left, play**}

(c) Words in sense bag S3

Figure 3. 26: Contents of first three sense bags *pen*

Case III: In this case, the setting of Case II is slightly modified with the update in context bag. All hypernyms of each word of context bag are determined and included in context bag. For the simplicity of illustration, only the word “poem” in context bag is considered for forming context bag in this case. This is also because "writing" is polysemy word. Therefore, it is not considered to make the illustration simple.

The context bags formed in this case are shown in Figure 3.28. The number of overlaps between context bag and sense bags are determined in case III and are shown in Figure 3.29 for the first 3 senses of ***pen***. The number of overlaps for the first three senses are 44, 59 and 61 respectively. The 61 is the maximum number of overlaps of context with the third sense. Therefore, the third sense S_3 of ***pen*** is returned as correct sense but it is wrong disambiguation.

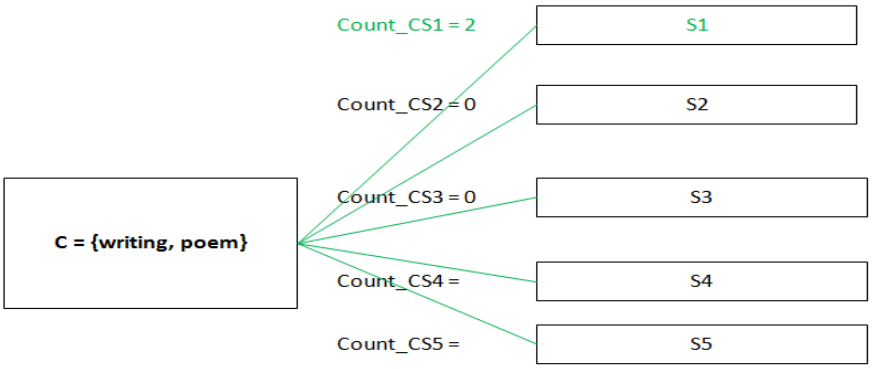


Figure 3. 27: Overlaps of words when the hypernyms are included in the sense bags

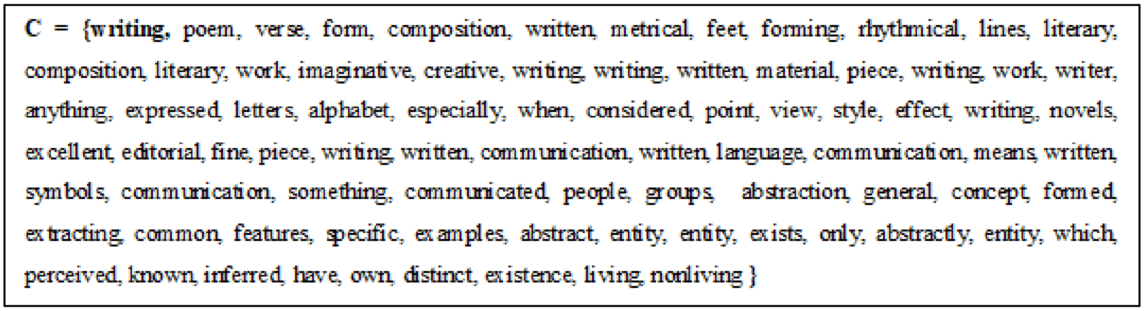

C = {**writing**, poem, verse, form, composition, written, metrical, feet, forming, rhythmical, lines, literary, composition, literary, work, imaginative, creative, writing, writing, written, material, piece, writing, work, writer, anything, expressed, letters, alphabet, especially, when, considered, point, view, style, effect, writing, novels, excellent, editorial, fine, piece, writing, written, communication, written, language, communication, means, written, symbols, communication, something, communicated, people, groups, abstraction, general, concept, formed, extracting, common, features, specific, examples, abstract, entity, entity, exists, only, abstractly, entity, which, perceived, known, inferred, have, own, distinct, existence, living, nonliving }

Figure 3. 28: Context bag when the hypernyms of "poem" are included

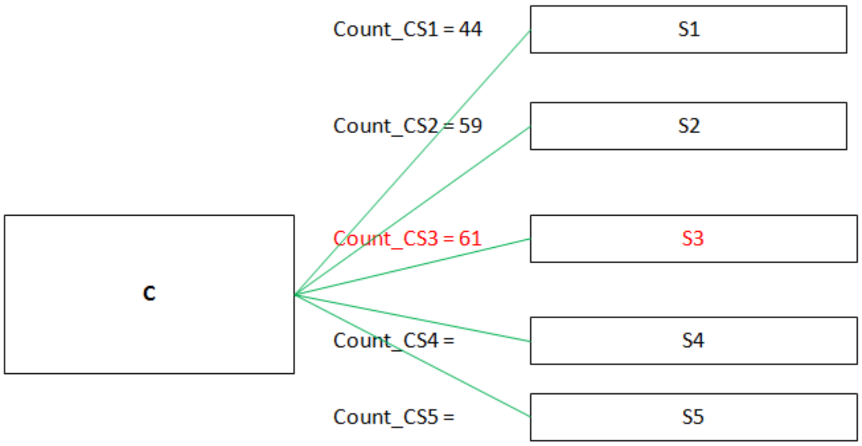


Figure 3. 29: Number of overlaps between context bag and first three sense bags of *pen* when all hypernyms are included in both bags

3.4.3 Analysis of Use of Hypernyms

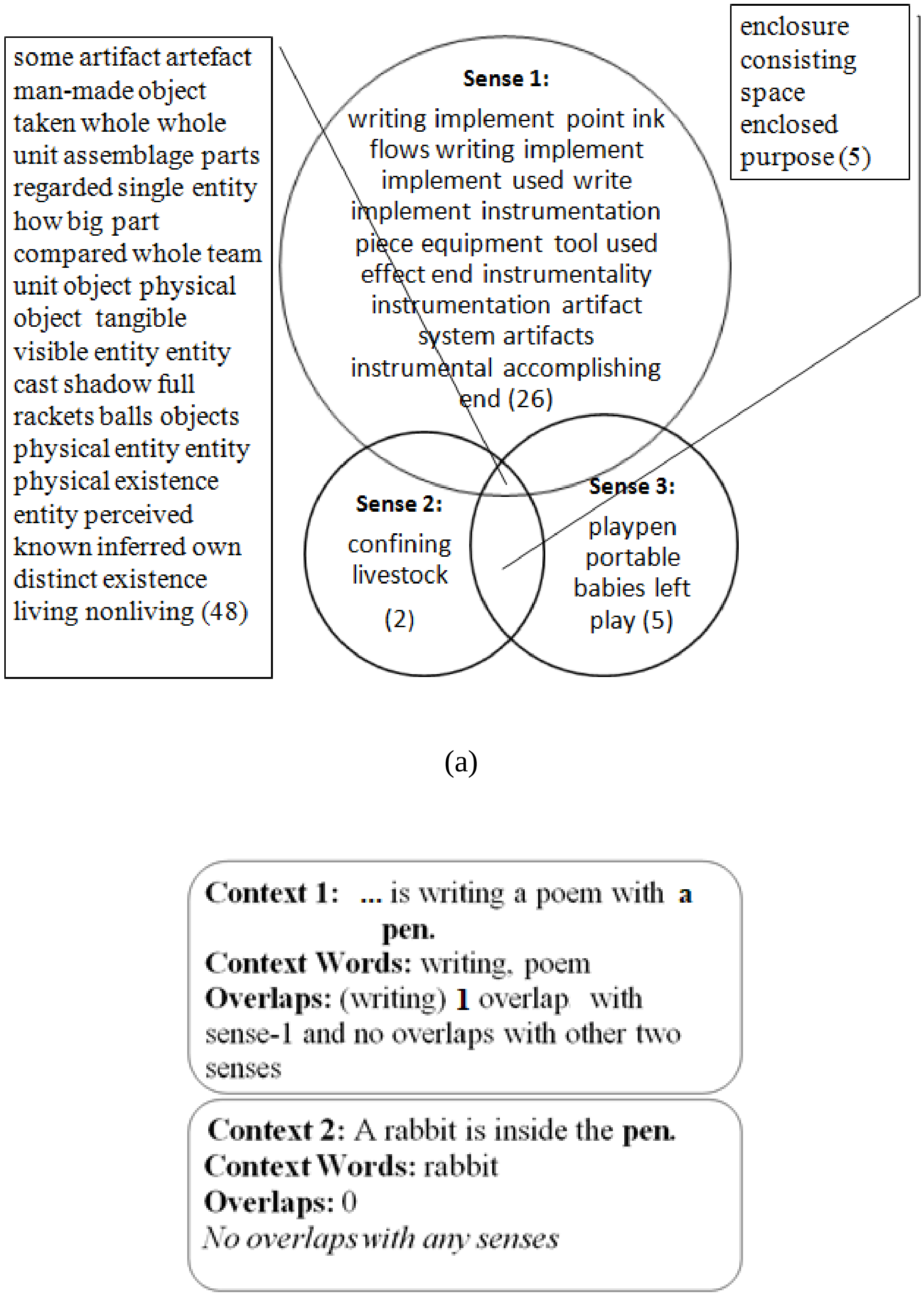
From the three cases illustrated in subsection 3.4.2, it is found that the info only from the word's gloss are less as in case I and sometimes not sufficient for sense disambiguation. When amount of info are collected from hypernyms hierarchies, more overlaps are found for correct sense as in case II. In addition, when info in context and sense bag is increased by considering hypernyms to make both bags larger as in case III, The number of overlaps for wrong sense is found to be highest. This causes the wrong disambiguation. The number of highest overlaps for wrong sense is found due to the entry of common information from hypernym hierarchies. This common information from hypernym increased the number of overlaps for wrong sense. Such information is called noise information.

This case of noise information and wrong disambiguation is also experienced in the work of Dhungana and Shakya [18] and in the work of Fragos [14]. When the deeper levels hypernyms are considered, the accuracy was found to be decreased. If hypernym hierarchies are common to the senses of a polysemy word, then there is no meaning to use such common hypernyms to distinguish the meaning of the polysemy word. It is just a waste of time to process that common info and is just a waste of memory to store that common info during processing.

Suppose the first three senses and their hypernym hierarchy of noun ***pen.*** The Figure 3.31 shows the first three senses of noun ***pen*** along with its gloss and all hypernym hierarchy. Using the set theory, Figure 3.30 (a) shows the common and distinct words among these three sense bags formed from gloss and all hypernym hierarchy of those three senses of noun ***pen*** after removing the duplicates words, prepositions, articles and pronouns. Only 26 words in sense 1, 2 words in sense 2 and 5 words in sense 3 are found to be distinct. This number of distinct words is very few since the sense bags consist of huge amount of words when all hierarchies are included as shown in Figure 3.26.

The most noticeable point here is that the number of common words in all these three sense bags. This number is 48 which is almost 2 times of distinct words in sense 1, 24 times of distinct words in sense 2 and almost 10 times of distinct words in sense 3. This evidence proves that only including more info from relations like hypernym of WordNet doesn't increases the number of overlaps between context bag and sense bags. This shows and proves that the information collected from WordNet contains more common information than the distinct information among the senses of polysemy word. Such information are not appropriate and cannot be used to distinguish the senses of a polysemy word.

Suppose two contexts *"A rabbit is inside the pen"* and *"... is writing a poem with a pen"*. The Figure 3.30 (b) shows the number of overlaps of these two contexts with the three sense bags of the first the three senses of noun ***Pen***. The first context has no overlaps with any sense. The number of overlaps of the second context is 1 with sense 1. This evidences shows that the number of overlaps depends upon the words that are used in sense's gloss. Sometimes there may not be the overlaps between context and any senses of the polysemy words. This is a big challenge for knowledge based overlap count WSD methods. The disambiguation in such WSD methods are found to be depend on the way how words are defined in dictionary or WordNet or any other lexical resources. The disambiguation should not be depend on the words that are used to define a gloss of a word. A noise information is produced when there are more number of overlaps between the context and the wrong sense of a polysemy word due to the more common information in that wrong sense with the context.


some artifact artefact man-made object taken whole whole unit assemblage parts regarded single entity how big part compared whole team unit object physical object tangible visible entity entity cast shadow full rackets balls objects physical entity entity physical existence entity perceived known inferred own distinct existence living nonliving (48)
Sense 1:
writing implement point ink flows writing implement implement used write implement instrumentation piece equipment tool used effect end instrumentality instrumentation artifact system artifacts instrumental accomplishing end (26)
enclosure consisting space enclosed purpose (5)
Sense 2:
confining livestock
(2)
Sense 3:
playpen portable babies left play (5)

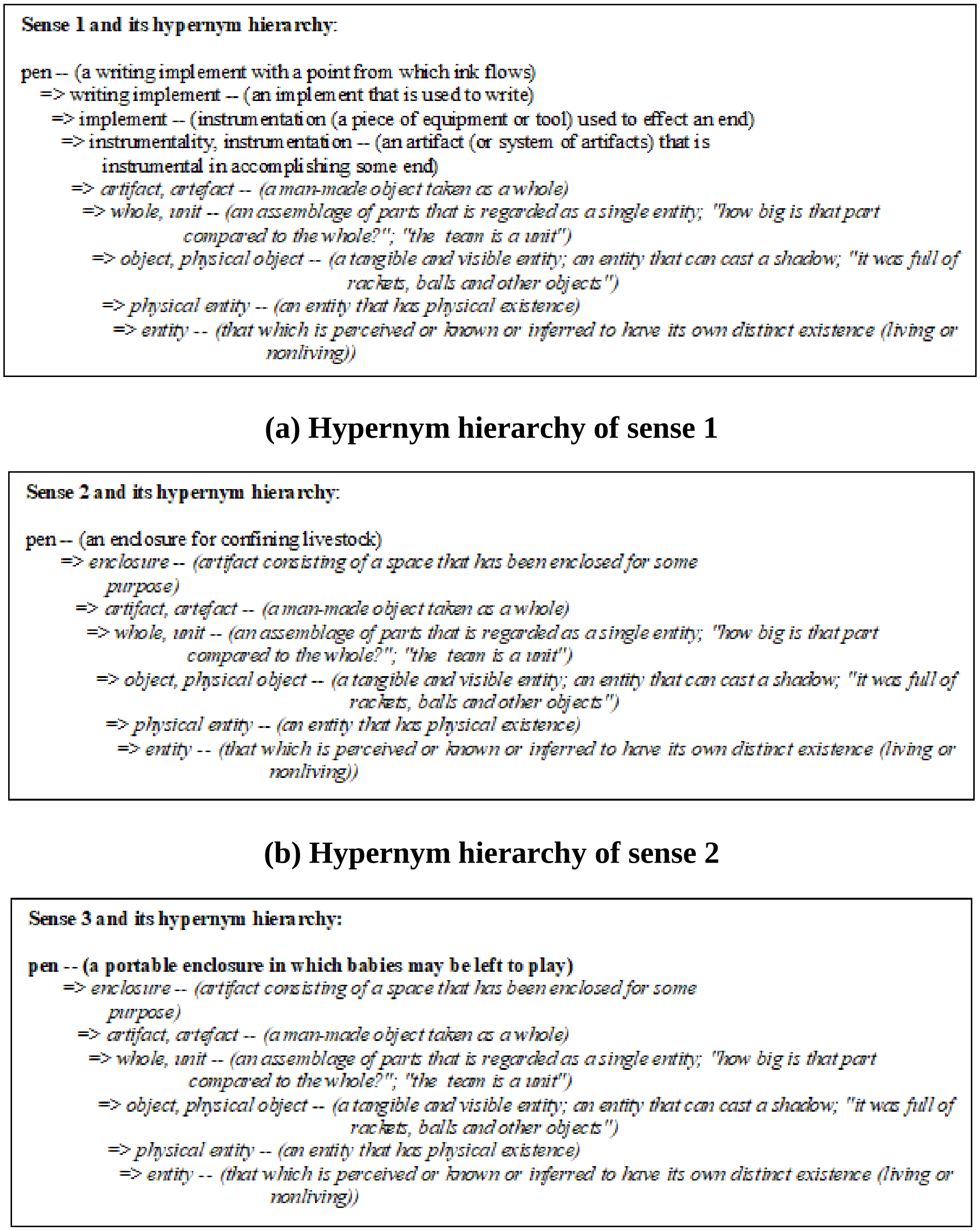
(a)

Context 1: ... is writing a poem with **a pen.**
Context Words: writing, poem
Overlaps: (writing) **1** overlap with sense-1 and no overlaps with other two senses

Context 2: A rabbit is inside the **pen.**
Context Words: rabbit
Overlaps: 0
No overlaps with any senses

(b)

Figure 3. 30: (a) Common and distinct words in the first three senses of *Pen* and (b) number of overlaps of two contexts with the first three senses of *Pen*



Sense 1 and its hypernym hierarchy:

pen -- (a writing implement with a point from which ink flows)
=> writing implement -- (an implement that is used to write)
=> implement -- (instrumentation (a piece of equipment or tool) used to effect an end)
=> instrumentality, instrumentation -- (an artifact (or system of artifacts) that is instrumental in accomplishing some end)
=> artifact, artefact -- (a man-made object taken as a whole)
=> whole, unit -- (an assemblage of parts that is regarded as a single entity; "how big is that part compared to the whole?"; "the team is a unit")
=> object, physical object -- (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets, balls and other objects")
=> physical entity -- (an entity that has physical existence)
=> entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

(a) Hypernym hierarchy of sense 1

Sense 2 and its hypernym hierarchy:

pen -- (an enclosure for confining livestock)
=> enclosure -- (artifact consisting of a space that has been enclosed for some purpose)
=> artifact, artefact -- (a man-made object taken as a whole)
=> whole, unit -- (an assemblage of parts that is regarded as a single entity; "how big is that part compared to the whole?"; "the team is a unit")
=> object, physical object -- (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets, balls and other objects")
=> physical entity -- (an entity that has physical existence)
=> entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

(b) Hypernym hierarchy of sense 2

Sense 3 and its hypernym hierarchy:

pen -- (a portable enclosure in which babies may be left to play)
=> enclosure -- (artifact consisting of a space that has been enclosed for some purpose)
=> artifact, artefact -- (a man-made object taken as a whole)
=> whole, unit -- (an assemblage of parts that is regarded as a single entity; "how big is that part compared to the whole?"; "the team is a unit")
=> object, physical object -- (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets, balls and other objects")
=> physical entity -- (an entity that has physical existence)
=> entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

(c) Hypernym hierarchy of sense 3

Figure 3. 31: The first three senses of noun *Pen* in WordNet 2.1 and their hypernym hierarchies

The causes of this noise information are the common hypernym hierarchies of multiple senses of the same polysemy words. There is no fixed rule for increasing the accuracy when more info is included from hypernym hierarchies. In many cases this inclusion of common hypernyms causes wrong disambiguation due noise information. The hypernym hierarchies follows “is a” relations. It means they have inheritance properties. When a lower level class inherits from upper class, its sibling classes contains the common properties as well. Those common properties cannot be used for distinguish the senses of same polysemy words. Rather their distinct properties should be used. Those common properties are the causes of induction of noise information. In some cases as in Figure 3.31 (b) and (c), only the glosses of senses are distinct and the whole hypernym hierarchy are same for all the senses of a polysemy words. The gloss’s average length in WordNet is 7 words [68]. This number of dintinct words will be very less and insufficient for sense disambiguation.

A closer look on the hypernym hierarchies of the first three senses of noun "Pen" from the English WordNet 2.1 is shown in Figure 3.31. It shows the first three senses of noun "Pen" along with the glosses and all hypernym hierarchy for each sense. From this, it is seen that the hypernym from the class "artifact" is common for all these three senses while in case of sense 2 and sense 3, only the gloss of each sense is distinct but the whole hypernym hierarchy for both sense is same. For disambiguation, distinct information is needed for each sense of polysemy word.

Suppose there are three objects of same class. To distinguish these three objects, the common properties inherited from the same parent class cannot be used. To clearly distinguish these object, their distinct properties should be used. Otherwise it is impossible to identify them using their common properties. In the same way, the common info from common hypernym hierarchy to multiple senses of the same polysemy cannot serve as a tool to distinguish their meaning. In the other hand, when the common hypernyms among the senses of the same polysemy words are excluded, the remaining information is very less and insufficient for sense disambiguation like the traditional dictionaries. These problems indicate to a direction that should be followed to develop a new lexical resource that provides a way of getting necessary and sufficient information for sense disambiguation.

3.5 PROBLEM STATEMENT

Based on the illustrations in section 3.4, the problems found in the overlap count knowledge based WSD algorithms using WordNet for sense disambiguation, are discussed in the following subsections.

3.5.1 Insufficient Information in WordNet for Disambiguation

In the illustration, it is seen that the WordNet has organized words in hierarchies. The lower levels of hierarchies are more semantically closer than the higher levels. It is already proved that only the gloss of word has less information for sense disambiguation. In addition, it was also seen that the hypernym hierarchies are common for many words and the higher level of hypernyms for different senses tend to similar. The common information is not useful for disambiguation. Seo et al. in [15] have also indicated from their experiment that the use of relatives' glosses of a word at higher level is not suitable. In addition, the definitions of words in the WordNet still don't contain sufficient information for disambiguation.

3.5.2 Noise Information and Wrong Disambiguation

The illustration in section 3.4 clearly indicates that there are many common hypernyms for the multiple senses of the same polysemy words and these common hypernyms are found to induce the noise information.

The noise information increase the number of overlaps of context with incorrect sense of polysemy word and thus cause the wrong disambiguation. If the hypernym are excluded to remove noise information, then the sense bag contains very less information as in dictionary and it is insufficient for disambiguation.

3.5.3 Disambiguation depends on the gloss's words

Many knowledge based WSD algorithms like adapted Lesk calculate the number of word overlaps between the context and the definition of words in dictionary or WordNet. The number of overlaps with context therefore depends on how the words in WordNet are defined.

The definitions of words are found different from one dictionary to another. In addition, they differ from one version to another version of the same dictionary as well. The number of overlaps may even differ when using a new version of lexical resource. Therefore, the disambiguation by such algorithm using WordNet depends on the gloss or definition of words. This is not fair for all contexts and therefore is not acceptable since it is not always constant and sufficient.

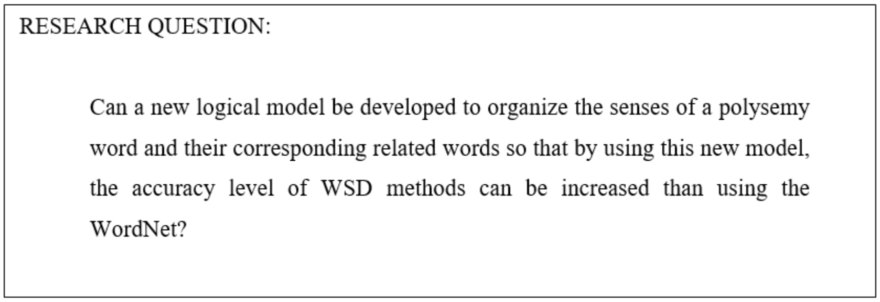
3.5.4 WordNet lacks dealing relations with polysemy words

WordNet has organized the words based on synonyms and connected the words using many relations like synset, antonym, hypernym, hyponym, holonym, meronym etc. However, it does not deal with the relationship of words with polysemy words which are the main cause of word ambiguity in any NLP tasks.

3.5.5 Research Question

The research formulated a research question which states:

> *"Can a new logical model be developed to organize the senses of a polysemy word and their corresponding related words so that by using this new model, the accuracy level of WSD methods can be increased than using the WordNet?"*



RESEARCH QUESTION:

Can a new logical model be developed to organize the senses of a polysemy word and their corresponding related words so that by using this new model, the accuracy level of WSD methods can be increased than using the WordNet?

Figure 3. 32: Research question

3.6 SOLUTION APPROACH

This section first presents the way how a human mind analyses the given context to understand the meaning of a polysemy word. It then describes how a context (containing related words) provides a clue to the correct meaning of a polysemy word for a given context and how human mind uses the context to disambiguate the sense of the polysemy word. Later, this section describes how the related words are generated, how these related words are organized and how these words are used for word sense disambiguation.

3.6.1 Human Mind and Word Sense Disambiguation

Suppose a context "She is eating bass”. Here, the word “bass” is polysemy word and its meaning needs to be disambiguated. When a human mind reads this context, the human mind is so intelligent that it finds the relation between “eating” and “bass”. It analyses these two words and concludes, what bass a human can eat is a fish bass. The word “eating” is a sufficient evidence for human mind to conclude the bass is a fish. These two words “eating” and “bass” are so connected and stored in human mind that they have strong relationship to disambiguate the sense of the word “bass” as a fish.

Suppose the same context with one more word- "She is eating bass with spoon". In the context, another word “spoon” has the relation with “eating” and “eating” has the relation with “bass”. The “spoon” is a means used to eat something and the bass fish can be eaten. These two words “spoon” and “eating” are so connected and stored in mind that the mind can conclude "spoon is used to eat something eatable". When the human mind reads this context, it finds the relation of “spoon” with “eating” and concludes so fast that what can be eaten with spoon is a bass fish. Here, the word “spoon” is supporting for the human mind to conclude the eaten bass in the given context is a fish. This is a way how a human mind analyzes the context and understands the meaning of polysemy words. These words “eating” and “spoon” are called as related words of word “bass” for its sense- "a fish".

Suppose a context "John likes bass". There is no any related word that is sufficient to disambiguate the meaning of the word “bass”. Even human cannot disambiguate the meaning in this context. John may like a bass fish or bass music. At least a sufficient related word to a polysemy word must be provided even for human to understand its meaning.

In human mind, the related words and the senses of polysemy words are so connected and stored that when human reads a context containing polysemy word, the human mind finds the related words, analyses and connects the related words with the respective sense of polysemy word. Motivated from such organization of polysemy words and their respective related words in human mind, PolyWordNet is developed to organize the polysemy words based on their related words.

3.6.2 Polysemy Words and Related Words

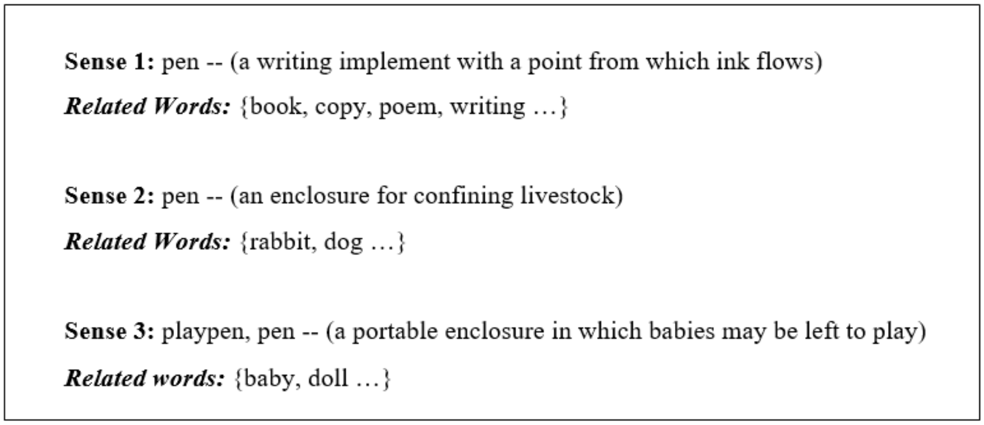
A polysemy word is a word which has multiple meanings according to the contexts where it is used. The context in which such word is used determines its meaning. A polysemy word without any context cannot be disambiguated even by a human. A context should be available or given to disambiguate a polysemy word. The words that determine correct sense of a polysemy word in a context are called their related words.

Suppose a context "The old tree has thick bark". In this context, the words “tree” and “thick” are related words of the polysemy word “bark” for its sense- “a tree bark”. These two related words provide sufficient context to human mind to understand the meaning of “bark”**.** If a polysemy word comes in a context, the context also contains at least another word that is sufficient to disambiguate the meaning of that polysemy word. Therefore, if the senses of polysemy words are semantically connected with their corresponding related words, the resulted lexical database can be used for word sense disambiguation to get exceptionally higher accuracy. Such lexical database can be used just like a human mind for word sense disambiguation.

This research strongly believes that if a context/sentence contains a polysemy word ***Pw***, it also simultaneously contains at least another word (or more than one word), say ***Rw***, together with the polysemy word. This word (s) ***Rw*** is called the related word (s) of $\boldsymbol{S_k}$ and can sufficiently disambiguate $\boldsymbol{S_k}$ in that context/sentence. A context simultaneously contains a polysemy word and its related word (s) together so that the related word clearly indicates the correct sense of that polysemy word. This research utilizes this fact and formulates a relation between $\boldsymbol{S_k}$ and ***Rw***.

Even human cannot disambiguate the meaning of a polysemy word if it is not used with any other word(s) to provide some context. When it is used with some other words providing a context, we can understand the sense. These related words need to be identified for each sense of polysemy word. The related words $\boldsymbol{Rw_k}$ are then connected with corresponding sense $\boldsymbol{S_k}$ of polysemy word ***Pw.*** The resulted organization of words is called **PolyWordNet.** This organization of words eliminates the noise information since a single related word is connected only with a single sense of a same polysemy word.

A related word can be a noun or a verb or an adjective or an adverb. The Figure 3.33 shows some related words for the first three senses of noun ***pen*** in WordNet 2.1. The prepositions can also serve as a very good related word for sense disambiguation. For example, for the context "in the bank" is the strong suggestive of the sense "financial institution" while the context “on the bank” is the strong suggestive for the sense “a river bank”. Therefore, the preposition which has the strong indication for a sense should be considered as a related word. The role of preposition for sense disambiguation must be analyzed in detail and can be another research work.



Sense 1: pen -- (a writing implement with a point from which ink flows)

Related Words: {book, copy, poem, writing …}

Sense 2: pen -- (an enclosure for confining livestock)

Related Words: {rabbit, dog …}

Sense 3: playpen, pen -- (a portable enclosure in which babies may be left to play)

Related words: {baby, doll …}

Figure 3. 33: Related words of first three senses of noun "pen"

3.6.3 Generation of Related Words

If a given context contains a polysemy word, the context should also contain at least one context word which must be able to provide a correct meaning of the polysemy word in the given context. Such context word or words are called as ***related words***. The ***related words*** are key component of **PolyWordNet**. It is because the related words are the only one components that can be used for word sense disambiguation using PolyWordNet.

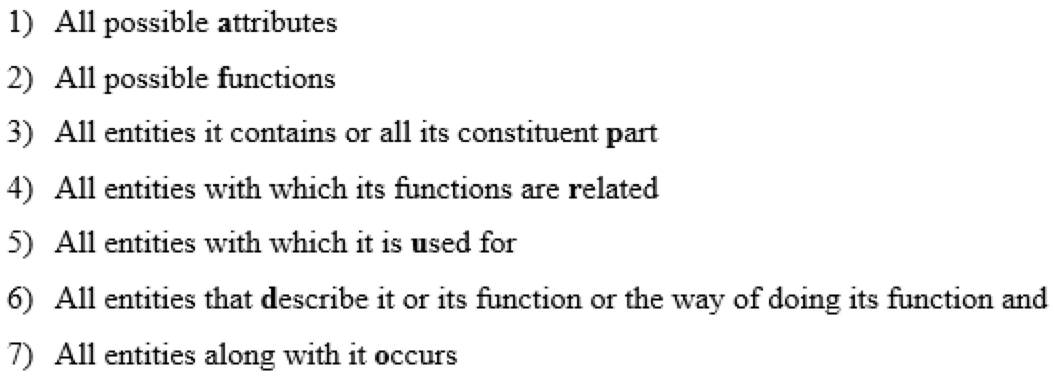
The task of finding a complete set of related words for senses of polysemy words is challenging and most difficult task. To generate a complete and perfect set of related words obviously needs further continuous research works. A good related word should have two characteristics. It shouldn't create ambiguity during disambiguation. It should be enough capable to disambiguate the sense in the given context.

Principle Idea for Generating Related Words

After a detail observation in the collected Test Sentences, it is found that the related words of senses of a polysemy word are either attributes, or functions they perform or its constituent parts or all entities with which its functions are performed or all entities with which it is used or all entities along with it occurs. Therefore, the principle idea to generate the related words to build PolyWordNet is to collect all these information about the senses of polysemy words.

Rules for Generating Related Words

Based on the principle idea used for generating related words, a set of simple rules are designed to find out the related words for the senses of polysemy words. To generate a set of related words, say $\boldsymbol{Rw_k}$ for a sense, say $\boldsymbol{S_k}$ of a polysemy word say, ***Pw***. Consider $\boldsymbol{S_k}$ as an object. Then, for each sense $\boldsymbol{S_k}$ of ***Pw,*** collect following information (if applicable):



1) All possible attributes
2) All possible functions
3) All entities it contains or all its constituent part
4) All entities with which its functions are related
5) All entities with which it is used for
6) All entities that describe it or its function or the way of doing its function and
7) All entities along with it occurs

The related words for all the senses of a polysemy words are generated at a time. After generating the related words for all senses of the polysemy words, if the two or more senses of the same polysemy words have same related words, such words are removed from all senses.

The sets of related words of multiple senses of a polysemy words contains only the distinct words. These sets of senses of the same polysemy word cannot contain the common word. This is because the common related words leads again to an ambiguity in sense as in WordNet.

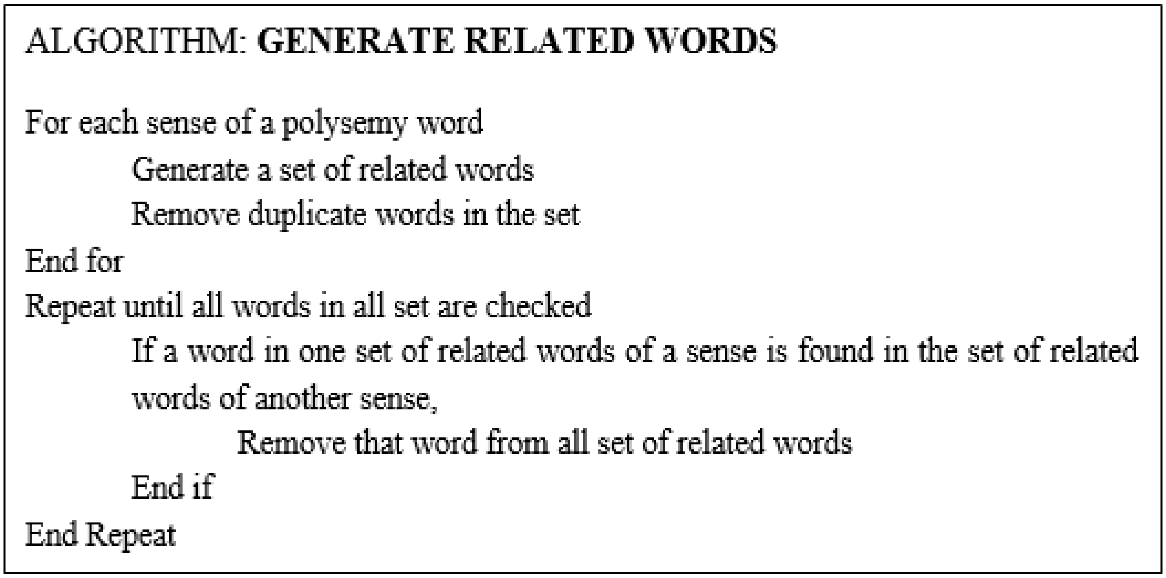
A related word of a sense of a polysemy word can be a related word of a sense of another polysemy word. Similarly, a sense of a polysemy word can also be a related word of a sense of another polysemy word.

Algorithm: Generation of Related Words

The related-word-generation algorithm is shown in Figure 3.34. After collecting info for each sense of a polysemy word, the info of all sense of a polysemy word are further processed. The duplicate words for a sense are removed. The words that are common for more than one senses of the same polysemy words are removed.

Generation of Related Words for Two Senses of Polysemy Word "Pen"

The Table 3.2 shows the generation of related words for the two senses of noun ***Pen*** in WordNet 2.1. The first two senses of ***Pen*** are 1) writing implement (**Sense 1**) and 2) enclosure for confining livestock (**Sense 2**). The related words for these senses are manually generated by using the rules shown in the first column of the table. The second column shows the related words for the first sense of ***Pen***. The “cylinder” is a shape attribute for a writing implement. The functions that a writing implement can do is to “write”. The constituents parts of a writing implement can be “cap”, “nib” etc. Similarly, for sense 2 of ***Pen***, “cuboid” is a shape attribute. The functions the enclosure for confining livestock can do is to “provide shelter”. Its constituent parts include “door”, “roof” and “panel”. In this way, related words are generated by using designed rules.

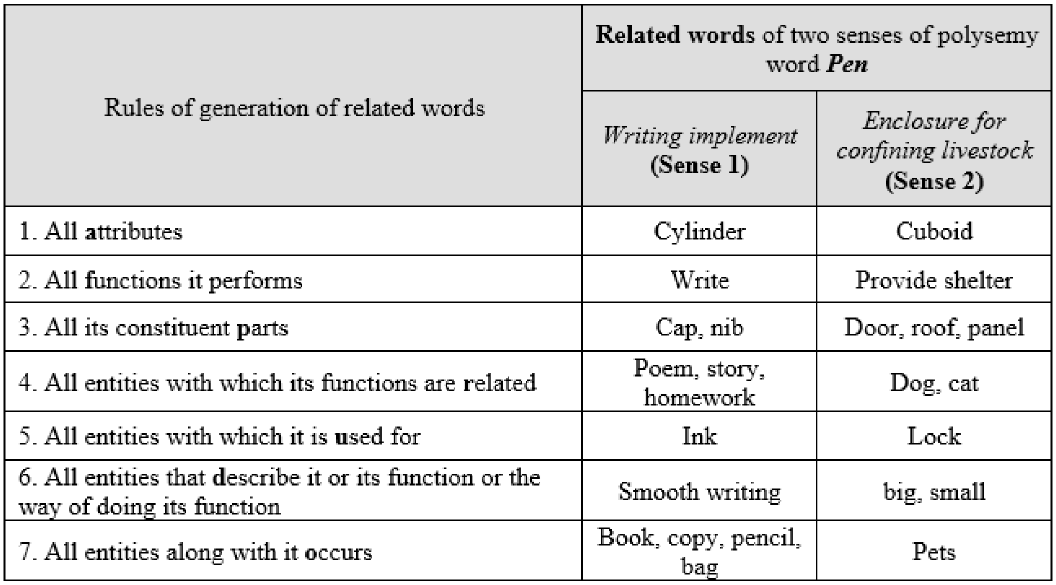


```
ALGORITHM: GENERATE RELATED WORDS

For each sense of a polysemy word
    Generate a set of related words
    Remove duplicate words in the set
End for
Repeat until all words in all set are checked
    If a word in one set of related words of a sense is found in the set of related
    words of another sense,
        Remove that word from all set of related words
    End if
End Repeat
```

Figure 3. 34: Algorithm for generating related words

The related words are manually generated by respondents of this research to build the PolyWordNet. If these related words could be automatically generated, it would be of great worthy. This research, therefore, recommend for auto-generation of related words to create a self-generating PolyWordNet.

Table 3. 2: Related words of first two senses of the noun "*pen*" in WordNet 2.1



Rules of generation of related words	**Related words** of two senses of polysemy word ***Pen***	
	Writing implement **(Sense 1)**	*Enclosure for confining livestock* **(Sense 2)**
1. All attributes	Cylinder	Cuboid
2. All functions it performs	Write	Provide shelter
3. All its constituent parts	Cap, nib	Door, roof, panel
4. All entities with which its functions are related	Poem, story, homework	Dog, cat
5. All entities with which it is used for	Ink	Lock
6. All entities that describe it or its function or the way of doing its function	Smooth writing	big, small
7. All entities along with it occurs	Book, copy, pencil, bag	Pets

3.6.4 Organization of words- *PolyWordNet*

The related words are the key words for PolyWordNet. Once they are created, they are then linked with senses of polysemy words to which they are related. The resulted lexical resource is named as **'PolyWordNet'**. PolyWordNet organizes the sense of polysemy words based on their respective related words which disambiguate their meaning in a context. The words are categorized as verbs, nouns, adverbs and adjectives. Links between related words and senses of polysemy words are stored in one of these categories based on the words are connected with which part of speech. If a word of any parts of speech is connected with another a noun word, then the links is stored in noun category. If the word is connected with a verb, then the link is stored in verb category [83]. A related word is connected with only one sense of the same polysemy word. A related word cannot be connected with more than one sense of the same polysemy word. If a related word is equally related with multiple senses of the same polysemy word, it cannot be connected as the related word of either sense. It is just ignored. The reason is that it leads to multiple senses of a polysemy causing another ambiguity. However, a same word can be connected with single sense of multiple polysemy words. Most importantly, a sense of a polysemy word can be a related word of a sense of another polysemy word.

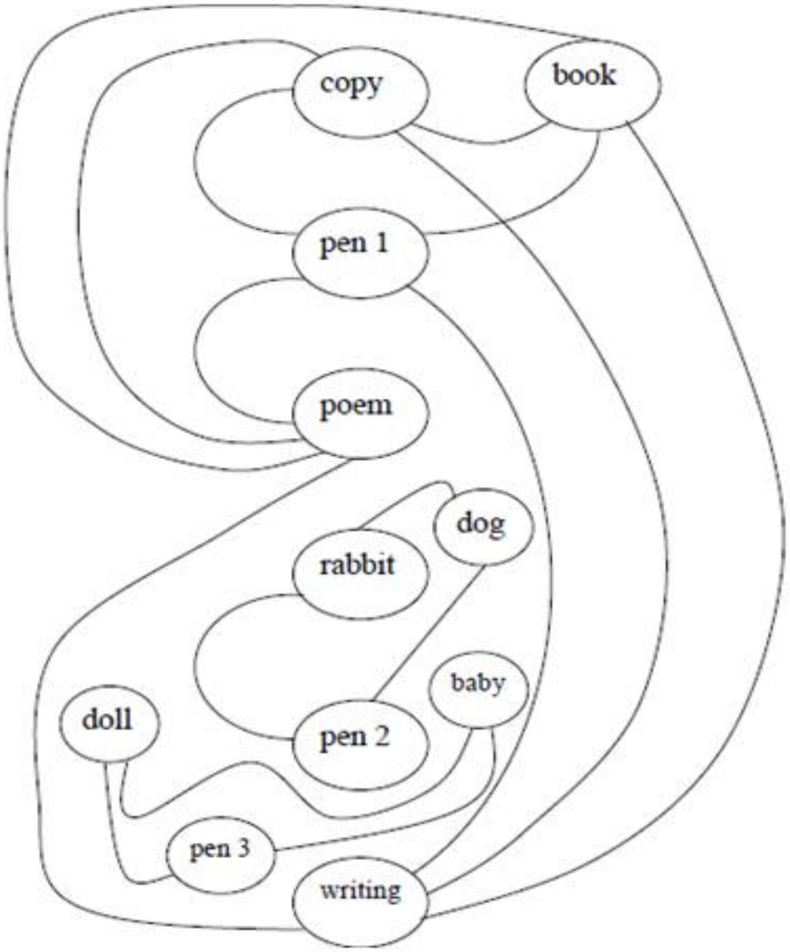


Figure 3. 35: Organization of words in PolyWordNet

The Figure 3.35 shows a part of PolyWordNet. It shows 3 senses of “pen” and their related words. The sense “pen1”, which means “*a writing implement*…. ”, has a set of related words {copy, book, poem write}. Sense “pen2”, which means “*an enclosure for confining*…. “, has a set of related words {rabbit, dog} and the sense “pen3”, which means “*a portable enclosure in which babies*….”, has a set of related words {doll, baby}.

The links between words are of two types. The first type of links are direct links in which a related word is directly linked with a sense of a polysemy word. For example, a link from “copy” to “pen1”. The second type links are indirect links in which one word is connected with many other words in a chain fashion and finally connected to its respective sense of a polysemy word. For example, a link from “book” to “poem” to “writing” to “copy” and finally to “pen1”. The words are connected in chain since they are related each other and can describe an object together.

The same words in WordNet are linked with multiple senses of same polysemy word. Such words which are linked with multiple sense of the same word cannot be utilized to disambiguate the meaning of polysemy word. This introduces noise information. This problem is resolved by connecting one word with a single sense of the same polysemy word in PolyWordNet.

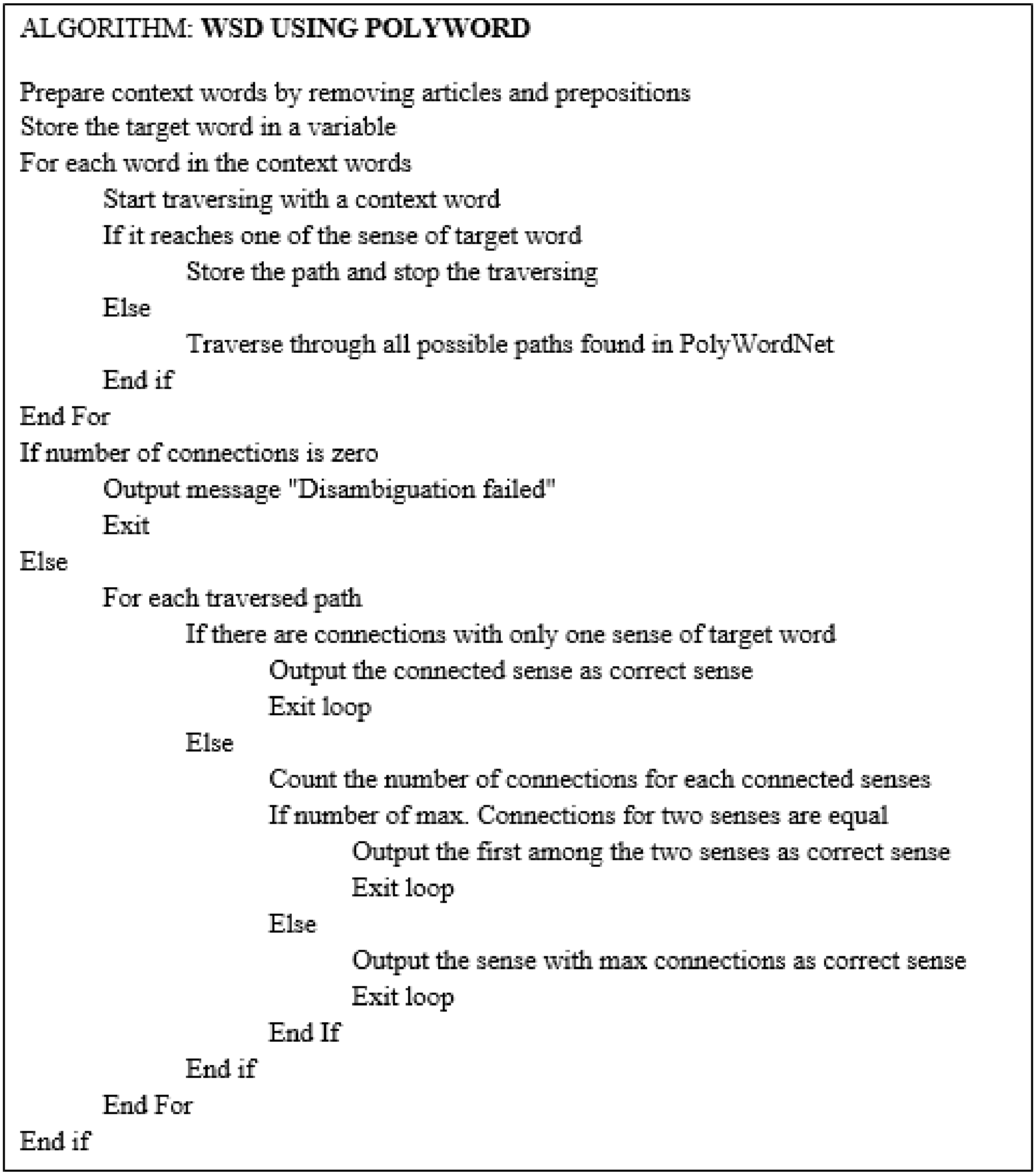
3.6.5 Disambiguation Process: A New WSD Algorithm

The previous subsection 3.6.4 discussed about how the related words and polysemy words are organized in the PolyWordNet. This subsection presents a simple WSD algorithm which uses PolyWordNet for word sense disambiguation. The purpose of this algorithm is only to access the relations of PolyWordNet for sense disambiguation. It is a very simple algorithm which can be used for word sense disambiguation using PolyWordNet. The aim of this research is to develop a new lexical database but not to develop an optimized algorithm for word sense disambiguation.

Description of Algorithm

The new WSD algorithm used in this research does not count the number of overlapped words. In contrast, this algorithm searches paths between context words and a sense of the polysemy word. If a path(s) between a context word (s) and a sense of a polysemy word is found, the sense is returned as a correct sense. However, if paths are found to connect multiple senses of the polysemy word, the sense with greater number of paths is returned as a correct meaning.

If the paths for two or more senses of the same polysemy word are found equal, the first sense is returned as correct sense. If no path is found, the algorithm returns a failure. Figure 3.36 shows the new WSD algorithm which uses PolyWordNet for sense disambiguation.



```
ALGORITHM: WSD USING POLYWORD

Prepare context words by removing articles and prepositions
Store the target word in a variable
For each word in the context words
    Start traversing with a context word
    If it reaches one of the sense of target word
        Store the path and stop the traversing
    Else
        Traverse through all possible paths found in PolyWordNet
    End if
End For
If number of connections is zero
    Output message "Disambiguation failed"
    Exit
Else
    For each traversed path
        If there are connections with only one sense of target word
            Output the connected sense as correct sense
            Exit loop
        Else
            Count the number of connections for each connected senses
            If number of max. Connections for two senses are equal
                Output the first among the two senses as correct sense
                Exit loop
            Else
                Output the sense with max connections as correct sense
                Exit loop
            End If
        End if
    End For
End if
```

Figure 3. 36: A new WSD algorithm using PolyWordNet

Inputs of Algorithm

The algorithms needs two inputs- 1) a context and 2) a polysemy word that need to be disambiguated in the context. The context can be a simple sentence or a compound sentence. The polysemy word that need to be disambiguated is also called a target word.

Sense Disambiguation Process

All the punctuations, prepositions, pronoun and articles are removed from the given sentence. The remaining words form a context bag. A context bag is a set of all words that are in a given context after removing all punctuations, prepositions, pronoun and articles. All the possible senses of target word are collected from the PolyWordNet to form a sense bag. A sense bag is a set of all senses of the target word that need to be disambiguated. For each word in context bag, the algorithm searches a path(s) from the word to one of the sense of the target word in sense bag using relations from PolyWordNet by exploring all connected words. This is repeated for all context words. If a path(s) between a word (s) in context bag and a single sense in sense bag, the sense as a correct meaning is sent for output. If paths are found for multiple senses in the sense bag, the sense with greater numbers of path connection is sent for output as a correct meaning. If the greater number of paths for more than one sense are found equal, then the first sense in the PolyWordNet is sent for output as a correct meaning. If no path is found, then sense disambiguation fails. If either all the context words in context bag or the target word are not found in the PolyWordNet, in such case also the sense disambiguation fails.

Output of Algorithm

The algorithm provides a single output. The output of the algorithm is the Word_ID of the calculated sense along with its gloss. The calculated sense by the algorithm can be correct or incorrect. If the algorithm fails to disambiguate the sense, there will be no output at all.

Illustration: To disambiguate the sense of “Pen” in context “Maria bought copy and pen”

Suppose a part of the PolyWordNet which is shown Figure 3.35. Consider a simple context "*Maria bought copy and pen"*. The context bag is a set {bought, copy} and the word to be disambiguated is {pen}. Few possible connecting paths from copy to pen are copy->pen1, copy->book->poem->pen1, copy->book->poem->writing->pen1,copy->poem->pen1, copy->writing->poem->pen1, copy->writing->pen1, copy->writing->poem->pen1 and so on. The word “brought” is not in the part of the PolyWordNet as shown in Figure 3.35. All connecting paths connects the context word “copy” with the sense 1 of “pen”. The algorithm select the "pen 1" sense of “pen” as a correct meaning for that context.

3.6.6 Mathematical Model- A New WSD method using PolyWordNet

In PolyWordNet, the way of organization of words is different than that of the dictionary and the WordNet. In PolyWordNet, the senses of a polysemy words and their related words are put together. Since the lexical database deals with the organization of collection of words, the principles of set theory and concept of relations are applied to model the organization of words in the PolyWordNet and the way how the new WSD algorithm uses the relations from PolyWordNet for sense disambiguation.

A PolyWordNet is a large set of relations between related words and their corresponding senses of polysemy words. The new WSD algorithm uses these relations of words from PolyWordNet to find the connection paths from words in a given context to the senses of the polysemy words. If it could find a path connecting the context words and a sense of the polysemy word, that sense is sent to output as a correct sense in the given context. The algorithm every time checks the relation between words starting from a word in context bag to reach to one of the sense of polysemy word. If it finds the path it assigns a value 1, otherwise assigns 0. The value 1 indicates, the algorithm is able to find a meaning for that polysemy words. If there are paths reaching to the same sense of the polysemy word, then that sense returned as a correct sense. If multiple paths are reaching to more than one sense of the same polysemy words, then the sense with multiple connecting paths is returned as a correct sense. In some cases, there may not be any paths that connect the context words and any sense of the polysemy word. In such case, the algorithm fails to disambiguate the sense of the polysemy word. This may occur when there are no context words or polysemy words in the PolyWordNet.

PolyWordNet

Suppose ***pw*** be a polysemy word. Assume ***pw*** has *n* different senses: - $\boldsymbol{S_1, S_2, S_3, S_4 \ldots S_n}$. Each sense $\boldsymbol{S_i}$ (where *i = 1, 2, 3, 4, 5... n*) has some related words. Suppose $\mathbf{RW_{si}}$ is a set of related words for $\boldsymbol{S_i}$ of ***pw***. Then PolyWordNet ***P*** is a collection of relations (*say* **Rel**) between $\boldsymbol{S_i}$ *and* $\mathbf{RW_{S_i}}$ for all polysemy words ***pw*** and is defined as

$$\mathbf{RW_{S_i}} = \{W : W \textit{ is a word that is strongly related with the sense } \boldsymbol{S_i} \textit{ of polysemy word } \boldsymbol{pw}\} \qquad \textbf{(3.1)}$$

$$\mathbf{Rel} = \left\{\boldsymbol{S_i}, \mathbf{RW_{S_i}} : \textit{each word in } \mathbf{RW_{S_i}} \textit{is connected with } \boldsymbol{S_i}\right\} \qquad \textbf{(3.2)}$$

$$\boldsymbol{P} = \{x : x \textit{ is a relation } \boldsymbol{Rel}\} \qquad \textbf{(3.3)}$$

Word Sense Disambiguation using Relations of PolyWordNet

Suppose, in a given sentence, ***pw*** is a target word which is polysemous and has ***n*** different senses - $\boldsymbol{S_1, S_2, S_3, S_4 \ldots S_n}$. The exact meaning of ***pw*** is to be determined using the context of given sentence. The remaining words in the sentence are context words. Suppose, there are ***k*** different context words $\mathbf{CW_k}$ where *k = 1,2,3,… , k* in the sentence.

The exact sense $\boldsymbol{S_i}$ **is** determined if there exists a function $\boldsymbol{f(}\mathbf{CW_k}, S_i\boldsymbol{)}$ in PolyWordNet ***P*** such that any one context word $\mathbf{CW_j}$ is in $\mathbf{RW_{S_i}}$ and there is a path from $\mathbf{CW_j}$ to $\boldsymbol{S_i}$**.** The function $\boldsymbol{f(}\mathbf{CW_j}, \boldsymbol{S_i)}$ is mathematically defined as

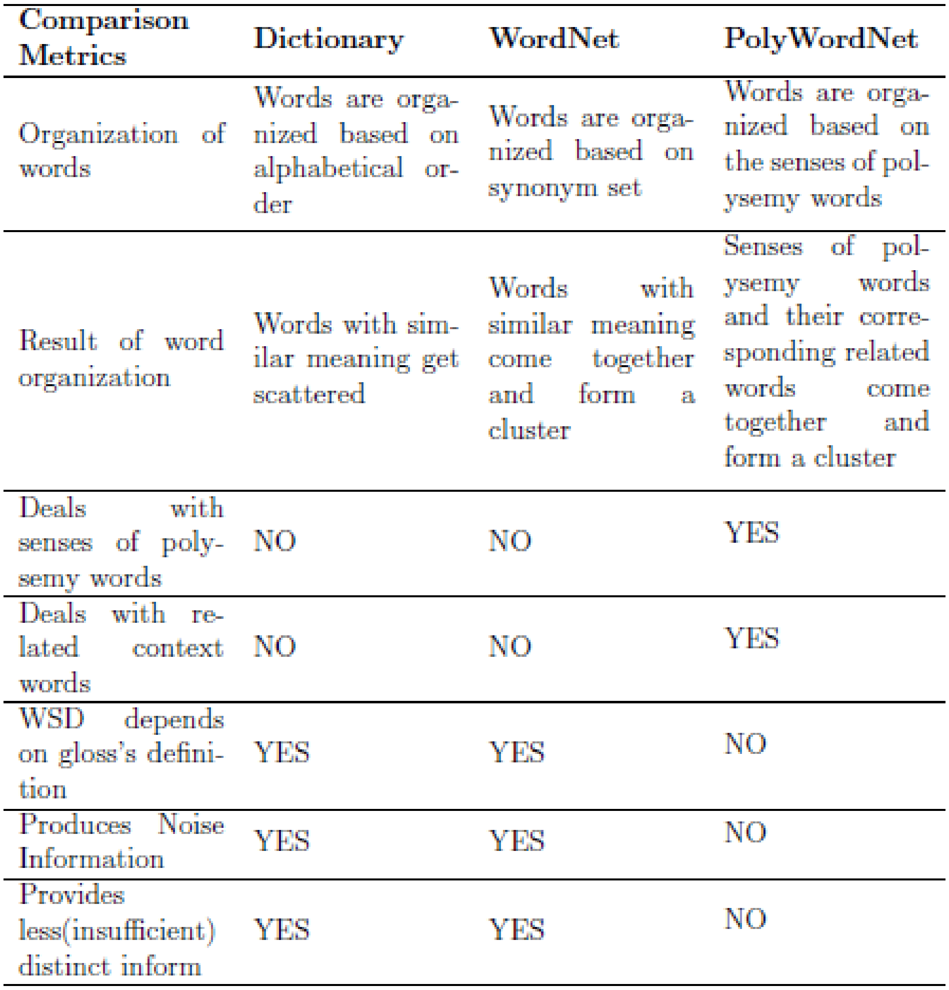
$$f(\boldsymbol{CW_j}, S_i) = \begin{cases} 1, & \textit{if } \boldsymbol{CW_j} \text{ is in } \mathbf{RW_{S_i}} \text{ and there is a path from } CW_j \text{ to } \boldsymbol{S_i} \text{ in } \boldsymbol{P} \\ 0, & \textit{Otherwise} \end{cases} \qquad \textbf{(3.4)}$$

Where the function $\boldsymbol{f(}\mathbf{CW_j}, \boldsymbol{S_i)}$ is a relation **Rel** in PolyWordNet ***P***. If there exists a relation between any word $\mathbf{CW_j}$ in the given sentence and $\boldsymbol{S_k}$ sense of the polysemy word ***pw*** in the PolyWordNet *P*, then the correct meaning of ***pw*** is $\boldsymbol{S_k}$**.** If there does not exists any relation between $\mathbf{CW_j}$ and $\boldsymbol{S_i}$ of ***pw*** in the PolyWordNet *P*, then the sense cannot be disambiguated.

3.7 POLYWORDNET, DICTIONARY AND WORDNET

PolyWordNet deals with the polysemy words and brings the senses of polysemy words and their related words together forming a cluster. The PolyWordNet does not contain any related word which is common to multiple senses of the same polysemy word. A same word can be a related word of a sense of different polysemy words. A sense of a polysemy word can be a related word of a sense of another polysemy word.

Table 3. 3 : Comparison among Dictionary, WordNet and PolyWordNet



Comparison Metrics	Dictionary	WordNet	PolyWordNet
Organization of words	Words are organized based on alphabetical order	Words are organized based on synonym set	Words are organized based on the senses of polysemy words
Result of word organization	Words with similar meaning get scattered	Words with similar meaning come together and form a cluster	Senses of polysemy words and their corresponding related words come together and form a cluster
Deals with senses of polysemy words	NO	NO	YES
Deals with related context words	NO	NO	YES
WSD depends on gloss's definition	YES	YES	NO
Produces Noise Information	YES	YES	NO
Provides less(insufficient) distinct inform	YES	YES	NO

In a given context, if there is even a single related word and is connected to respective sense of a polysemy word, it is sufficient to find the correct meaning of the polysemy word using

PolyWordNet. This resolves the problem of insufficient information for sense disambiguation while using WordNet.

A brief comparison among the dictionary, WordNet and PolyWordNet is shown in Table 3.3. PolyWordNet is built based on the senses of polysemy words. The Table 3.3 compares the PolyWordNet based on seven metrics which include 1) the way of organizing words, 2) result of the organization of words, 3) whether it deals with polysemy words or not, 4) whether it deals with the related words or not, 5) whether the WSD that uses PolyWordNet depends on gloss's definition or not, 6) whether it produces noise information during disambiguation process or not and 7) whether the information provided for sense disambiguation is sufficient or not.

The PolyWordNet resolves the problems of noise information and insufficient information for sense disambiguation. It organizes the senses of polysemy words based on their related words. The resulted PolyWordNet is, therefore, especially suitable for word sense disambiguation. There is no any other lexical database that deals with the polysemy words and their related words.

Chapter 4

Research Methodology

This chapter describes the way followed systematically to answer the stated research question and to achieve the objectives and aim of this research. Research methodology is explained in the following sections: - research design, participants, and validation methods which are used to validate the collected data, test sentences and new lexical database-PolyWordNet.

4.1 RESEARCH DESIGN: EXPERIMENTAL

This research work needs experimental evidence to check whether the organization of words in new lexical database is acceptable for word sense disambiguation or not. For this, the similar lexical databases like dictionary and WordNet are taken as references. Altogether 7 different experiments are designed and these are run 63 times by changing parameters like amount of data and number of Test Sentences. A detail description of purpose of the experiments and the different experimental settings are discussed in sub-section 4.1.1.

4.1.1 Experimental Designs

This sub-section describes the purpose of the experiments and the experimental setup details based on different rationales and intents.

Purpose of Experiments

The aim of this research is to develop a new lexical database which organizes the senses of polysemy words based on their corresponding related words. The polysemy words are the big problem in any NLP tasks. However, till now no lexical databases deal with the organization of polysemy words. The intent to organize the senses of polysemy words based on their corresponding related words is to bring the related words and the senses of polysemy words together. If a new lexical database can be developed and it is able to connect all the related words of all senses of polysemy words, then the big problem of polysemy words in NLP task will be resolved greatly.

The proposed new lexical database is PolyWordNet. It organizes the senses of polysemy words based on their corresponding related words. Therefore, in PolyWordNet, the senses of polysemy words and their corresponding related words get linked together and form clusters. This organization of words can be used to disambiguate the senses of polysemy words with better accuracy. However, some experiment results should provide the evidences to accept the organization of words in PolyWordNet. Since the PolyWordNet is completely new type of lexical database, it cannot be directly compared with the exactly same lexical database to observe its acceptability. Therefore, a similar and popular lexical database- WordNet is chosen as a reference lexical database.

It is important to note a point that the PolyWordNet and WordNet are not compared in this research to check which one is best. Since the WordNet is a widely used popular lexical database for word sense disambiguation by knowledge-based approaches, it is chosen as a reference lexical database to prove the word organization of PolyWordNet is acceptable for word sense disambiguation. How the WordNet is used as a reference lexical database is as follows:

A sample amount of data is taken from WordNet. It is used to build the PolyWordNet. Since the motivation towards this research is originated from the problems found in overlap-count knowledge-based WSD algorithms, the simplified Lesk algorithm is chosen for the experiments which uses the WordNet as a reference lexical database for word sense disambiguation. Therefore, a set of experiments which use WordNet and simplified Lesk algorithms are designed. Similarly, another set of experiments which use PolyWordNet with its simple algorithm are designed. The data and the number of Test Sentences are kept constant during disambiguation process in both WordNet and PolyWordNet. The results obtained in the both sets of experiments are compared and analyzed. If the range of the accuracies of the experiments using PolyWordNet are matched at least or found to be greater, then the word organization of PolyWordNet can be accepted for word sense disambiguation with reference to the WordNet. The purpose of the experiments designed in this research is to analyze the acceptability of the organization of words in PolyWordNet by taking the WordNet as a reference lexical database.

Experimental Settings

Altogether 7 different experiments are set up. These experiments are named as 1) ***Exp 1 Run 1,*** 2) ***Exp 1 Run 2,*** 3) ***Exp 2 Run 1,*** 4) ***Exp 2 Run 2,*** 5) ***Exp 3 Run 1,*** 6) ***Exp 3 Run*** and 7) ***Exp 4***. These **7** different experiments are repeated and run **9** times diving into 6 series. This results 63 runs in total. The 6 series are named as Series A, Series B, Series C, Series D, Series E and Series F. The 7 experiments are run in Series A to Series E by increasing the amount of data in PolyWordNet and by increasing the number of Test Sentences (TS). The total amount of data in PolyWordNet is 3541.

Series A to Series E experiments are tested by test sentences generated in this research. The total number of these test sentences generated in this research are 2905. The Series F experiments are run only by increasing the number of test sentences which are randomly taken from the ***news*** category of Brown Corpus. The total number of these test sentences taken from Brown corpus are 1200. Therefore, total number of test sentences used in this research are 4105.

The first 6 experiments out of 7 have used simplified Lesk algorithm and have utilized information from WordNet for sense disambiguation. The difference in these 6 experiments is the amount of information utilized for word sense disambiguation. As moving from experiment 1 to 6, the amount of information is increased in every next experiment. The seventh experiment (i.e. ***Exp 4)*** uses the new WSD algorithm and PolyWordNet for sense disambiguation.

The first 6 experiments are actually variations of 3 experiments each of which has two runs. The two runs are named as Run 1 (*also called as* Run A) and Run 2 (*also called as* Run B). The detail experimental settings of the 7 experiments are described in the following subsection.

a) Experiment 1: - *Exp 1 Run 1* and *Exp 1 Run 2*

Rationale: The rationale behind carrying out this experiment is that the information taken only from the word's gloss from WordNet is not sufficient for sense disambiguation.

Intent: The intent of this experiment to represent the knowledge-based contextual overlap count WSD method that uses information only from synset and gloss of word in WordNet.

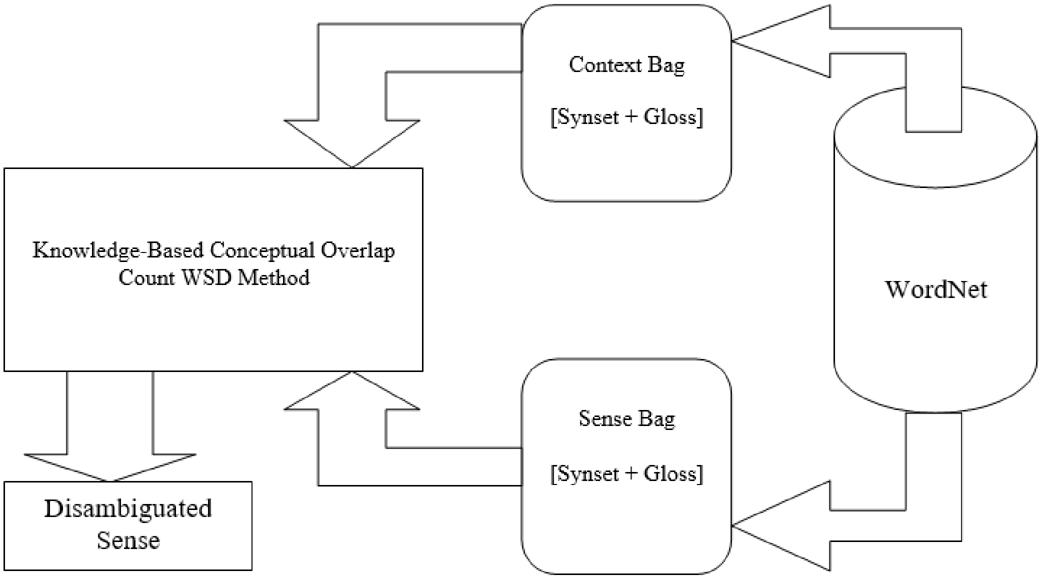
Experimental Setting: In first run- **Exp 1 Run 1**, only the synset and gloss of words from WordNet are used to form sense bags and context bag. In the second run**- Exp 1 Run 2**, each word's hypernym from WordNet are also added to sense bags to observe the effect of increased info from hypernyms. The Figure 4.1 shows the experimental setting of Experiment 1.

b) Experiment 2: - *Exp 2 Run 1* and *Exp 2 Run 2*

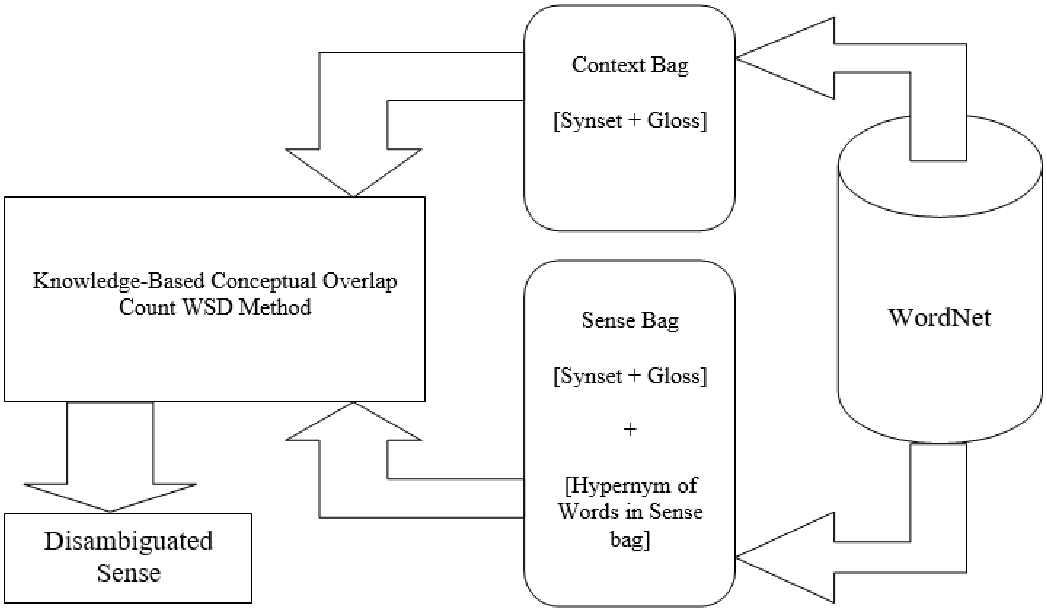
Rationale: The rationale behind carrying out this experiment is that if the information in the sense bag and context bag is increased, this will increase in the relatedness of correct sense with the context.

Intent: The intent of this experiment to represent the knowledge based contextual overlap count WSD method which utilizes information only from synset, gloss and hypernyms of word in WordNet.

Experimental Setting: In **Exp 2 Run 1**, the synset, gloss and hypernyms of words from WordNet are used to form sense bags and context bag. In **Exp 2 Run 2**, the same information as in **Exp 2 Run 1** are utilized plus word's hypernym from WordNet are also added to each sense bags. The Figure 4.2 shows the experimental setting of Experiment 2.

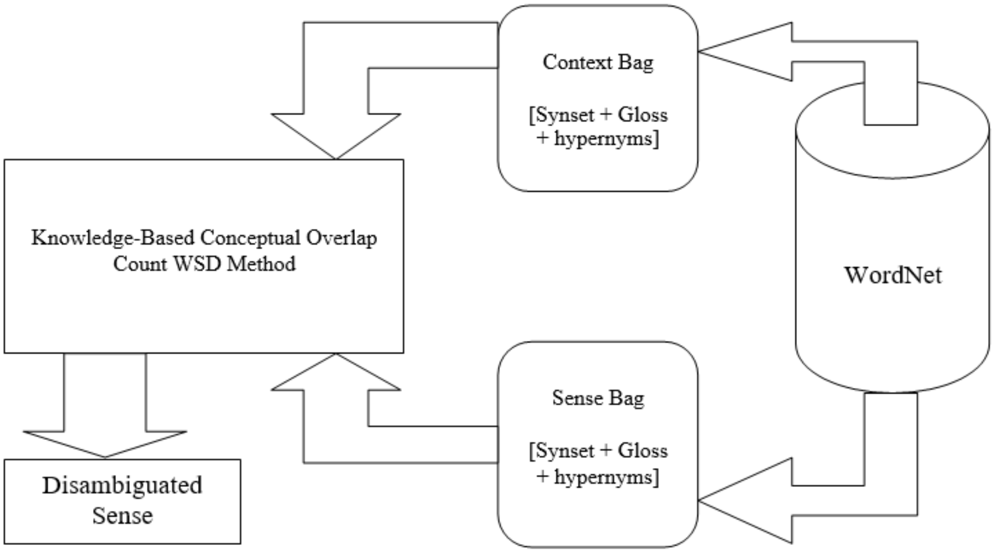


a) **Exp 1 Run 1**

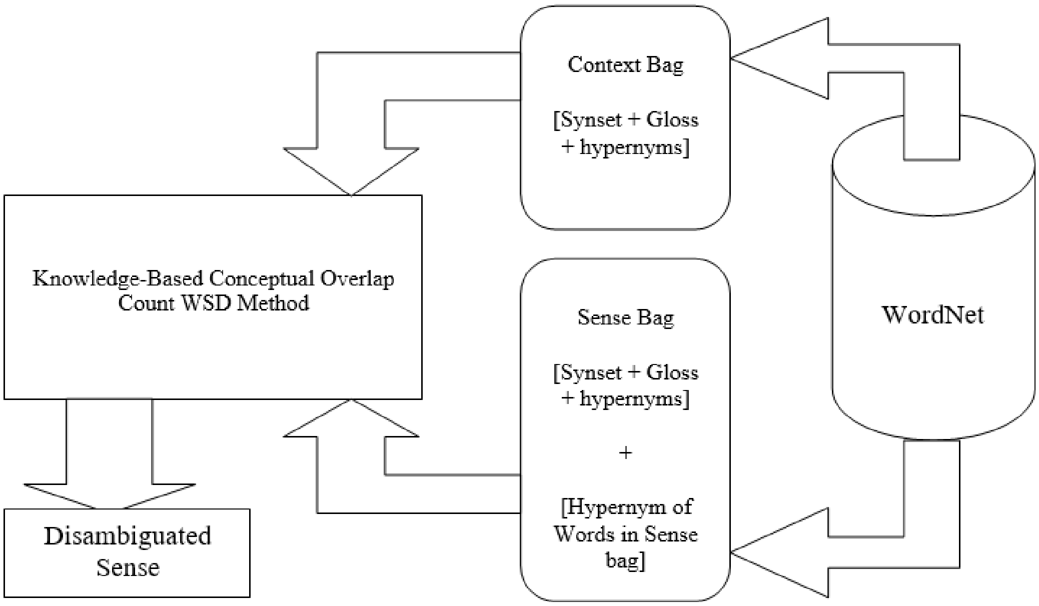

Context Bag
[Synset + Gloss]
WordNet
Knowledge-Based Conceptual Overlap Count WSD Method
Sense Bag
[Synset + Gloss]
+
[Hypernym of Words in Sense bag]
Disambiguated Sense

b) **Exp 1 Run 2**

Figure 4. 1: Setting of Experiment 1 (a) Exp 1 Run 1 and (b) Exp 1 Run 2

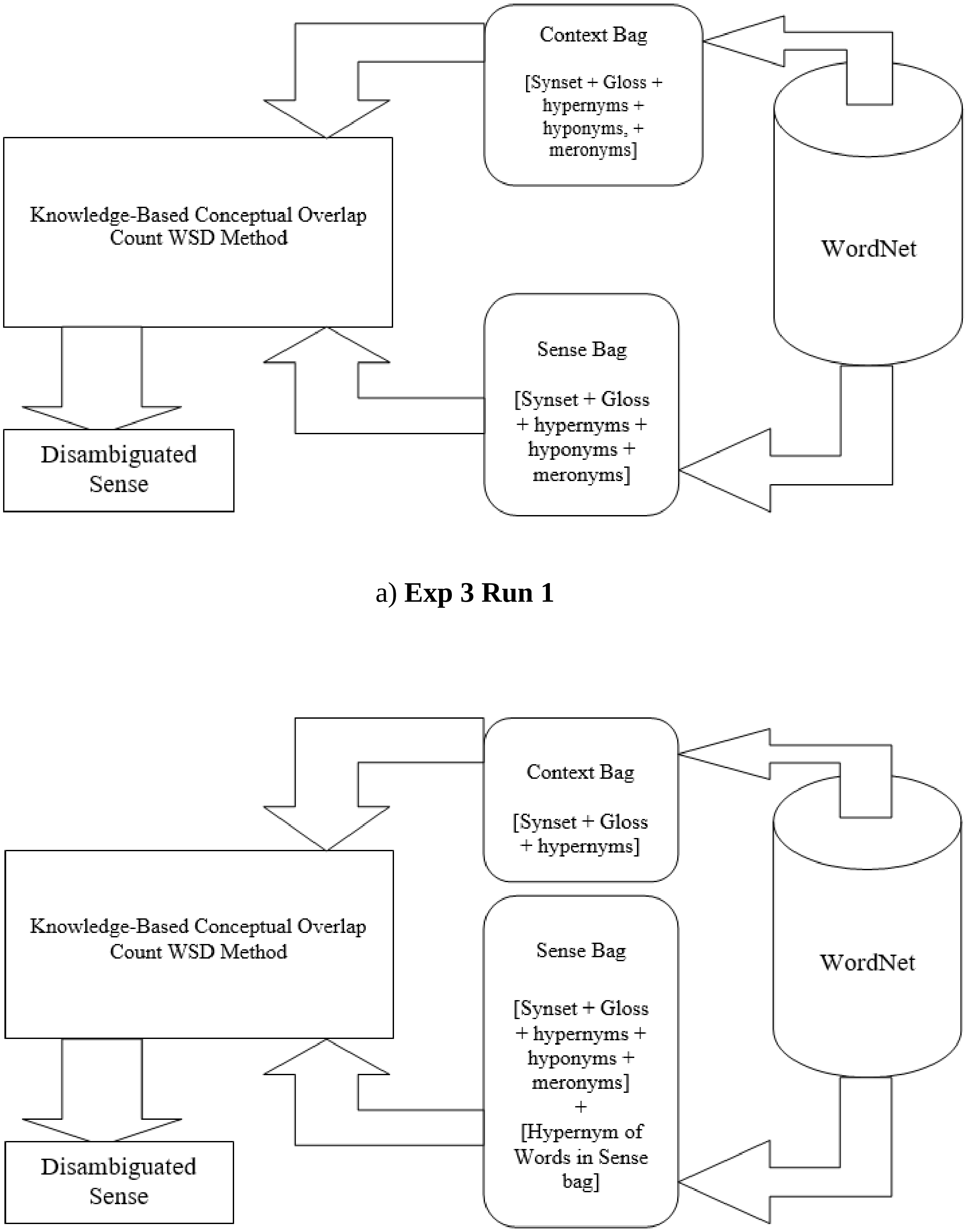

Context Bag
[Synset + Gloss + hypernyms]
Knowledge-Based Conceptual Overlap Count WSD Method
WordNet
Sense Bag
[Synset + Gloss + hypernyms]
Disambiguated Sense

a) **Exp 2 Run 1**


Context Bag
[Synset + Gloss + hypernyms]
Knowledge-Based Conceptual Overlap Count WSD Method
WordNet
Sense Bag
[Synset + Gloss + hypernyms]
+
[Hypernym of Words in Sense bag]
Disambiguated Sense

b) **Exp 2 Run 2**

Figure 4. 2: Setting of Experiment 2 (a) Exp 2 Run 1 and (b) Exp 2 Run 2


Context Bag
[Synset + Gloss + hypernyms + hyponyms, + meronyms]
Knowledge-Based Conceptual Overlap Count WSD Method
WordNet
Sense Bag
[Synset + Gloss + hypernyms + hyponyms + meronyms]
Disambiguated Sense
Context Bag
[Synset + Gloss + hypernyms]
Knowledge-Based Conceptual Overlap Count WSD Method
Sense Bag
[Synset + Gloss + hypernyms + hyponyms + meronyms] + [Hypernym of Words in Sense bag]
WordNet
Disambiguated Sense

a) **Exp 3 Run 1**

b) **Exp 3 Run 2**

Figure 4. 3: Setting of Experiment 3 (a) Exp 3 Run 1 and (b) Exp 3 Run 2

c) **Experiment 3: -** ***Exp 3 Run 1*** **and** ***Exp 3 Run 2***

Rationale: The rationale behind carrying out this experiment is that if hyponym's information and meronym's information in the sense bag and context bag are increased, this will further increase in the relatedness of correct sense with the context.

Intent: The intent of this experiment to represent the knowledge based contextual overlap count WSD method which utilizes information from synset, gloss, hypernym, hyponym and meronym of word in WordNet.

Experimental Setting: In **Exp 3 Run 1**, the synset, gloss, hypernyms, hyponyms and meronyms of words from WordNet are used to form sense bags and context bag. In **Exp 3 Run 2**, the same information as in **Exp 3 Run 1 is** utilized plus word's hypernym from WordNet are also added to each sense bags. The Figure 4.3 shows the experimental setting of Experiment 3.

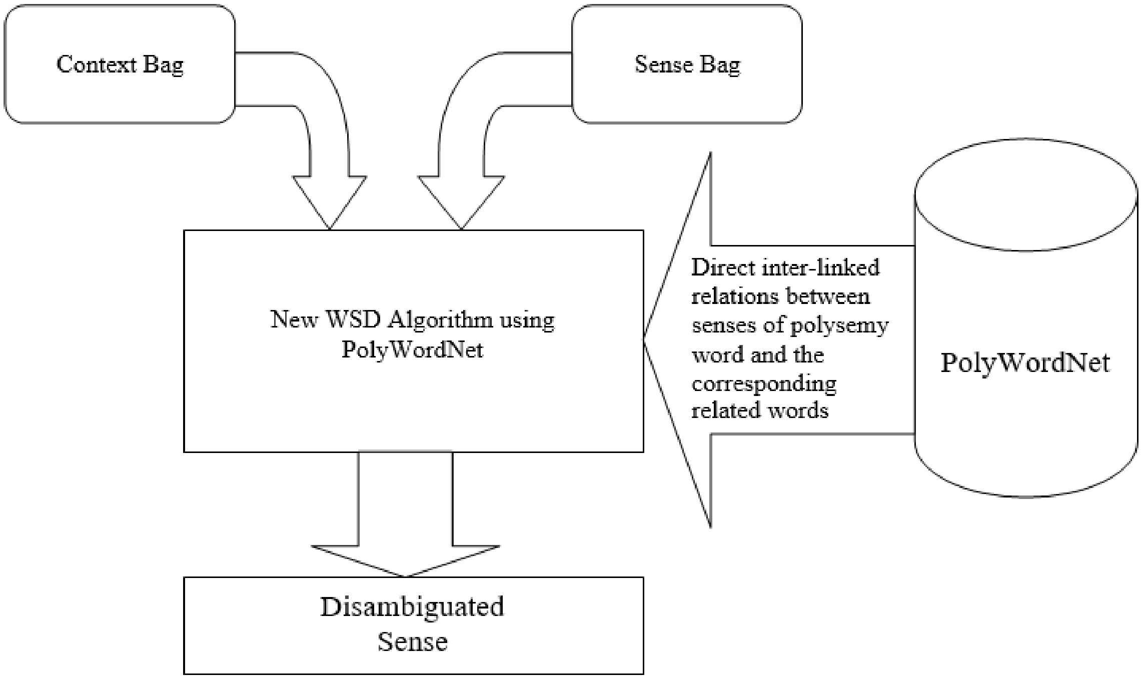

Context Bag
Sense Bag
New WSD Algorithm using PolyWordNet
Direct inter-linked relations between senses of polysemy word and the corresponding related words
PolyWordNet
Disambiguated Sense

Figure 4. 4: Setting of Experiment 4 (Exp 4)

d) Experiment 4: - *Exp 4*

Rationale: The interconnection of senses of polysemy words and their corresponding related words brings them together to form a cluster in PolyWordNet. These relations can be used to disambiguate the sense of polysemy words more accurately when a context is given.

Intent: The intent of this experiment is to show the interlinked relation of related words and their respective senses of polysemy word resolves noise information and to check whether the new WSD algorithm using PolyWordNet obtains higher accuracy for sense disambiguation as compared to using WordNet.

Experimental Setting: In Exp 4, the WordNet is replaced by new lexical database-PolyWordNet. The simplified Lesk algorithm is also replaced with new WSD algorithm that utilizes the relations from PolyWordNet for sense disambiguation. The Figure 4.4 shows the experimental setting of Experiment 4.

4.1.2 PolyWordNet Statistics

The current statistics of PolyWordNet shows only 3541 words. This number includes both polysemy words and single sense words. There are altogether 762 polysemy words with 1748 occurrences. The number of single sense words is 1793. The number of noun, verbs, adverbs and adjectives are 2264, 859, 51 and 367 respectively. In the other hand, the total number of words in English WordNet is 155287 [84]. Due to a high difference in the amount of current data that these two lexical databases contains, it does not give fair comparison, if the experiments are run in those databases directly in such situation. Therefore, using the same data in same amount as the PolyWordNet contains currently, a sample WordNet is built.

To compare these two lexical databases in the same environment and conditions, the amount of data and number of Test Sentences are kept constant during experiments in both cases when using the WordNet and PolyWordNet for word sense disambiguation. Since the data are taken from WordNet, it is already valid. This ensures that these both databases contain the same data in same amount. The only difference is the organization of the words in these two databases.

4.1.3 System Development

Based on the settings of experiments, a computer software is developed. In the software, there are the options to adjust 7 different experimental settings as they are described in experimental design. To develop the software, MS Visual Studio Express 2013 and MS SQL Server 2012 are used and the system is coded in C#. Python is also used to extract the test sentences from Brown corpus.

PolyWordNet Database

The database of PolyWordNet is designed based on the mathematical expressions described in subsection 3.6.6. The mathematical expressions state that the PolyWordNet is a collections of relations of senses of polysemy words and their corresponding related words. Therefore, to establish the relations between senses of polysemy words and related words, four tables- "noun", "verb", "adjective" and "adverb" are designed. These tables stored the relations of senses of polysemy words and related words. In addition, to keep the information and to keep the track of all relations of all words like glosses, parts_of_speach (pos), and four links- named as verb_id, noun_id, adverb_id and adjective_id, a table called "word_info" is designed. This table stores all the words whether they are polysemy words or related words. A word in “word_info” can have relation with other words which can be a noun or a verb or an adverb or an adjective. If the word has related words, which are nouns, then those relations with nouns are stored in the noun table. If the word has related words, which are verb, then those relations with verbs are stored in the verb table and so on. The word_info table contains words, their glosses, parts_of_speach (pos), and four links- named as verb_id, noun_id, adverb_id and adjective_id that linked the words in word_info with the four tables- verb, noun, adverb and adjective. The tables- verb, noun, adverb and adjective are used to store the links between the related words and corresponding senses of polysemy words. Each of these tables contains three fields. For example, the noun table contains, 1) noun_id which is same as the noun_id in word_info, 2) word_id which is the id of related word of the word with noun_id and this word_id is found in the word_info table and 3) relation which explain a relation between the linked words. However, in this research the third field- relation is not used. The database structure of lexical database PolyWordNet is shown in Figure 4.5.

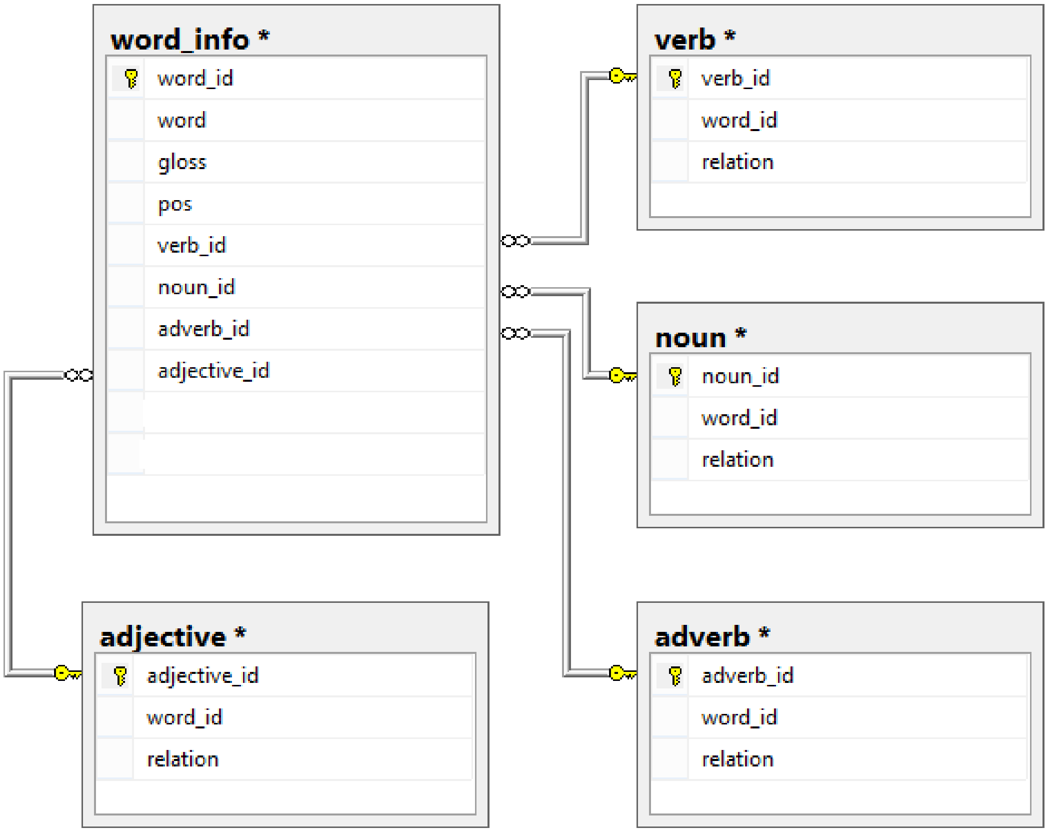


Figure 4. 5: Database structure of PolyWordNet

WSD Algorithm using PolyWordNet

The PolyWordNet is a collection of relations between senses of polysemy words and their respective related words. The links of these relations provided by PolyWordNet are used for sense disambiguation. The new WSD approach, which utilizes the relations from the PolyWordNet, utilizes a recursive function to search correct meaning of polysemy/target word starting from each context word available in a given context/sentence.

4.1.4 Data Collection

In this research, Method Triangulation is adopted for data collection, since more than one data generation/collection methods are used. The data required to build PolyWordNet and test sentences are collected by using different techniques and resources such as questionnaire, telephone interview and web materials. Web materials include the online WordNet 3.1 [85] and Oxford Lerner's Dictionary [86] and Cambridge Dictionary [87].

The online WordNet 3.1 and Oxford Lerner's Dictionary are used for only finding the polysemy words. The WordNet 2.1 is used collect the data required to build the PolyWordNet. The required Test Sentences are collected from Cambridge Dictionary and Brown Corpus.

Data Collection to Build PolyWordNet

To keep the data and its amount constant in both lexical databases- PolyWordNet and WordNet throughout the experiments, a set of data is taken from Princeton WordNet. This set of data is used to build the new lexical database: - PolyWordNet. To do this, 20 polysemy words are selected (See Appendix **1**) at first from respondents. These polysemy words are selected in three stages: - 1) researcher listed 10 polysemy words and 2) Remaining 10 polysemy words are collected in second stage by telephone interview with an English lecturer of Pokhara University. An English lecturer is selected from Pokhara University and asked whether he is comfortable to provide 10 polysemy words with the senses of which he is familiar with. The lecturer provided 10 polysemy words after a couple of days. In the third stage of data collection, 5 students studying at last year of Computer Engineering at Pokhara Engineering College, Pokhara University are selected. Each of these students collected 100 plus polysemy words and their related words. To build the PolyWordNet, first the senses of polysemy words and the glosses are uploaded in PolyWordNet. Then, the related words for each sense are collected from the gloss of each sense of polysemy words and test data. The synsets and glosses of the senses of polysemy words and related words are collected from WordNet 2.1. The related words are also uploaded in PolyWordNet. The senses of polysemy words are then linked with corresponding related words in PolyWordNet.

Data Collection to build Test Sentences

In first stage of collection of test sentences, 4 set of questionnaires are designed. Each set of questionnaire contained 5 polysemy words. Providing the questionnaires, respondents are requested to collect simple sentences for each sense listed in the questionnaire from sources like books and web resources. They are also asked to provide related words as many as they can do. Altogether, 280 test sentences are collected in this stage.

In second stage, the same way is used to collect test sentences from 5 students studying at last year of Computer Engineering at Pokhara Engineering College, Pokhara University. They collected 2725 test sentences from online dictionary- Cambridge Dictionary. The related words of each senses are also collected. A part of related words in PolyWordNet are shown in Appendix **2**. A sample of Test Sentences is shown in Appendix **3**.

In third stage of test sentences collection, 1200 test sentences are randomly collected form the ***news*** category of popular Brown Corpus. In the first 100 sentences, 24 sentences out of 100 are simple sentences. The remaining 76 sentences are compound sentences having one or more conjunction words and are very ambiguous. The most of the remaining test sentences out of 1200 are compound sentences which are highly ambiguous as well. Altogether, 4105 test sentences are collected and are used to test the developed system.

4.1.5 Results Analysis and Evaluation Tool

The results obtained from experiments are tabulated to facilitate the comparison analysis of data. ***The recall, precision*** and ***coverage*** metrics are adapted in this research for evaluation of results obtains from WSD algorithms. The WSD methods are commonly evaluated by using recall, precision and coverage metrics [88] [89] [90]. Since the WSD systems used in this research has ***one output per input***, the recall equals accuracy [91]. The recall (i.e. accuracy) of each experiment is calculated by dividing the number of correctly disambiguated sentences by the total number of sentences that are tested.

$$Recall = \left(\frac{No.of\ correctly\ disambiguated\ sentences}{Total\ No.of\ test\ sentences\ that\ are\ tested} \right) \qquad \textbf{(4.1)}$$

Precision provides the information about how many sentences are correctly disambiguated out of the number of test sentences for which the WSD algorithms made a prediction. It is calculated as:

$$Precision = \left(\frac{No.of\ correctly\ disambiguated\ sentences}{Total\ No.of\ test\ sentences\ for\ which\ WSD\ algorithm\ made\ a\ prediction}\right) \qquad \textbf{(4.2)}$$

There may be the cases in which the WSD algorithms cannot predict any result. That is the WSD algorithm may not produce any answer to the input. This situation can be expressed by using coverage to the test data. It is defined as:

$$Coverage = \left(\frac{Total\ No.of\ test\ sentences\ for\ which\ WSD\ algorithm\ made\ a\ prediction}{Total\ No.of\ test\ sentences}\right) \qquad \textbf{(4.3)}$$

Finally, the recall, precision and coverage observed on the experiments are presented in line diagrams. These results are then analyzed and compared to draw conclusions.

4.2 PARTICIPANTS OF THE RESEARCH

The participants of this research include 1) the researchers themselves and 2) the respondents who are interviewed and asked to answer questionnaires to collect the data required for this research.

4.2.1 Role of researcher

The researcher prepared the questionnaires, selected the respondents and distributed the questionnaires to the respondents. After the respondents returned the questionnaires, researcher sent the answered questionnaires to English lecturer to check the data collected in questionnaires are correct semantically and grammatically.

This respondent is responsible to edit the data obtained in questionnaires if any errors (syntactic or semantic) are found. After this, researcher uploaded the data to build the lexical databases and test data.

4.2.2 Respondents

In this research, the data are collected in three stages. In first stage of data collection, an English lecturer is selected from the Central College of Pokhara University to find out 10 polysemy words. This respondent is also responsible to check and edit the data obtained in questionnaires if any errors (syntactic or semantic) are found. In second stage data collection, 15 graduate students are randomly chosen. The 10 out of 15 were selected from Pulchowk Campus, IoE, Tribhuvan University, Nepal and remaining 5 are selected from Darmstadt University of Applied Sciences, Darmstadt, Germany. Firstly, 2 from Pulchowk Campus and 1 from Darmstadt University are chosen by researcher and each are requested to choose 5 others including themselves. These 3 were responsible for distributing the provided questionnaires, collecting and returning back to the researcher. These respondents are contacted via telephone and email. The questionnaires are distributed and finally collected via email. In third stage, In the third stage of data collection, 5 students studying at last year of Computer Engineering at Pokhara Engineering College, Pokhara University. Each of these students collected 100 plus polysemy words and their related words. They also collected 2725 test sentences.

The new lexical database PolyWordNet and a new WSD algorithm which uses PolyWordNet for sense disambiguation are developed. Then, the questionnaires prepared and the required data are collected. After this, a sample WordNet and PolyWordNet are built. Then, the experiments are run and the results are collected for analysis.

4.3 VALIDATION OF POLYWORDNET, DATA AND TEST DATA

The validation of the collected data, validation of developed PolyWordNet and the validation of test data/sentences are done by using the WSD system as a validation tool.

4.3.1 Validation of PolyWordNet

BalkaNet is a European project which was developing WordNets for 5 Balkan languages [92]. These 5 Balkan languages includes: - Greek, Serbian, Bulgarian, Turkish and Romanian. To validate the quality assurance of BalkaNet, Tufins et. al. (2004) developed a language independent word sense disambiguation WSD tool to validate the new lexical database BalkaNet. They have used WSD approach as a validation tool and calculated the accuracy using BalkaNet and this accuracy is compared with the accuracy obtained with the Princeton WordNet to check the alignment of BalkaNet with Princeton WordNet. Following the similar concept used to validate the BalkaNet, we have also used WSD methods as a tool to validate our new lexical database- PolyWordNet. We have used popular knowledge-based overlap count WSD algorithm for word sense disambiguation using information from WordNet.

4.3.2 Validation of Data that is used to build PolyWordNet and Test Data

The data that are used to build the PolyWordNet and the Test Data used in this research to test the experiments are validated using the WSD system as a validation tool. For this, we have again used well known popular simplified Lesk algorithm and WordNet as benchmark tool to validate the data and test data. We have run 30 different experiments (that uses the simplified Lesk algorithm and WordNet for word sense disambiguation) on the our data (altogether 3541 words and their information and relations like their gloss, hypernyms, hyponyms and meronyms) and tested these experiments by using our test sentences containing altogether 2905 test sentences. We noted and analyzed the accuracies obtained in these 30 different experiments to check whether the accuracy range obtained in these experiments are aligned the accuracy range of various Lesk algorithms that uses the WordNet and tested under standard evaluation exercises like SenseVal [12] [13] [14] [15] [69] [93] [94].

4.3.3 Reference Accuracy Range

A reference accuracy range is defined to check whether the accuracy range obtained in this research will align with the other WSD algorithms that uses the relations from WordNet and evaluated with standard WSD evaluation exercises like Senseval, SemCore and Brown corpus. To define the reference accuracy range, the WSD methods proposed in [13], [14] [15], [93], [94], [95] and [96] are chosen. The reasons behind to choose these WSD methods are that they all uses the information from WordNet and they are evaluated by using the standard WSD evaluation exercise like Senseval, SemCore and Brown corpus. The purpose of defining this reference accuracy range is to find out an overall picture of accuracies that are observed in

WSD algorithms which uses the information from WordNet. This accuracy range then can be used to check whether the accuracy range of the experiments of this research will align with this range. If these accuracy ranges are aligned, the data used to build the PolyWordNet and the Test Sentences used to evaluate the experiments in this research can be then accepted as valid.

Before finding the reference accuracy range, all these WSD methods chosen to define the reference accuracy range are described in brief. In 2002, Banerjee and Pedersen adjusted the original Lesk algorithm by taking advantage of more information provided by the lexical database WordNet [13]. The results of the experiments on their system when tested by **Senseval-2** showed an overall accuracy of 32%. Altogether 73 polysemy words are tested with a total of 4328 instances. Seo et al. developed an unsupervised WSD method. They utilized a set of relations- synonyms, hyponyms and hypernyms from WordNet [15]. The results from their experiments when tested by **SemCor** shows the accuracy range (48.39 to 52.34) % and when tested by **Senseval-2** show the accuracy range (42.24-45.48) %.

Table 4. 1. Accuracies of various WSD methods which uses information from WordNet and are evaluated using standard WSD evaluation exercises

SN	Method used	Accuracy/%)
1	Adapted Lesk Algorithm (Banerjee & Pedersen, **2002**)	32
2	Unsupervised WSD using WordNet relatives (Seo, et al., **2004**)	42.24 – 52.34
3	WSD using WordNet relations (Fragos, et al., 2003)	52.5 and 66.2
4	WSD using cosine similarity collaborates… (Orkphol, **2019**)	48.7 and 50.9
5	WordNet based algorithm for WSD (Li, **1995**)	57
6	An Optimized Lesk-Based Algo. for WSD (Ayetiran & Agbele, **2018**)	37.7 and 65.7
7	WSD: A comprehensive knowledge exploitation…(Wang, et al., **2020**)	66.1 and 69.6

Fragos proposed the "Weighted Overlapping" disambiguation method [14]. They used the definitions of sense of word, synset definitions and hypernymy relations to form both sense bag and context bag for disambiguation. They evaluated their algorithm using **Brown corpus** data and observed 52.5% accuracy. With the use of heuristic, they attained the accuracy of 66.2%. Orkphol and Yang used sense definition and relations except antonyms retrieved from WordNet in their approach which uses both context sentence vector and sense definition vectors constructing from Word2vec to provide each word a score using cosine similarity [94]. The valuated their experiments by using **Senseval-3** and the result shows the accuracy of 50.9%. It is 48.7% when calculated without considering sense distribution probability.

Li et al. developed an algorithm for word sense disambiguation based on lexical knowledge contained in WordNet [93]. The tested experiments using the **Canadian Income Tax Guide**. The evaluation is done on a test of 397 cases and shows the accuracy of 57%. Ayetiran and Agbele developed an optimized Lesk-based using the knowledge recourse including WordNet [95]. They evaluated their system using **Senseval**. They found the recall to be 0.377 for simplified Lesk algorithm and recall to be 0.657 for optimized Lesk algorithm. Wang et al. proposed a WSD method which uses the knowledge base- WordNet. They have used semantic space and semantic path hidden behind a sentence [96]. When they tested their system with the **Senseval-2**, it shows accuracy of 69.6%. It shows accuracy of 66.1% when it is tested with **Senseval-3**. The Table 4.1 shows the list of various WSD methods which uses information from WordNet and are evaluated using standard WSD evaluation exercises along with their accuracies.

The accuracy of these WSD methods falls in the range from 30% to 70%. Therefore the reference accuracy range is defined as a range (30-70) %. This accuracy range is used to check whether the accuracy range of the experiments of this research will align with this range.

The accuracies of the following WSD methods also align with the defined reference accuracy range. Therefore these WSD methods also support for the defined reference accuracy range. Sharma and Joshi used knowledge based Lesk algorithm to disambiguate Hindi language using Hindi WordNet [97]. They used 3000 ambiguous sentences to test their system and found the accuracy of 71.43%. Roy et al. proposed an overlap-based WSD algorithm for Nepali word sense disambiguation using Nepali WordNet built at Assam University [63]. They tested their experiment for on 1663 words with 912 nouns and 751 adjectives. They found accuracy of 54% for nouns and 42% for adjectives for their overlap-based approach. Bhingardive and Bhattacharyya used unsupervised WSD methods using IndoWordNet for word sense disambiguation of Indian languages [98]. They found the accuracy (recall) in range 38.13% to 61.58%. Kumar et al. used Adapted Lesk Algorithm based Word Sense Disambiguation using the Context Information [99]. They tested their system on 50 polysemy words with 2000 sentences. They got the recall to be in range 0.23 to 0.665 in different variation of Lesk algorithm. Their system shows the recall of 0.665.

4.3.4 Validation Method

The underlying principle to validate the new lexical database is to use a sample amount of data from WordNet and a set of Test sentences. Using a simplified Lesk algorithm, a set of experiments are run for word sense disambiguation using that data sample and tested by the collected set of Test Sentences. The obtained accuracy range is then compared with a reference accuracy range of various already tested and approved WSD methods that uses the WordNet for word sense disambiguation. If the accuracy range obtained in this research is at least aligned with or greater than that of the reference accuracy range, then the data taken from WordNet and the set of Test Sentences are valid. Using this valid data and Test Sentences, if the WSD using PolyWordNet also shows the accuracy range that at least aligns with or greater than that of the reference range of accuracy, then the PolyWordNet is also valid. The detail validation method is as follows:

1. A fix amount of data are taken from WordNet. The amount of data taken is 3541 words along with their synset, gloss, hypernym, hyponyms and meronyms. A simplified Lesk algorithm is used for word sense disambiguation using only these data with 3541 words including their synset, gloss, hypernym, hyponyms and meronyms. As described in subsection 4.1.1, six experiments with different settings (***Exp 1 Run 1,*** 2) ***Exp 1 Run 2,*** 3) ***Exp 2 Run 1,*** 4) ***Exp 2 Run 2,*** 5) ***Exp 3 Run 1,*** 6) ***Exp 3 Run***) are designed. All these experiments use simplified Lesk algorithm and WordNet for sense disambiguation. The difference in these experiments is only the amount of data they

used in context and sense bags. As going from ***Exp 1 Run 1*** to ***Exp 3 Run 2,*** the amount of data in context and sense bags are increased in context and sense bags. The simplified Lesk algorithm and all experimental settings are described respectively in sections – 3.1.1 and 4.1.1 in detail.

2. A fix number of Test Sentences is collected. The 2905 Test Sentences are collected from respondents and 1200 Test Sentences are collected from ***news*** category of Brown Corpus. These Test Sentences are used to test the accuracies (i.e. recall including precision and coverage).
3. The accuracies obtained in these experiments- ***Exp 1 Run 1,*** 2) ***Exp 1 Run 2,*** 3) ***Exp 2 Run 1,*** 4) ***Exp 2 Run 2,*** 5) ***Exp 3 Run 1,*** 6) ***Exp 3 Run*** are compared with the ***reference accuracy range*** (30-70) %.
4. If the accuracy range of experiments- ***Exp 1 Run 1,*** 2) ***Exp 1 Run 2,*** 3) ***Exp 2 Run 1,*** 4) ***Exp 2 Run 2,*** 5) ***Exp 3 Run 1,*** 6) ***Exp 3 Run*** is aligned with the reference accuracy range (30-70) %, then the data (3541 words along with their synset, gloss, hypernym, hyponyms and meronyms) and the Test Sentences (2905 + 1200) are valid.
5. The PolyWordNet is built from the validated same data in same amount (3541 words) in which experiments- ***Exp 1 Run 1,*** 2) ***Exp 1 Run 2,*** 3) ***Exp 2 Run 1,*** 4) ***Exp 2 Run 2,*** 5) ***Exp 3 Run 1,*** 6) ***Exp 3 Run*** are run and tested. Similarly, the WSD algorithm in experiments- ***Exp 4*** using PolyWordNet are tested using the same Test Sentences (2905 + 1200) which are already validated. In this way, the PolyWordNet built from the validated data is also valid. In addition, If the accuracy range of new WSD method using PolyWordNet is at least aligned or greater than the reference accuracy range (30% to 70%), the new lexical database- PolyWordNet is valid. The WSD algorithm using PolyWordNet and the experimental setting of the experiment- ***Exp 4*** are described respectively in subsections- 3.6.5 and 4.1.1 in detail.

Chapter 5

Result, Analysis and Comparison

In chapter 4, the methodology adopted in this research is presented in detail. For instance, the experiment settings, the data collection methods and validation methods are discussed in detail. In this chapter, the results obtained in six series of experiments are presented, analyzed and finally findings are presented.

5.1 EXPERIMENT SERIES

To test whether the word's organization of PolyWordNet is acceptable for word sense disambiguation with reference to other lexical resources like dictionaries and WordNet, 7 different experiments are set up. These 7 experiments with distinct set up are repeated for 9 times dividing into 6 series of experiment runs. Thus, altogether experiments have 63 runs. The detail of these experiments are explained in Chapter 4. The experiment runs are mainly divided into 6 series named as Series A, B, C, D, E and F. The Series A to E experiments are run by increasing the data in PolyWordNet in each next series. In addition, these runs are also tested by increasing the number of test sentences in each next series. The series A to series E experiments are tested by altogether 2905 test sentences collected in this research. The total amount of data in the Series E experiments is 3541 words. All experiments in Series F are tested by using the 1200 test sentences randomly taken from Brown Corpus. The data in PolyWordNet is 3541 words during all experiments in Series F experiments. In this series, the system is tested by increasing the number of test sentences from Brown Corpus.

The first 6 experiments (*Exp 1 Run 1, Exp 1 Run 2, Exp 2 Run 1, Exp 2 Run 2, Exp 3 Run 1 and Exp 3 Run 2*) out of 7 have used simplified Lesk algorithm and have utilized information from WordNet for sense disambiguation. The difference in these 6 experiments is the amount of information utilized for word sense disambiguation. As moving from experiment 1 to 6, the amount of information is increased in every next experiment. The seventh experiment (i.e. ***Exp 4)*** uses the new WSD algorithm and PolyWordNet for sense disambiguation. Actually the first 6 experiments are 3 experiments each of which having 2 variations. This is described in detail in Chapter 4. The results obtained in all these 63 runs of are presented in the following sections.

5.2 SERIES A EXPERIMENTS

The **Series A Experiments** contains single of each of the 7 different experiments. These experiments are run at the system using PolyWordNet and sample WordNet which contain only 280 words and are tested by 180 test sentences. The statistics of the words used in Series A Experiments are shown in Table 5.1. It shows that there are 56 occurrences of polysemy words. The nouns are 213, verbs 56, adverb 1 and adjectives 10.

Table 5. 1: Series A Experiments and Word Statistics in PolyWordNet

Total Number of Words	Polysemy Words	Single Sense Words	Nouns	Verbs	Adverbs	Adjectives
280	56	224	213	56	1	10

Table 5. 2: Results obtained in Series A Experiments

	Exp 1 Run 1	Exp 1 Run 2	Exp 2 Run 1	Exp 2 Run 2	Exp 3 Run 1	Exp 3 Run 2	Exp 4
No of C.D.S.*	77	78	96	64	105	77	173
Accuracy (%) = Recall	42.78	43.33	53.33	35.56	58.33	42.78	96.11

**C.D.S = Correctly Disambiguated Sentences*

The Figure 5.2 shows the number of correctly disambiguated sentences out of the 180 test sentences and observed accuracies in 7 experiments. The detail of result of each experiments in **Series A Experiments** are described in the following subsections.

5.2.1 Results of Experiment 1- *Exp 1 Run 1* and *Exp 1 Run 2*

In ***Exp 1 Run 1***, 77 out of 180 test sentences are disambiguated correctly with accuracy of 42.78%. The total number of correctly disambiguated test sentences is found to be increased by 1 in ***Exp 1 Run 2*** with accuracy of 43.33%. An important point observed is that the test sentences that are correctly disambiguated in ***Exp 1 Run 1*** are not correctly disambiguated in ***Exp 1 Run 2***.

These results indicates with the increase of information in sense and context bags can contribute to increase the number of overlaps between the context and the correct sense of polysemy words resulting in better accuracy.

5.2.2 Results of Experiment 2- *Exp 2 Run 1* and *Exp 2 Run 2*

Strange results are observed in two variation of experiment 2. The accuracy of ***Exp 2 Run 1*** is found to be increased by a large amount of 10% as compared with the accuracy obtained in ***Exp 1 Run 2***. Out of 180 test sentences, 96 sentences are disambiguated correctly in ***Exp 2 Run 1*** with accuracy of 53.33%. Another strange result is found in ***Exp 2 Run 2***. The accuracy of ***Exp 2 Run 2*** is found to be decreased by a large percentage of 17.77% as compare with the accuracy of ***Exp 2 Run 1.*** Out of 180 sentences, only 64 sentences are disambiguated correctly with accuracy of 35.56%.

In ***Exp 2 Run 1,*** the accuracy of the system is increased with the increase in formation in sense and context bags by a large percentage of 10% as compared to ***Exp 1 Run 2***. In ***Exp 2 Run 2***, with the increase in information in sense bag decreased the accuracy again by a large percentage of 17.77% as compared with ***Exp 2 Run 1***. This indicates that only increasing more information in sense or context bag does not always increases the accuracy. The reason is the induction of noise information in sense and context bags. The noise information causes the higher overlaps for context with the wrong sense of polysemy words. The reason behind the noise information is the more common hypernyms of the words in context and the wrong sense of polysemy word.

5.2.3 Results of Experiment 3- *Exp 3 Run 1* and *Exp 3 Run 2*

Among the first 6 experiments which use WordNet for sense disambiguation, the highest accuracy is observed in ***Exp 3 Run 1***. The 105 sentences are disambiguated correctly with the accuracy of 58.33%. However, with the inclusion of information from hypernym of words in sense bag, the number of sentences correctly disambiguated is found to be greatly decreased to 77 with accuracy of 42.78% in ***Exp 3 Run 2***.

The accuracy of ***Exp 3 Run 2*** is found to be same as the accuracy of ***Exp 1 Run 1*** which uses only the word's gloss and synset for sense disambiguation. Even after including a huge amount of information from WordNet in sense and context bags in ***Exp 2 Run 2***, it has the same accuracy. This indicates that the inclusion of more information in sense and context bag does not always increase the relatedness between the context and correct sense.

5.2.4 Results of Experiment 4- *Exp 4*

The ***Exp 4*** uses the relations from PolyWordNet for sense disambiguation. The result of ***Exp 4*** shows a better result as compared with the previous first 6 experiments. Out of 180 sentences, 173 sentences are disambiguated correctly in ***Exp 4*** with accuracy of 96.11%.

With the same amount of data in the system, when it is tested by the same test sentences, is found to be more accurate when it uses information from PolyWordNet. This indicates the organizations of words in PolyWordNet is better especially for word sense disambiguation. The reason behind this is that the PolyWordNet deals with the relation between the related words and sense of polysemy words and connects those words. This results the cluster of sense of a polysemy word and its related words which are sufficient to disambiguate its sense in context.

5.3 SERIES B TO SERIES E EXPERIMENTS

The intend to run Series B to Series E Experiments is to observe the effect on results of the 7 experiments on increasing the number of data in lexical databases- PolyWordNet and sample

WordNet. The Series B Experiments are run on 290 words and tested by 180 test sentences. Series C Experiments are run on 1477 words and tested by 930 test sentences. Series D Experiments are run on 2501 words and tested by 1930 test sentences.

Table 5. 3: Number of test sentences that are correctly disambiguated in Series B to Series E Experiments

Exp Series	No of Words	No of TS	Exp 1 Run 1	Exp 1 Run 2	Exp 2 Run 1	Exp 2 Run 2	Exp 3 Run 1	Exp 3 Run 1	Exp 4
Series B	290	180	83	100	88	94	92	108	177
Series C	1477	930	387	490	493	460	506	494	918
Series D	2501	1930	876	1104	998	1026	1019	1098	1914
Series E	3541	2905	1307	1680	1494	1521	1491	1657	2883

Table 5. 4: *Recall, Precision* and *Coverage* of Series B to E Experiments

Exp Series	Evaluation Metrics	Exp 1 Run 1	Exp 1 Run 2	Exp 2 Run 1	Exp 2 Run 2	Exp 3 Run 1	Exp 3 Run 2	Exp 4
Series B	Recall	0.461	0.556	0.489	0.522	0.511	0.600	0.983
	Precision	0.716	0.617	0.492	0.528	0.511	0.603	0.994
	Coverage	0.644	0.900	0.994	0.989	1.000	0.994	0.989
Series C	Recall	0.416	0.527	0.530	0.495	0.544	0.531	0.987
	Precision	0.681	0.590	0.548	0.501	0.559	0.535	0.994
	Coverage	0.611	0.892	0.967	0.988	0.973	0.994	0.994
Series D	Recall	0.454	0.572	0.517	0.532	0.528	0.569	0.992
	Precision	0.688	0.622	0.529	0.534	0.534	0.570	0.994
	Coverage	0.660	0.919	0.977	0.996	0.989	0.997	0.998
Series E	Recall	0.450	0.578	0.514	0.524	0.513	0.570	0.992
	Precision	0.672	0.619	0.523	0.525	0.517	0.571	0.994
	Coverage	0.670	0.935	0.984	0.997	0.992	0.998	0.999

Series E Experiments are run on 3541 words and tested by 2905 test sentences. In each series of experiments, the number of Test Sentences (TS) in is also increased except for Series B Experiments. The intent to increase the number of TS in each series of experiments is to observe the effects on results when the experiments are tested with the new set of larger number of test sentences.

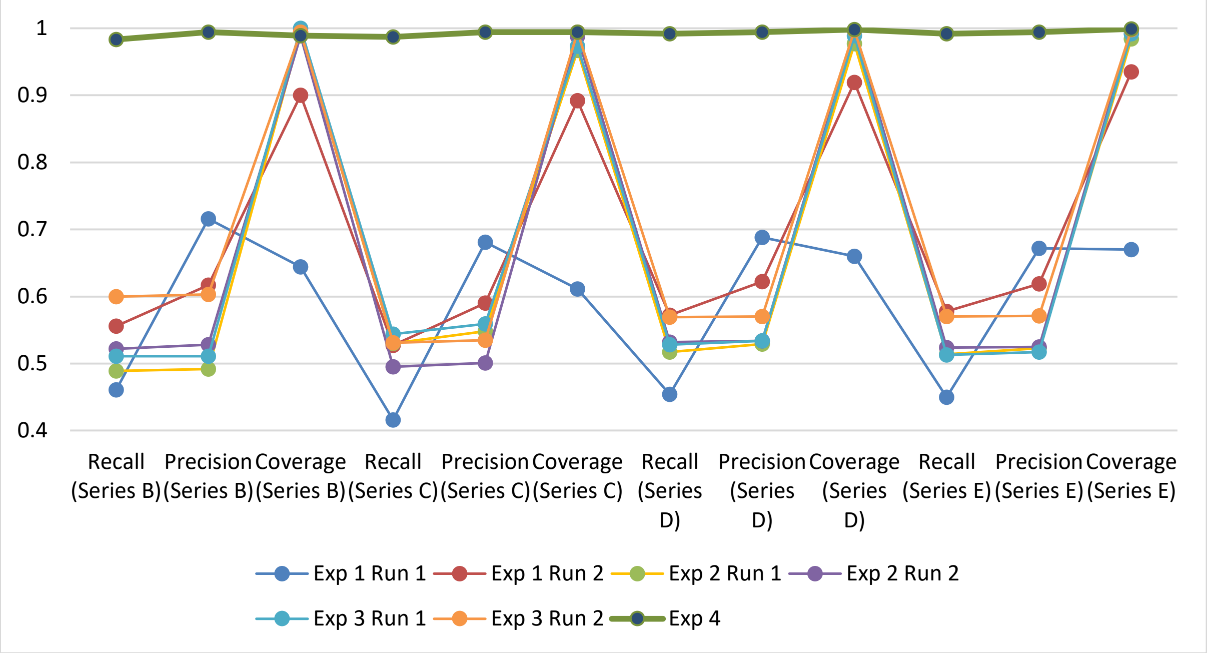


Figure 5. 1: *Recall, Precision* and *Coverage* of Series B to E Experiments

The Table 5.3 show the number of the test sentences that are correctly disambiguated in Series B to Series E Experiments. The Table 5.4 shows observed recall, precision and coverage of each run of the experiments in Series B to Series E.

The obtained result shows the maximum recall of WSD method using WordNet is 0.60 which is observed in ***Exp 3 Run 2*** of Series B experiments with the precision of 0.603 and coverage of 0.994. The minimum recall obtained for the WSD method using WordNet is 0.416 which is observed in ***Exp 1 Run 1*** of Series C experiments with the precision of 0.681 and coverage of 0.611. The average recall of these experiments is 0.52. The maximum and minimum recall of the WSD method using PolyWordNet are found to be 0.992 (with precision of 0.994 and coverage of 0.998 in Series D and precision of 0.994 and coverage of 0.999 in Series E), 0.983 (with precision of precision of 0.994 and coverage of 0.989). These are shown in Figure 5.1, 5.2, 5.3 and 5.4. The average recall of WSD method using PolyWordNet is found to be 0.989. The obtained results shows that the recalls of WSD method using the PolyWordNet are observed better as compared to the WSD that are using WordNet. This indicates the word organization of PolyWordNet better suits for sense disambiguation.

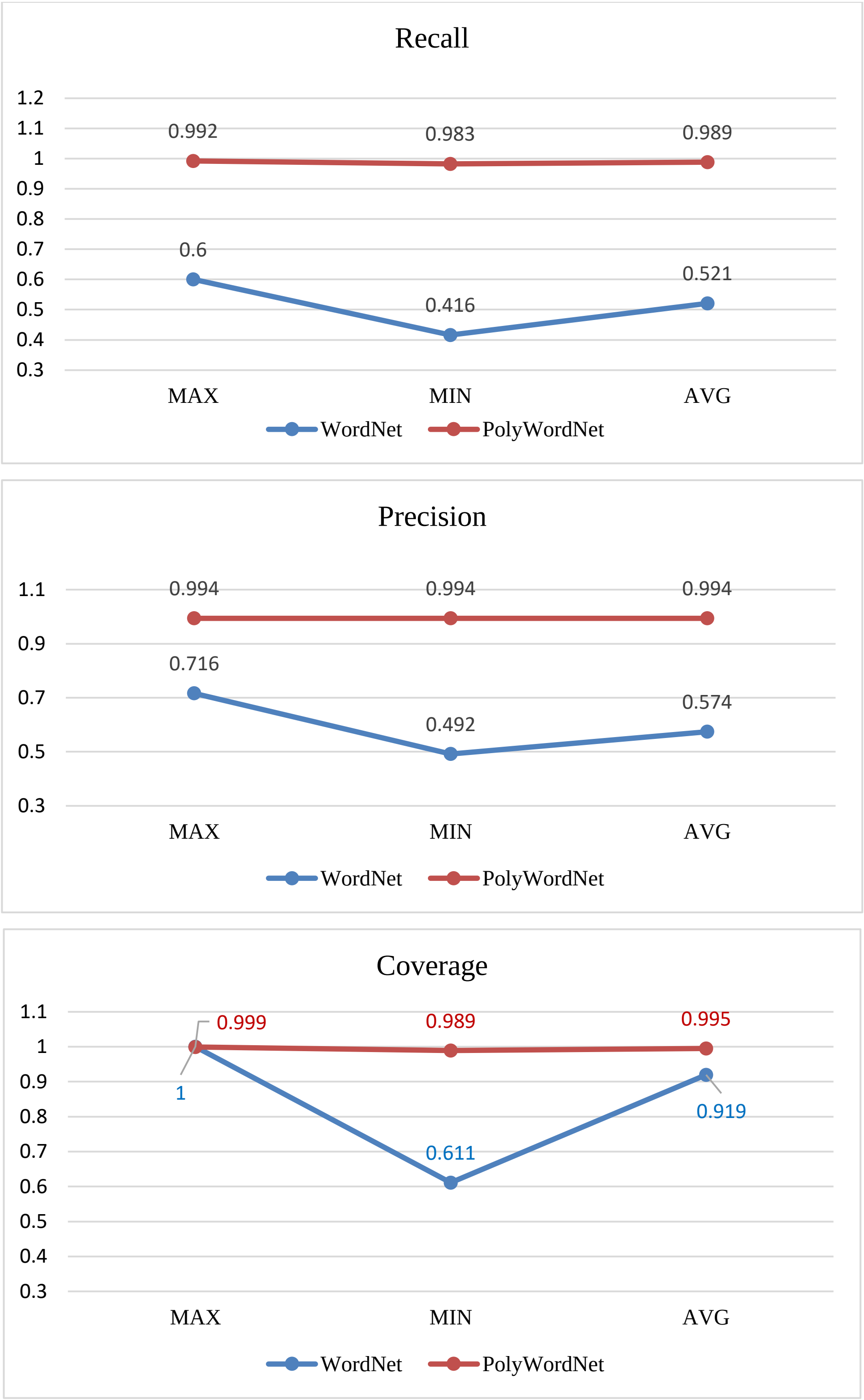


Figure 5. 2: Average *Recall, Precision* and *Coverage* of Series B to E Experiments

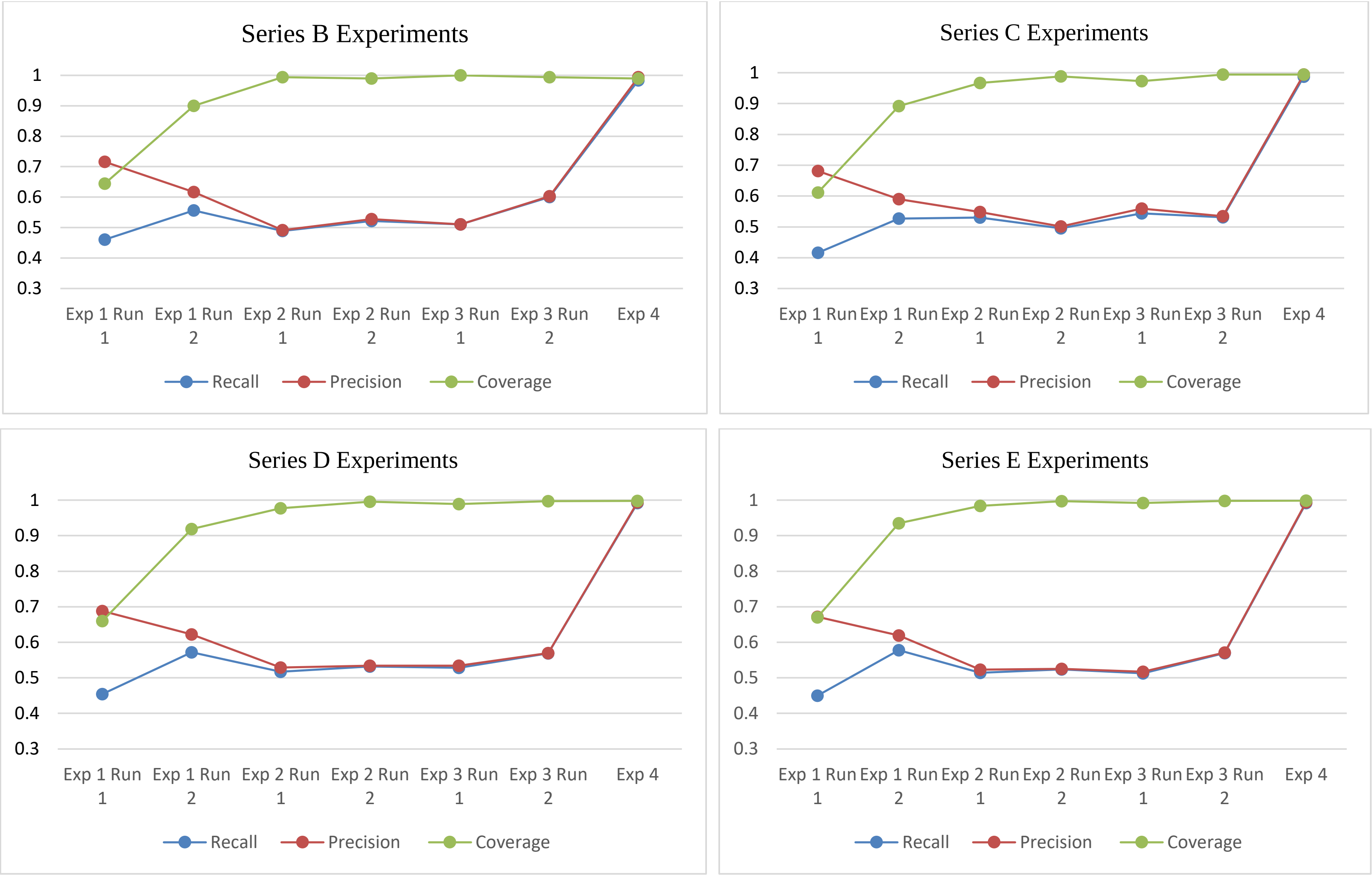


Figure 5. 3: Series-wise comparison of *Recall, Precision* and *Coverage* of Series B to E Experiments

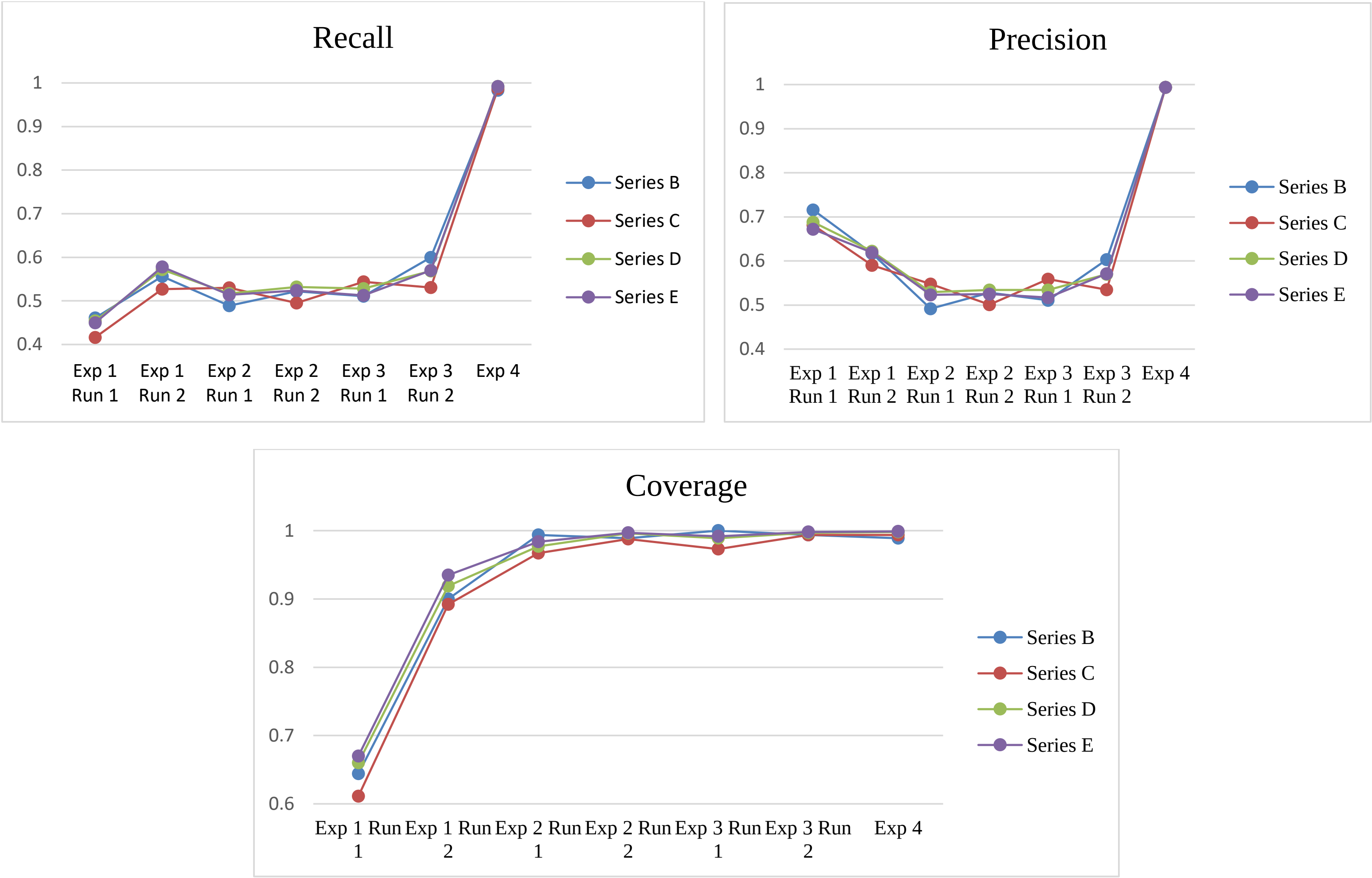


Figure 5. 4: *Recall, Precision* and *Coverage* comparison of Series B to E Experiments (*Recall, Precision* and *Coverage-wise*)

The result of WSD methods using WordNet shows fall and rise in recall and precision with the increase of information in sense and context bags and with the increase in data in PolyWordNet and sample WordNet. This indicates the increase in information in sense and context bags does not ensure increase in recall and precision.

The results show the maximum precision observed for the WSD method using WordNet is 0.716 in ***Exp 1 Run 1*** of Series B experiment but the recall obtained here is 0.461. The WSD method using WordNet has maximum coverage of 1 and average coverage is 0.919. The average coverage of WSD using PolyWordNet is better than that of the WSD method using WordNet.

The total number of words used in the Series E Experiments are 3,541 containing 762 polysemy words with 1748 occurrences. The detail word statistics of PolyWordNet in the Series E Experiments is shown in Table 5.5.

Table 5. 5: Final Word Statistics in PolyWordNet and Sample WordNet

Total Number of Words	Polysemy Words	Single Sense Words	Nouns	Verbs	Adverbs	Adjectives
3541	1748	1793	2264	859	51	367

5.4 SERIES F EXPERIMENTS

The experiments in Series A to E are tested with the test sentences collected in this research. They show exceptionally higher recalls for the WSD method that uses the PolyWordNet for sense disambiguation with very high precision and coverage. However, the experiments that uses WordNet has the accuracy range of 41.6% to 60% which is quite normal and represents the accuracies obtained in other similar works [12] [13] [15] [69] [14] [93] [94]. This evidence indicates that both the data used in PolyWordNet (as it is same as data used in sample WordNet) and test sentences used to test these experiments are both valid. The reason for the exception accuracy observed in WSD methods that uses PolyWordNet is due to its high coverage since the PolyWordNet contains almost all related words for the polysemy words stored in PolyWordNet. Therefore, to get rid of this doubt arose, 1200 test sentences are randomly collected from ***news*** category of popular Brown Corpus. These test sentences collected from Brown Corpus are then used to test the experiments. The amount of data in PolyWordNet and sample WordNet throughout the Series F experiments are same (i.e. 3541 words) as in Series E experiments.

The intent of Series F experiments is to check the recall, precision and coverage of the 7 different experiments when tested with the 1200 test sentences collected from popular **Brown corpus**. The 7 different experiments are repeated 4 times and tested with different numbers of these test sentences. The experiments are tested by 100 sentences in first, then by 400 sentences in second repetition, 800 hundred sentences in third repetition and finally by 1200 sentences in fourth repetition. The Table 5.6 show the recall, precision and coverage of Series F experiments.

Table 5. 6: Recall, Precision and Coverage observed in Series F experiments

No of Test Sentences	Measure Metrics	**Exp 1 Run 1**	**Exp 1 Run 2**	**Exp 2 Run 1**	**Exp 2 Run 2**	**Exp 3 Run 1**	**Exp 3 Run 2**	**Exp 4**
100	Recall	0.36	0.38	0.46	0.38	0.5	0.38	0.62
	Precision	0.48	0.391	0.474	0.391	0.515	0.387	0.873
	Coverage	0.75	0.97	0.97	0.97	0.97	0.98	0.71
400	Recall	0.357	0.31	0.375	0.297	0.415	0.332	0.682
	Precision	0.481	0.32	0.385	0.303	0.423	0.336	0.925
	Coverage	0.74	0.967	0.972	0.98	0.98	0.987	0.737
800	Recall	0.407	0.455	0.427	0.412	0.463	0.45	0.673
	Precision	0.529	0.468	0.436	0.418	0.47	0.453	0.93
	Coverage	0.77	0.971	0.98	0.985	0.986	0.991	0.723
1200	Recall	0.345	0.368	0.395	0.328	0.418	0.366	0.655
	Precision	0.426	0.377	0.402	0.333	0.423	0.369	0.911
	Coverage	0.81	0.975	0.981	0.987	0.986	0.991	0.719

As moving horizontal from ***Exp 1 Run 1*** to ***Exp 3 Run 2*** the information in sense and context bag is increased. As moving vertical down, the number of test data is increased. The result of the Table 5.6 shows the similar pattern of recall and precision of both WSD using WordNet and PolyWordNet as in the results obtained in Series A to E experiments. However, the coverage of

WSD using PolyWordNet is greatly decreased in F series experiments as compared to Series A to E experiments. The maximum coverage of WSD using PolyWordNet in Series F experiments is 0.737 while it is 0.991 for WSD using WordNet. The minimum coverage found for WSD using WordNet is 0.74 which is greater than the maximum coverage obtained by WSD method using PolyWordNet. The WSD method using PolyWordNet even in the low coverage has greater recall and precision. The minimum recall of WSD method using PolyWordNet is 0.620 which is even greater that the maximum recall (0.50) obtained by WSD method using WordNet. The WSD method using PolyWordNet has exceptionally higher precision as compared with the ASD method using WordNet. The average precision of WSD method using PolyWordNet is 0.910 which is more than double of the average precision (0.416) of WSD method using WordNet. The comparison of the result obtained in Series F experiments are shown in Figure 5.5, Figure 5.6, Figure 5.7 and Figure 5.8. The reason behind the decreased coverage of WSD method in Series F experiments is that the test sentences are taken from the Brown Corpus and no more related words are added to the PolyWordNet. Even in this case, WSD using PolyWordNet has a better recall with high precision as compared to WSD method using WordNet. This indicates that the word organization of PolyWordNet better suits for word sense disambiguation.

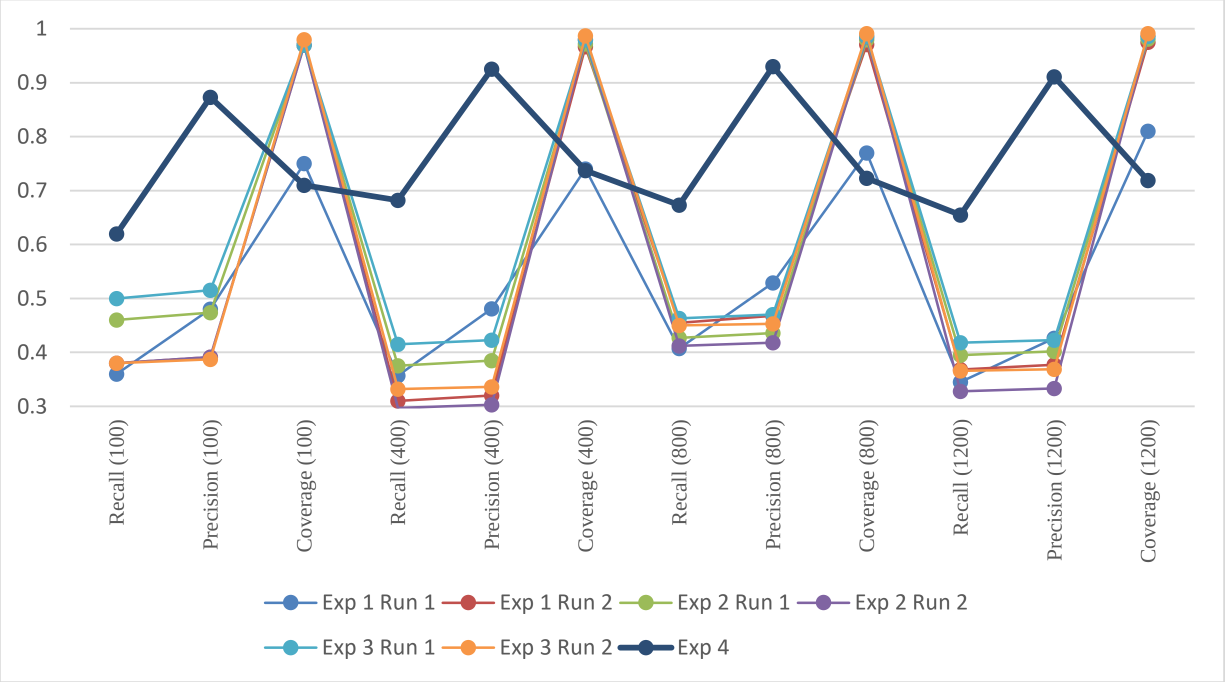


Figure 5. 5: *Recall, Precision* and *Coverage* obtained in Series F Experiments

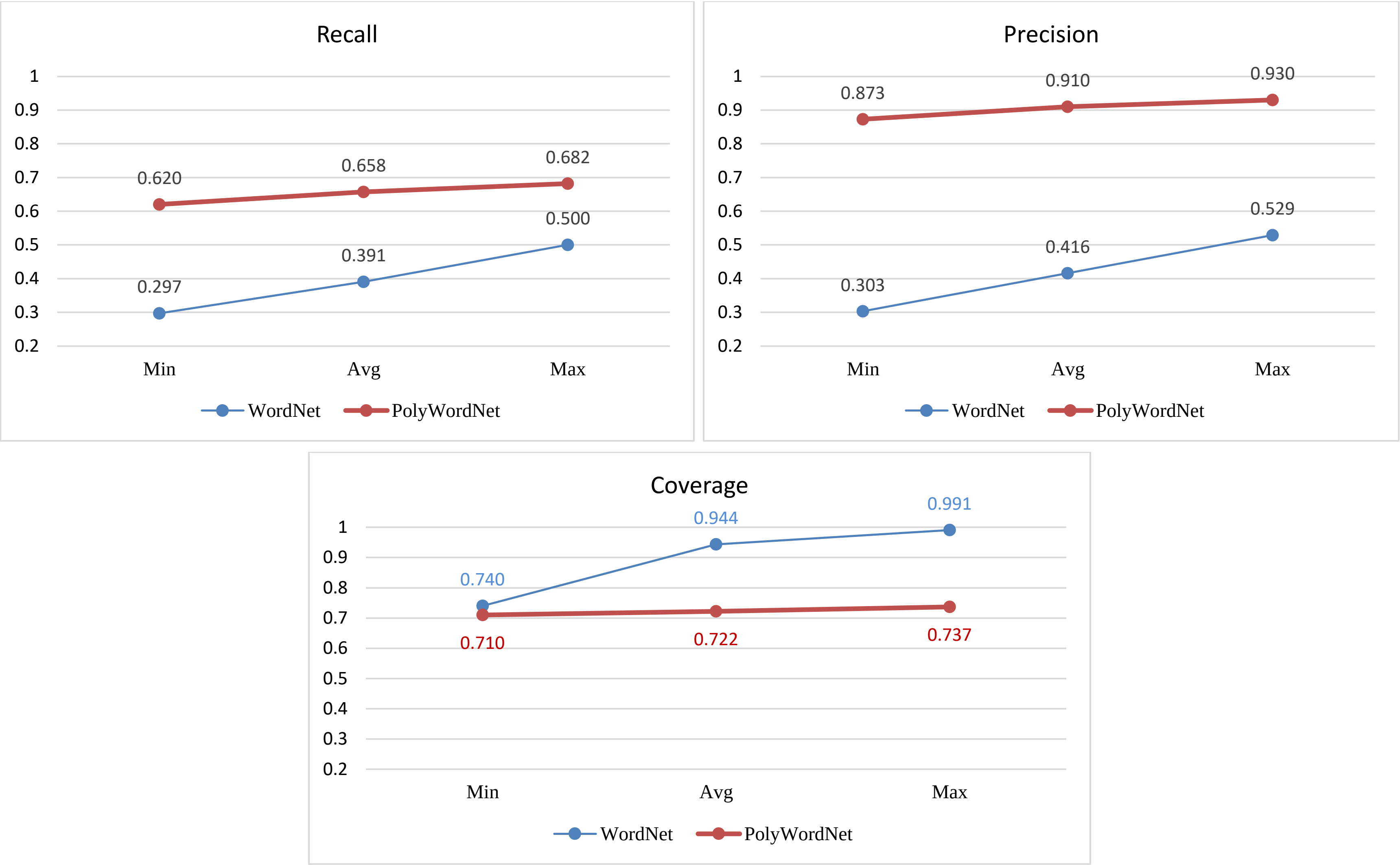
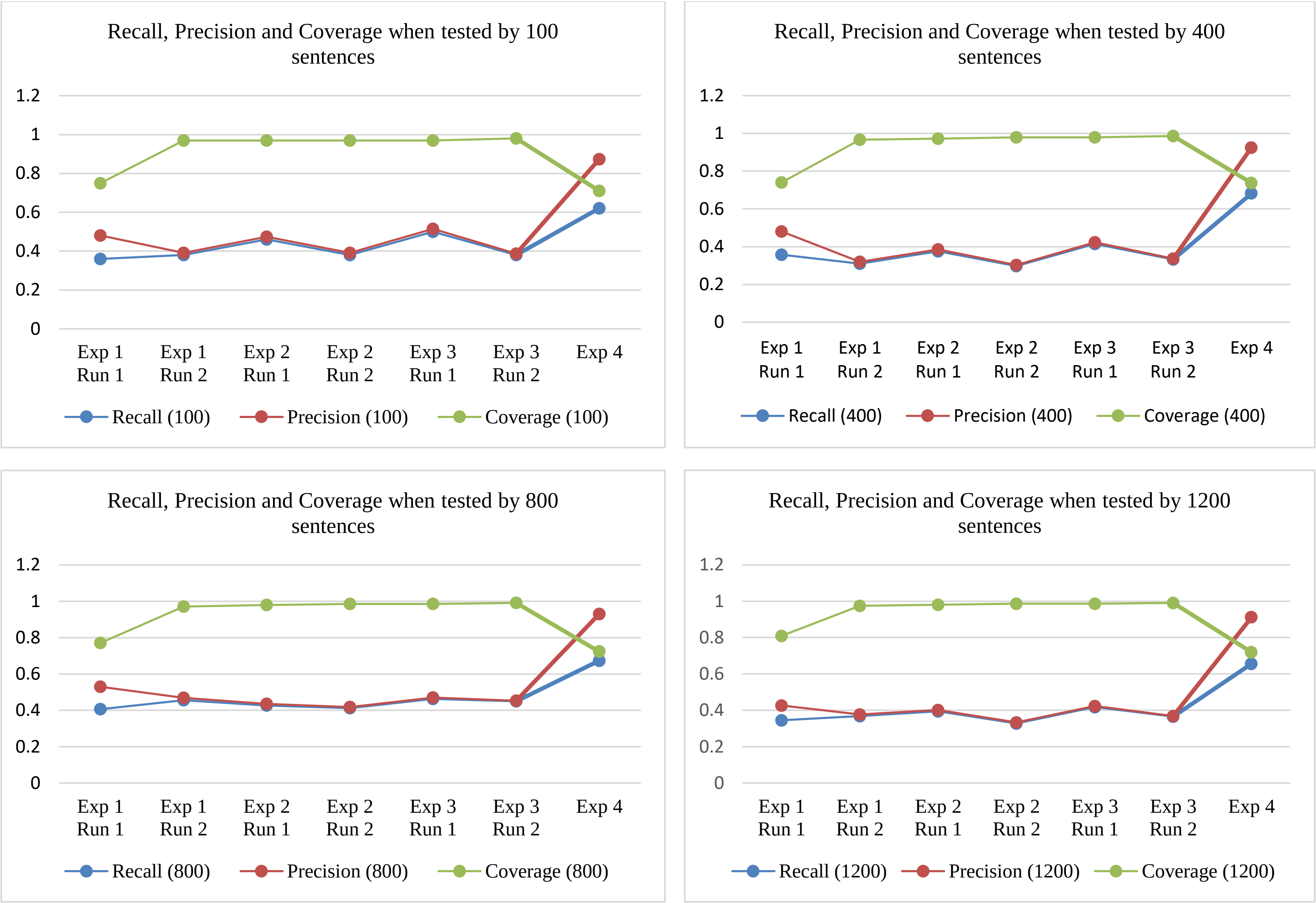


Figure 5. 6: Average *Recall, Precision* and *Coverage* of Series F Experiments



Recall, Precision and Coverage when tested by 100 sentences

Recall, Precision and Coverage when tested by 400 sentences

Recall, Precision and Coverage when tested by 800 sentences

Recall, Precision and Coverage when tested by 1200 sentences

Figure 5. 7: *Recall, Precision* and *Coverage* obtained in Series F Experiments in each increase in number of test sentences

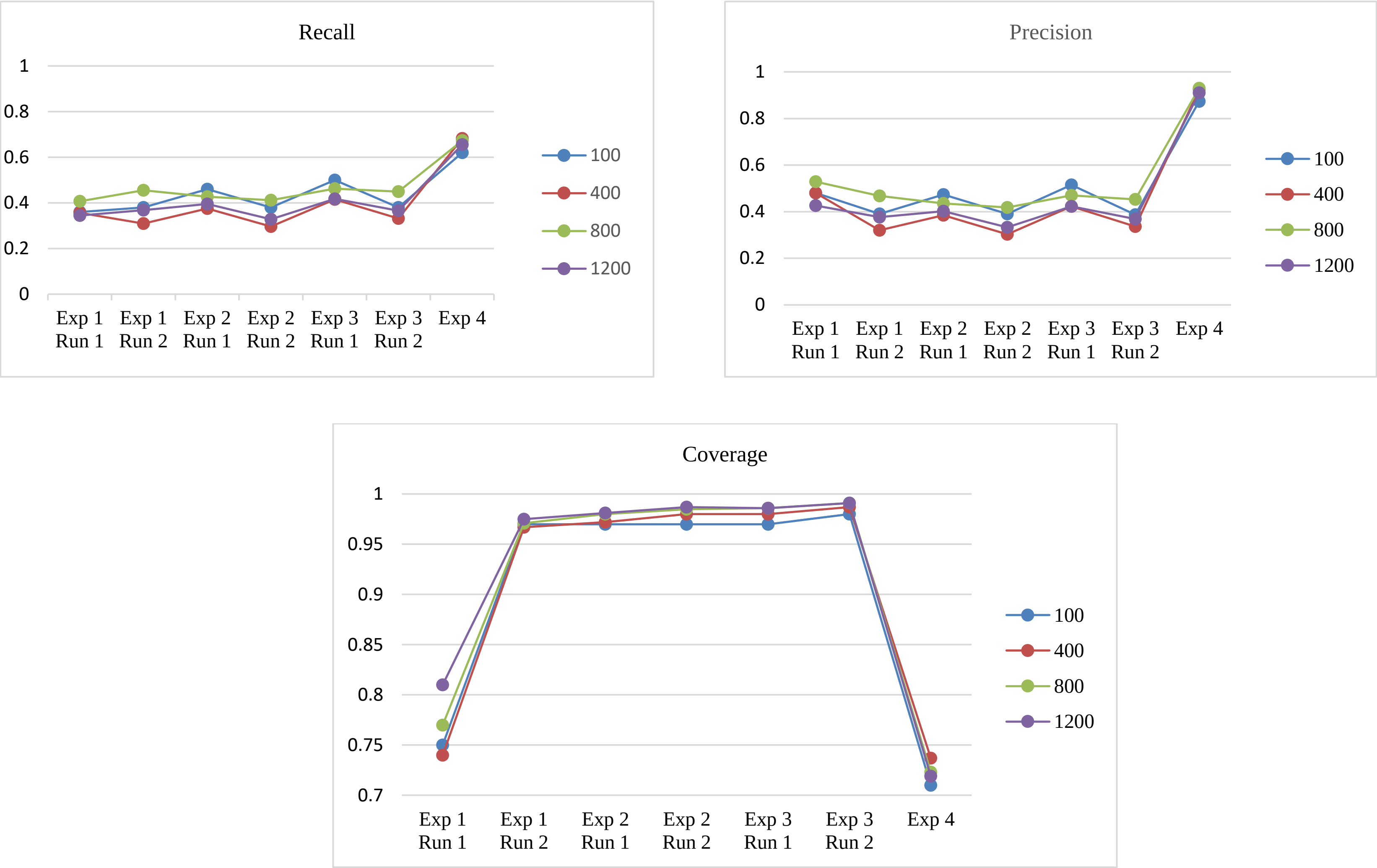


Figure 5. 8: *Recall, Precision* and *Coverage* obtained in Series F Experiments (*Recall, Precision* and *Coverage-wise*)

5.5 RESULT, ANALYSIS AND COMPARISON

This section first analyzes the results obtained in all experiments in a series-wise manner and finally summaries the findings drawn from the analysis of results of all experiments.

5.5.1 Result Analysis of Series A Experiments

The results of experiments show the increase in information in sense and context bags does not always increase the number of overlaps between context and correct sense of polysemy words. Sometimes, the inclusion of more information increases the number of overlaps between context and incorrect sense of polysemy word causing wrong disambiguation. In the experiments the ***Run 2*** always contains more information from WordNet with the inclusion of information from hypernyms of each word in sense bag. In addition, as going from ***Exp 1*** to ***Exp 3***, the information are increased in the next experiment as compared to previous experiment. It is observed that correctly disambiguated sentences in ***Run 1*** are incorrectly disambiguated in the ***Run 2***.

Table 5.7: Number of sentences which are correctly disambiguated in one run but incorrectly disambiguate in other run in Series A Experiments

SN	Result of test sentence in run A and B	Exp 1	Exp 2	Exp 3
1	No of correctly disambiguated sentences in ***Run 1*** but incorrectly disambiguated in ***Run 2*** (C -> W)	13	57	58
2	No of incorrectly disambiguated sentences in ***Run 1*** but correctly disambiguated in ***Run 2*** (W -> C)	14	25	30
3	No of correctly disambiguated sentences in both ***Run 1*** and ***Run 2*** (C -> C)	64	39	47
4	No of incorrectly disambiguated sentences in both ***Run 1*** and ***Run 2*** (W -> W)	89	59	45

The Table 5.7 shows that the number of correctly disambiguated sentences in ***Exp 1 Run 1*** but incorrectly disambiguate in ***Exp 1 Run*** 2 is 13. This number is 57 for ***Exp 2*** and is 58 for ***Exp 3***.

However, the number of incorrectly disambiguated sentences in ***Exp 2 Run 1*** but correctly disambiguated in ***Exp 2 Run 2*** is only 25 which is less than half of the number of correctly disambiguated sentences in ***Exp 2 Run 1*** but incorrectly disambiguated in ***Exp 2 Run 2***. Similarly, the number of incorrectly disambiguated sentences in ***Exp 3 Run 1*** but correctly disambiguated in ***Exp 3 Run 2*** is only 30 which is almost half of the number of correctly disambiguated sentences in ***Exp 3 Run 1*** but incorrectly disambiguated in ***Exp 3 Run 2.***

The result of first 6 experiments of Series A experiments show the fall and rise in accuracy as going from ***Exp 1 Run 1*** to ***Exp 3 Run 2***. In ***Exp 2 Run 1,*** the accuracy of the system is increased with the increase in information in sense and context bags by a large percentage of 10% as compared to ***Exp 1 Run 2***. In ***Exp 2 Run 2***, with the increase in information in sense bag decreased the accuracy again by a large percentage of 17.77% as compared with ***Exp 2 Run 1*** This indicates that increase in information from WordNet in sense and context bag doesn't ensure to increase the accuracy. The reason is the induction of noise information in sense and context bags. The noise information causes the higher overlaps for context with the wrong sense of polysemy words. The reason behind the noise information is the more common hypernyms of the words in context and the wrong sense of polysemy word. The accuracy of ***Exp 3 Run 2*** is found to be same as the accuracy of ***Exp 1 Run 1*** which uses only the word's gloss and synset for sense disambiguation. Even after including a huge amount of information from WordNet in sense and context bags in ***Exp 2 Run 2***, it has the same accuracy. This also indicates that the inclusion of more information in sense and context bag does not always increase the relatedness between the context and correct sense. With the same amount of data in the system and tested by the same test sentences, the accuracy of WSD method using PolyWordNet is found higher. This indicates the organizations of words in PolyWordNet is better especially for word sense disambiguation. The reason behind this is that the PolyWordNet deals with the relation between the related words and sense of polysemy words and directly connects them.

5.5.2 Result Analysis of Series B-E Experiments

In Series B to E experiments, the 7 different experiments are repeated and run for four times. In each next series, the data in PolyWordNet and sample WordNet is increased. The Series B experiments are run on data having 290 words, Series C experiments are run in 1477 words, Series D experiments is run in 2501 words and Series E experiments are run in 3541 words. In addition to the increase in data in PolyWordNet in each successive series, the amount of information to build sense and context bag is also increased in each successive experiments as moving from ***Exp 1 Run 1*** to ***Exp 3 Run 2***. As one moves horizontal from ***Exp 1 Run 1*** to ***Exp 3 Run 2***, there is increased amount of information in sense and context bags in each next experiment. As one move vertical from ***Series B to E***, there is increased number of data in PolyWordNet and sample WordNet. The purpose of increasing the data in PolyWordNet and sample WordNet in each successive iteration of experiments and increasing the information in sense and context bag is to observe how recall, precision and coverage changes with the increase in data in these lexical resources and with the increase of information in sense and context bags. These different Series of experiments are also tested using different number of test sentences. The Series B experiments are tested with 180 test sentences. The Series C, D and E are tested with 930, 1930 and 2905 test sentences.

From the results obtained in the first 6 experiments in Series B to E, four important observations are noted. The ***first*** observation is that the recall and precision are found increased as well as decreased as moving in horizontal successive experiments (see Table 5.4). Similar pattern is found on moving vertical down in each successive series. This means the WSD methods using WordNet has ups and downs in recall and precision with the increase in data in WordNet and with the increase in amount in sense and context bags. ***Second*** observation is that the precision of WSD algorithms are found to be very low as compared to the WSD methods using PolyWordNet. ***Third*** observation is that the coverage of WSD methods using WordNet is found to be in the range 0.611 to 1. However, even for high coverage, the WSD method using WordNet has poor recall as compared to the WSD methods using PolyWordNet which has higher recall even for low coverage. These are shown in Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4. The ***fourth*** observation is that there is no significant change in the recall, precision and coverage obtained in Series B as compared with the recall, precision and coverage obtained in Series E even if there is a big change in data from 290 in Series B to 3541 in Series E. This indicates that these recall, precision and coverage represents and shows the same pattern if the data in these lexical resources are made same as in current English WordNet.

The highest recall 0.60 is observed in ***Exp 3 Run 2*** in Series B and highest precision of 0.716 is observed in ***Exp 1 Run 1*** in Series B. The highest recall and precision are found when there is only 290 words in WordNet. This indicates only increasing more information in context bags and sense bags, does not improve the recall and precision. The recall of ***Exp 1 Run2*** is 0.578. When the information in sense and context bag is increased than in ***Exp 1 Run2***, the precision is decreased to 0.514 in ***Exp 2 Run1***. This indicates that the inclusion of more information in sense and context bag does not always increase the relatedness between the context and correct sense. The reason is the induction of noise information in sense and context bags. The noise information causes the higher overlaps for context with the wrong sense of polysemy words. The reason behind the noise information is the more common hypernyms of the words in context and the wrong sense of polysemy word.

The average recall and precision of ***Exp 4*** which uses the PolyWordNet are observed to be 0.989 and 0.994 which are exceptionally higher than that of the WSD methods using WordNet. The reason is that PolyWordNet has a better average coverage of 0.995 which is better than that of average coverage (0.919) of WSD methods using WordNet for the data in PolyWordNet and test data used to test the system. This is because almost all the related words were covered in the PolyWordNet for that test data by which it is tested. This is the reason why Series F experiments are run taking a separate test sentences from a popular Brown corpus.

5.5.3 Result Analysis of Series F Experiments

The Series F experiments are run on the same data (i.e. 3541 words in both PolyWordNet and sample WordNet) as in Series E experiments. The Series F experiments are tested with 1200 test sentences randomly taken from Brown Corpus. The results obtained even in Series F experiments shows better recall and precision of WSD method using PolyWordNet (***Exp 4)*** with a great difference comparing to WSD method using WordNet even in the low coverage of 0.722 (Avg.). No single data is added in the PolyWordNet and sample WordNet and no single related word is added in PolyWordNet. All settings are kept constant as in Series E experiments. Only the Series F experiments are tested by another set of test sentences which are randomly taken from Brown corpus. The reason behind running these F series experiments is that the recall, precision and recall of WSD method using PolyWordNet are observed exceptionally high.

Therefore, the systems are tested by a new set of randomly collected test sentences from Brown corpus. In Series F, the 7 different experiments are repeated and run for four times each time increasing the number of test sentences. The results are shown in Table 5.6. The Series F experiments are first tested by 100 test sentences, then by 400, 800 and 1200 test sentences.

The results of the first 6 experiments which uses WordNet show the similar pattern in Series F experiments just like in the results of Series B to Series E experiments. The four important observations noted in Series B to Series E experiments are also noted in Series F experiments. The ***first*** observation is that the recall and precision are found increased as well as decreased as moving in horizontal successive experiments (see Table 5.6). Similar pattern is found on moving vertical down in each successive series. This means the WSD methods using WordNet has ups and downs in recall and precision with the increase in data in WordNet and with the increase in amount in sense and context bags. ***Second*** observation is that the precision of WSD algorithms are found to be low enough as compared to the WSD methods using PolyWordNet. ***Third*** observation is that the coverage of WSD methods using WordNet is found to be in the range 0.740 to 0.991. However, even for high coverage, the WSD method using WordNet has poor recall as compared to the WSD methods using PolyWordNet which has higher recall even for low coverage. The ***fourth*** observation is that the recall, precision and coverage of first 6 experiments (which uses WordNet) in Series F represents the recall, precision and coverage obtained in Series A to Series E experiments.

The highest recall 0.50 is observed in ***Exp 3 Run 1*** and highest precision 0.529 is observed in ***Exp 1 Run 1***. There is no only rise in recall and precision when the information in sense and context bags are increased as in Series A to Series E experiments. The results show there are falls in recall and precision even when the information in sense and context bags are increased. This indicates only increasing more information in context bags and sense bags, does not improve the recall and precision. The inclusion of more information in sense and context bag does not always increase the relatedness between the context and correct sense. The reason is the induction of noise information in sense and context bags. The noise information causes the higher overlaps for context with the wrong sense of polysemy words. The reason behind the noise information is the more common hypernyms of the words in context and the wrong sense of polysemy word.

The results of Series F experiments shows minimum recall of WSD method using PolyWordNet (***Exp 4)*** is 0.62 which is more than double of minimum recall (0.297) and is also greater than the maximum recall (0.50) of WSD method using WordNet. The precisions of WSD method using PolyWordNet (***Exp 4)*** are much better than that of WSD method using WordNet. The minimum precision of WSD method using PolyWordNet (***Exp 4)*** is 0.873 which is almost 3 times greater than that of minimum precision (0.303) and is also greater than the maximum precision (0.529) of WSD method using WordNet. However, the WSD method using WordNet has a greater coverage up to 0.991 (max). The minimum coverage observed for WSD method using WordNet is 0.740 which is even greater than maximum coverage of WSD method using PolyWordNet. These are shown in Figure 5.5, Figure 5.6, Figure 5.7 and Figure 5.8.

5.5.4 Findings from Result Analysis

For Overlap Count Knowledge based Word Sense Disambiguation method, the information of word’s synset and gloss from WordNet is found to be very less as in the case of ordinary dictionary. When the information in sense and context bag are increased with inclusion of hypernyms and other relations of WordNet, the number of overlaps between the context and the correct sense is found to be increased resulting in better recall and precision. However, the result of experiments also show the great fall of recall and precision when the information in sense and context bag are increased with inclusion of hypernyms and other relations of WordNet. Therefore, from this it can be concluded that increase in information in sense and context bag does not ensure for better recall and precision. The only reason for fall in recall with increase in information in sense and context bag is due to the induction of noise information which causes the more number of overlaps between context bag and incorrect sense bag resulting in wrong disambiguation. The cause of induction of this noise information is found to be the common hypernyms for multiple senses of the same polysemy word in WordNet.

The WSD methods using WordNet show less recall and precision as compared with the recall and precision of WSD methods using PolyWordNet. The result of all series experiments including Series F experiments show a significantly higher recall and precision of WSD method using PolyWordNet even for the low coverage on data. These results shows the word organization of PolyWordNet best suits for word sense disambiguation. The only reason is that the PolyWordNet organizes the related words based on senses of polysemy words. The organization of words in PolyWordNet resolves the problem of noise information which was caused by the common hypernyms from WordNet. Since a related word is not connected with the multiple senses of polysemy word in PolyWordNet, it eliminates the induction of noise information resulting in high recall and precision for word sense disambiguation.

Inclusion of information only from word’s gloss and synset from WordNet contain less information. The inclusion of more information from hypernym and other relations of WordNet introduce the noise information. The PolyWordNet resolves this problem of less distinct information from WordNet since a single word which provides the exact meaning of a polysemy word in a context is directly connected with the correct sense of the polysemy word. Therefore, even a single related word in PolyWordNet is sufficient for sense disambiguation in the given context. However, for this PolyWordNet needs high coverage.

The number of overlaps between a context bag and correct sense of a polysemy words in Overlap Count Knowledge based Word Sense Disambiguation method depends on the words used in gloss to define the meaning of the word. This is resolved in PolyWordNet since the WSD using PolyWordNet depends only on the relation between related words and their respective senses of polysemy words. Higher the relations between related words and their respective senses of polysemy words in PolyWordNet, greater the recall and precision for word sense disambiguation.

Finally, the PolyWordNet is the only lexical resource that deals with the senses of polysemy words and their respective related words. It organizes the words based on the relation between the senses of polysemy words and their respective related words. These relations from PolyWordNet can be used to achieve higher recall and precision for word sense disambiguation. The results from the experiments indicates the organization of words in PolyWordNet is better especially for word sense disambiguation.

5.5.5 Handled and Unhandled Cases of WSD using PolyWordNet

This subsection describes the various cases that can be handled by WSD methods using PolyWordNet. It also presents the cases which are not handled at the moment by the WSD methods using PolyWordNet for word sense disambiguation.

Cases that can be handled by WSD methods using PolyWordNet

The WSD method using PolyWordNet can disambiguate the sense of a polysemy word in a context for both simple sentences as well as compound sentences in the following conditions:

1. There should be at least a related word in the context for the polysemy word.
2. The related word should be at PolyWordNet and connected with its respective sense of the polysemy word.
3. Two or more context words should not lead to multiple senses of the same polysemy words. This means context should be simple enough.
4. In a paragraph, even if a related word is in one sentence and the polysemy word is in another sentence, WSD method using PolyWordNet can disambiguate the sense of the polysemy word. For example- “The book and copy are on the table. He has a pen already in his hand”. In such context, the sense of polysemy word “pen” can be determined by using the context words “book” and “copy” in the previous sentence of the same paragraph.

Cases that cannot be handled by WSD methods using PolyWordNet

The WSD method using PolyWordNet cannot disambiguate the sense of a polysemy word in a context for a sentence in the following conditions:

1. If a context contains only a single word which is polysemy.
2. If a context contains context words but are not found in PolyWordNet.
3. Whether the context is simple or compound or even a paragraph, it should not contain two different context words that represent the two different meaning of the same polysemy word. For instance, suppose a polysemy word “bank” in a context “I deposited

money in the bank which is at the side of Bagmati river" In such context, the related word such as "deposited" and "money" lead to the sense "a financial institution" at the same time the related word "Bagmati" and "river" lead to the sense "a river bank". In such case, the WSD method may produce wrong sense.

4. Suppose a context "He is her man". In this context "man" is polysemy word. In this sentence, the meaning of man is husband. The meaning of the polysemy word "man" is given by the pronoun "her". At the moment, pronouns are not included in PolyWordNet. Therefore, WSD method using PolyWordNet cannot handle such context for sense disambiguation.

5.6 RESULT OF EXPERIMENTS AND VALIDATION

PolyWordNet is a new lexical database. It organizes the words based on the senses of polysemy words. The validity of the word's organization in PolyWordNet is tested using a standard validation tool. WSD algorithm is used as a validation tool in the similar way it was done to validate the BalkaNet WordNet [92]. In the same way, the WSD algorithm tool is also used to validate the data in PolyWordNet and test sentences.

5.6.1 Validation of PolyWordNet

The PolyWordNet is built using the data from WordNet. The data taken from WordNet are already valid. However, the validity of PolyWordNet is again checked by using the WSD algorithm as a validation tool. The experiments ***Exp 1 Run 1, Exp 1 Run 2, Exp 2 Run 1, Exp 2 Run 2, Exp 3 Run 1*** and ***Exp 3 Run 2*** in each series are tested by 4105 test sentences. The accuracy range of experiments in Series A to E is 35.56% to 60%. In addition, the range of accuracy for Series F experiments is 29.7% to 50%. This accuracies ranges are aligned with and represent the ***reference accuracy range*** (30 - 70) %. Further, the result from the experiments shows the higher accuracy of WSD method which uses PolyWordNet. These experimental evidences clearly indicate the PolyWordNet is aligned with WordNet and the word's organization in PolyWordNet is acceptable and valid for word sense disambiguation.

5.6.2 Validation of Data and Test Sentences

The result of 35 different experiments in Series A to E shows that their accuracy range from 35.56% to 60%. This accuracy range is aligned with and represent the ***reference accuracy range*** (30 - 70) %. These accuracies were obtained by testing 2905 test sentences on 3541 data. Therefore, these experimental results clearly indicates that the data used to build PolyWordNet and the test sentences that are used to test the experiments are also valid. In addition, the 1200 test sentences taken from the Brown corpus show the accuracy range from 29.7% to 50%. This accuracy range is also aligned with the accuracy range obtained in Series A to E experiments and with the ***reference accuracy range*** (30 - 70) %. This also indicates the 2905 test sentences collected in this research are valid. These test sentences are available in Kaggle dataset [100].

Chapter 6

Conclusions and Recommendations

6.1 CONCLUSION

The structures of existing various WordNets such as GermaNet, EuroWordNet, Japanese WordNet, Chinese WordNet, Hindi WordNet, Nepali WordNet and various multilingual WordNets such as EuroWordNet and IndoWordNet are studied in detail. In addition, the organization of words in these WordNets are observed in detail. Different variation of original Lesk algorithms that uses the information from WordNets and their obtained results are studied. During this study, various factors and issues are investigated. The first cause for WSD approach using WordNet is noise information which causes the wrong disambiguation. The second cause or issue found in WordNet is that it lacks dealing with relations with polysemy words. Similarly, it is investigated that the overlap count WSD process depends on the words used in definition of a word in dictionary or WordNet. It is the third issue that is investigated.

Every context, if it contains a polysemy word, it also contains at least a related word. Based on this fact, a new organization of words is developed. The idea is to organize the words based on polysemy words. A sense of a polysemy word and its related words are connected creating the links between these words. This organization of words results in forming the clusters of polysemy words and their related words. This new lexical resource is called as PolyWordNet. In addition, a new WSD algorithm is also developed. This algorithm uses the relations from PolyWordNet for word sense disambiguation. The first and second issues are resolved in PolyWordNet by organizing the words in PolyWordNet based on senses of polysemy words. Similarly, the third issue is resolved in WSD process that uses the PolyWordNet. WSD that uses PolyWordNet for sense disambiguation does not depend on the words in gloss rather it depend on the relation among polysemy words and related words.

The PolyWordNet currently contains 3541 words. This number includes both polysemy words and single sense words. There are altogether 762 polysemy words with 1748 occurrences. The number of single sense words is 1793. The number of noun, verbs, adverbs and adjectives are 2264, 859, 51 and 367 respectively. The number of test sentences (Test Data) generated in this research is 2905. There are 1200 sentences collected from ***news*** category of Brown corpus. Altogether 4105 test sentences are used to test the experiments repeating for 63 times. Six series of experiments (named as series A to series F) are designed. Each series of experiments contains 7 different settings (named as ***Exp 1 Run 1, Exp 1 Run 2, Exp 2 Run 1, Exp 2 Run 2, Exp 3 Run 1, Exp 3 Run 2 and Exp 4***). The Series A to Series E experiments are tested by 2905 test sentences generated in this research. The Series F experiments are tested by 1200 sentences taken from Brown corpus.

The results of **63** experiments performed on **3541 word**s and tested by **4105 test sentences** show the significantly higher recall and precision of new WSD algorithm using PolyWordNet than that of the contextual overlap count WSD approaches which uses the information from WordNet. The average recall of WSD algorithm that uses PolyWordNet when tested with 1200 sentences taken from Brown corpus is 0.658 with average precision of 0.910. The average recall of WSD algorithm that uses WordNet when tested with those 1200 sentences is 0.391 with average precision of 0.416. The recall and precision of WSD algorithm using the PolyWordNet are found significantly higher than that of the contextual overlap count WSD methods that uses the WordNet in all series of experiments. These results have proved that the organization of the words in **PolyWordNet** is acceptable for word sense disambiguation with reference to the lexical database- **WordNet.**

6.2 RECOMMENDATION

The findings of this research work show significantly higher accuracy of WSD method that uses PolyWordNet than that of contextual overlap count WSD methods using WordNet. This indicates that the PolyWordNet is a very useful lexical database for word sense disambiguation. Therefore, the further research for the improvement of PolyWordNet is highly recommended. In this research work, the prepositions and pronoun are not included as related words. However, they can play very important role as related words for sense disambiguation. Therefore, the inclusion of prepositions and pronouns in related words is recommended.

The related words which are the key elements for PolyWordNet are manually generated in this research. If they can be generated automatically using the information from websites like Wikipedia and available corpus like Brown corpus, it will automate the building of PolyWordNet. Automatic generation of related words is, therefore, recommended for future work to develop a self-organizing PolyWordNet.

References

[1] "Ethnologue Languages of the World," Feb 2015. [Online]. Available: http://www.ethnologue.com/world. [Accessed Feb 2015].

[2] "Languages of the World - Interesting facts about languages," Feb 2015. [Online]. Available: http://www.bbc.co.uk/languages/guide/languages.shtml. [Accessed Feb 2015].

[3] N. Ide and J. Véronis, "Word Sense Disambiguation: The State of the Art," *Computational Linguistics*, vol. 24, pp. 2-40, 1998.

[4] G. A. Miller, "WordNet 2.1," 2005.

[5] G. A. Miller, "WordNet: A Lexical Database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39-41, Nov. 1995.

[6] G. A. Miller, "Nouns in WordNet: A Lexical Inheritance System," *International journal of Lexicography*, vol. 3, no. 4, pp. 245-264, 1990.

[7] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. J. Miller, "Introduction to WordNet: An On-line Lexical Database," *International journal of lexicography*, vol. 3, no. 4, pp. 235-244, 1990.

[8] G. A. Miller, C. Fellbaum, R. Tengi, P. Wakefield, H. Langone and B. R. Haskell, "Wordnet," 12 June 2012. [Online]. Available: http//:wordnet.princeton.edu/wordnet/. [Accessed 12 June 2012].

[9] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller, "WordNet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, no. 4, pp. 235--244, 1990.

[10] E. Agirre and P. Edmonds, Word Sense Disambiguation: Algorithms and Applications, 1st ed., Springer Publishing Company, Incorporated, 2007.

[11] W. Weaver, "Translation," *Machine translation of languages,* vol. 14, pp. 15-23, 1955.

[12] M. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionries: How to Tell a Pine Cone from an Ice Cream Cone," in *Proceedings of the 5th Annual International Conference on Systems Documentation*, Toronto, Ontario, Canada, 1986.

[13] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet," in *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing,* 2002.

[14] K. Fragos, Y. Maistros and C. Skourlas, "Word sense disambiguation using wordnet relations," in *First Balkan Conference in Informatics*, Thessaloniki, Greece, 2003.

[15] H.-C. Seo, H. Chung, H.-C. Rim and S. H. Myaeng, "Unsupervised word sense disambiguation using WordNet relatives," *Computer Speech & Language,* vol. 18, no. 3, pp. 253-273, July 2004.

[16] A. Montoyo, A. Suárez, G. Rigau and M. Palomar, "Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods," *Journal Of Artificial Intelligence Research,* vol. 23, no. 1, pp. 299-330, March 2005.

[17] S. Liu, C. Yu and W. Meng, "Word Sense Disambiguation in Queries," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, Bremen, Germany, 2005.

[18] U. R. Dhungana and S. Shakya, "Word sense disambiguation in Nepali language," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on*, Bangkok, Thailand, May 2014.

[19] U. R. Dhungana and S. Shakya, "Hypernymy in WordNet, Its Role in WSD and Its Limitations," *International Journal of Simulation, Systems, Science & Technology,* vol. 16, no. 6, December 2015.

[20] U. R. Dhungana and S. Shakya, "Hypernymy in WordNet, Its Role in WSD, and Its Limitations," in *The 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN),*, Riga, 2015.

[21] A. Nikunen, "Different approaches to word sense disambiguation," Language technology and applications, 2007.

[22] A. Kaplan, "An experimental study of ambiguity and context," *Mechanical Translation,* vol. 2, no. 2, pp. 39-46, November 1955.

[23] S. Madhu and D. W. Lytle, "A Figure of Merit Technique for the Resolution of Non-Grammatical Ambiguity," *Mechanical Translation,* vol. 8, no. 2, pp. 9-13, February 1965.

[24] Y. Bar-Hillel, "The Present Status of Automatic Translation of Languages," *Advances in Computers,* vol. 1, pp. 91-163, December 1960.

[25] J. Hutchins, "ALPAC: the (in)famous report," *MT News International,* no. 14, pp. 9-12, June 1996.

[26] D. Yarowsky, "Word-sense disambiguation.," in *The Handbook of Natural Language Processing*, R. Dale, H. Moisl and H. Somers, Eds., New York, Marcel Dekker, 2000, pp. 629-654.

[27] R. Navigli, "Word Sense Disambiguation: A Survey," *ACM COMPUTING SURVEYS,* vol. 41, no. 2, pp. 1-69, 2009.

[28] Y. Wilks, "A preferential, pattern-seeking, Semantics for natural language inference," *Artificial Intelligence,* vol. 6, no. 1, pp. 53-74, 1975.

[29] T. Pedersen, "Unsupervised corpus-based methods for WSD," pp. 133-166, June 2006.

[30] J. A. Guthrie, L. Guthrie, Y. Wilks and H. Aidinejad, "Subject dependent co-occurrence and word sense disambiguation," in *The 29th Annual Meeting of the Association for Computational Linguistics*, 1991.

[31] D. Yarowsky, "Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora," Nantes, France, 1992.

[32] M. Á. M. J. L. a. J. M. Jorge Morato, "WordNet Applications".

[33] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra and R. L. Mercer, "Word-sense Disambiguation Using Statistical Methods," in *The 29th Annual Meeting on Association for Computational Linguistics*, Berkeley, California, 1991.

[34] X. Zhou and H. Han, "Survey of Word Sense Disambiguation Approaches," in *FLAIRS Conference*, 2005.

[35] B. Mutlum, *Word Sense Disambiguation Based on Sense Similarity and Syntactic Context,* 2005.

[36] R. Sharma and P. G. Bhatia, *Word Sense Disambiguation for Hindi Language*, 2008.

[37] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 1995.

[38] S. Brody, *Closing the Gap in WSD: Supervised Results with Unsupervised Methods,* Citeseer, 2009.

[39] A. Fujii, "Corpus-Based Word Sense Disambiguation," *arXiv preprint cmp-lg/9804004*, 1998.

[40] E. Agirre, O. L. d. Lacalle and A. Soroa, "Knowledge-Based WSD and Specific Domains: Performing Better than Generic Supervised WSD," in *IJCAI*, 2009.

[41] C. Fellbaum, "WordNet," in *Theory and applications of ontology: computer applications*, Springer, 2010, pp. 231-243.

[42] D. Jurafsky and J. H. Martin, "WordNet: Word Relations, Senses and Disambiguation," in *Speech and Language Processing*, 2018.

[43] Oxford Advanced Learner's Dictionary, New York: Oxford University Press, 2005.

[44] C. Fellbaum, Ed., WordNet: An Electronic Lexical Database, Cambridge, MA: MIT Press, 1998.

[45] "WordNet," 7 Jan 2014. [Online]. Available: https://www.en.wikipedia.org/wiki/WordNet. [Accessed 7 Jan 2014].

[46] D. A. Cruse, Lexical Semantics, New York: Cambridge University Press, 1986.

[47] C. Fellbaum, "English Verbs as a Semantic Net," *International Journal of Lexicography,* vol. 3, no. 4, pp. 278-301, 1990.

[48] C. Fellbaum, D. Gross and K. Miller, "Adjectives in WordNet," *International Journal of Lexicography,* vol. 3, no. 4, pp. 265-277, 1990.

[49] B. Hamp and H. Feldweg, "Germanet-a lexical-semantic net for german," in *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997.

[50] P. Vossen, "EuroWordNet: a multilingual database for information retrieval," in *Proceedings of the DELOS workshop on Cross-language Information Retrieval*, 1997.

[51] P. Vossen, "Introduction to eurowordnet," *Computers and the Humanities,* vol. 32, no. 2-3, pp. 73--89, 1998.

[52] H. ISAHARA, F. BOND and K. UCHIMOTO, "Development of the Japanese WordNet," 2008.

[53] F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi and K. Kanzaki, "Enhancing the japanese wordnet," in *Proceedings of the 7th workshop on Asian language resources*, 2009.

[54] C.-R. Huang, R.-Y. Chang and H.-P. Lee, "Sinica BOW (Bilingual Ontological Wordnet): Integration of Bilingual WordNet and SUMO," 2004.

[55] R. Xu, Z. Gao, Y. Pan, Y. Qu and Z. Huang, "An integrated approach for automatic construction of bilingual Chinese-English WordNet," in *Asian Semantic Web Conference*, 2008.

[56] C.-R. Huang, S.-K. Hsieh, J.-F. Hong, Y.-Z. Chen, I.-L. Su, Y.-X. Chen and S.-W. Huang, "Chinese Wordnet: Design, implementation, and application of an infrastructure for cross-lingual knowledge processing," *Journal of Chinese Information Processing,* vol. 24, no. 2, pp. 14-23, 2010.

[57] S. Wang and F. Bond, "Building the chinese open wordnet (cow): Starting from core synsets," in *Proceedings of the 11th Workshop on Asian Language Resources, a Workshop at IJCNLP*, 2013.

[58] I. Niles and A. Pease, "Mapping WordNet to the SUMO ontology," in *Proceedings of the ieee international knowledge engineering conference*, 2003.

[59] P. Bhattacharyya, "Hindi Wordnet," Centre for Indian Language Technology (CFILT), Computer Science and Engineering Department, IIT Bombay, Mumbai, 5 March 2017. [Online]. Available: http://www.cfilt.iitb.ac.in/wordnet/webhwn/other/hwn_docs_2.pdf. [Accessed 5 March 2017].

[60] P. Bhattacharyya, "IndoWordnet," in *Proceedings of LREC*, 2010.

[61] A. Nagvenkar, J. Pawar and P. Bhattacharyya, "IndoWordNet Conversion to Web

Ontology Language (OWL)," 2016.

[62] A. Chakrabarty, B. S. Purkayastha and A. Roy, "Experiences in building the Nepali WordNet - insights and challenges," in *The Fifth Global Wordnet Conference @ CFILT, IIT Bombay*, Mumbai, 2010.

[63] A. Roy, S. Sarkar and B. S. Purkayastha, "Knowledge Based Approaches to Nepali Word Sense Disambiguation," *International Journal on Natural Language Computing (IJNLC)*, vol. 3, no. 3, pp. 51-63, 2014.

[64] K. Robkop, S. Thoongsup, T. Charoenporn, V. Sornlertlamvanich and H. Isahara, "WNMS: Connecting the distributed wordnet in the case of Asian WordNet," in *Principles, Construction, and Applications of Multilingual Wordnets. Proceedings of the Fifth Global WordNet Conference (GWC 2010), India. Narosa Publishing*, 2010.

[65] F. Bond and R. Foster, "Linking and Extending an Open Multilingual Wordnet," 2013.

[66] S. Patwardhan, S. Banerjee and T. Pedersen, "Using measures of semantic relatedness for word sense disambiguation," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2003.

[67] T. Pedersen, S. Patwardhan and J. Michelizzi, "WordNet:: Similarity: measuring the relatedness of concepts," in *Demonstration papers at HLT-NAACL 2004*, 2004.

[68] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *International Joint Conference on Artificial Intelligence (Ijcai)*, 2003.

[69] M. Sinha, M. Kumar, P. Pande, L. Kashyap and P. Bhattacharyya, "Hindi word sense disambiguation," in *Proceedings of International Symposium on Machine Translation, Natural Language Processing and Translation Support Systems*, Delhi, India, 2003.

[70] S. G. Kolte and S. G. Bhirud, "WordNet: a knowledge source for word sense disambiguation," *International Journal of Recent Trends in Engineering*, vol. 2, no. 4, 2009.

[71] K. Shirai and T. Yagi, "Learning a Robust Word Sense Disambiguation Model Using Hypernyms in Definition Sentences," in *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 2004.

[72] S. G. Kolte and S. G. Bhirud, "Exploiting Links in WordNet Hierarchy for Word Sense

Disambiguation of Nouns," in *Proceedings of the International Conference on Advances in Computing, Communication and Control*, Mumbai, India, 2009.

[73] U. R. Dhungana, S. Shakya, K. Baral and B. Sharma, "Word Sense Disambiguation using WSD specific WordNet of polysemy words," in *Semantic Computing (ICSC), 2015 IEEE International Conference on*, California, USA, Feb, 2015.

[74] J. Cowie, J. Guthrie and L. Guthrie, "Lexical disambiguation using simulated annealing," in *Proceedings of the 14th conference on Computational linguistics*, 1992.

[75] A. Kilgarriff and J. Rosenzweig, "English Senseval: Report and Results.," in *LREC*, 2000.

[76] R. Garside, G. Sampson and G. Leech, The computational analysis of English: A corpus-based approach, vol. 57, Longman, 1988.

[77] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," *CoRR,* Vols. abs/cmp-lg/9511007, 1995.

[78] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," *WordNet: An electronic lexical database,* vol. 49, no. 2, pp. 265-283, 1998.

[79] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet: An electronic lexical database,* vol. 305, pp. 305-332, 1998.

[80] P. Resnik, "Selectional preference and sense disambiguation," in *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How*, Washington, DC, 1997.

[81] G. K. Zipf, *Human behavior and the principle of least effort,* JOHN WILEY \& SONS INC 111 RIVER ST, HOBOKEN, NJ 07030 USA, 1950.

[82] J. Lyons, "Semantics.(2 vols.) Cambridge," *Cambridge University,* 1977.

[83] U. R. Dhungana and S. Shakya, "Polywordnet: A lexical database," in *Inventive Computation Technologies (ICICT), International Conference on*, 2017.

[84] C. Fellbaum and R. Tengi, "WordNet- A Lexical Database for English," 2020. [Online]. Available: https://wordnet.princeton.edu/documentation/wnstats7wn.

[85] "WordNet Search - 3.1," 2016. [Online]. Available: http://wordnetweb.princeton.

edu/perl/webwn.

[86] " Oxford Learner’s Dictionary," 2016. [Online]. Available: https://www.oxfordlearners dictionaries.com/.

[87] "Cambridge Dictionary," 2016. [Online]. Available: https://dictionary.cambridge.org/.

[88] A. Clark and S. Lappin, The Handbook of Computational Linguistics and Natural Language Processing, 2010, pp. 197 - 220.

[89] P. Edmonds and E. Agirre, "Word sense disambiguation," *Scholarpedia,* vol. 3, no. 7, p. 4358, 2008.

[90] T. Cohn, "Performance metrics for word sense disambiguation," Melbourne, Australia, 2003.

[91] A. Clark and C. Fox, The Handbook of Computational Linguistics and Natural Language Processing, The Handbook of Computational Linguistics and Natural Language Processing, 2010.

[92] D. Tufis, R. Ion and N. Ide, "Word Sense Disambiguation as a Wordnets' Validation Method in Balkanet," *LREC,* Jan 2004.

[93] X. a. S. S. Li, "A WordNet-based Algorithm for Word Sense Disambiguation," 07 1995.

[94] K. a. Y. W. Orkphol, "Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet," *Future Internet,* vol. 11, p. 114, 05 2019.

[95] E. F. Ayetiran and K. K. Agbele, "An Optimized Lesk-Based Algorithm for Word Sense Disambiguation," *Open Computer Science,* vol. 8, pp. 165 - 172, 2018.

[96] Y. Wang, M. Wang and H. Fujita, "Word Sense Disambiguation: A comprehensive knowledge exploitation framework," *Knowledge-Based Systems,* vol. 190, p. 105030, 2020.

[97] P. Sharma and N. Joshi, "Knowledge-Based Method for Word Sense Disambiguation by Using Hindi WordNet," *Engineering, Technology & Applied Science Research,* vol. 9, no. 2, pp. 3985-3989, April 2019.

[98] S. Bhingardive and P. Bhattacharyya, "Word Sense Disambiguation Using IndoWordNet," in *The WordNet in Indian Languages,* N. S. Dash, P. Bhattacharyya and J. D. Pawar, Eds., Singapore, Springer Singapore, 2017, pp. 243-260.

[99] M. Kumar, P. Mukherjee, M. Hendre, M. Godse and B. Chakraborty, "Adapted Lesk Algorithm based Word Sense Disambiguation using the Context Information," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, 2020.

[100] U. R. Dhungana, "Kaggle- Test Data for Word Sense Disambiguation Evaluation," 2020. [Online]. Available: https://www.kaggle.com/udayarajdhungana/test-data-for-word-sense-disambiguation. [Accessed 19 04 2020].

Appendices

Appendix 1: Polysemy Words in PolyWordNet

1
10
11
12
2
3
4
5
6
7
8
9
abortion
abroad
academic
accent
account
accounting
action
active
address
advance
advantage
advise
aim
Air
alpha
amazon
amend
amended
amendment
amount
anchor
appeal
apple
arctic
area
arm
arrest
arrow
Atlantic
atom
atrocity
attempt
audio
audit
away
back
bag
bake
baked
balance
balloon
band
bank
bankrupt
bar
bark
base
basic
basin
basketball
bass
bat
bath
bathroom
be
beam
bear
beat
beautiful
bed
beep
before
begin
belly
bench
bend
Bet
better
bill
binary
biscuit
biscuits
bit
blank
block
blue
board
boil
bold
bolt
book
booth
bore
bottle
bottom
bow
bowl
box
brace
bravery
break
bright
buck
buckle
building
business
C
cabinet
call
calm
can
capital
captain
card
carry
case
cast
cell
centre
cereal
chain
chair
change
channel
charge
charm
chase
check
chew
chip
chipping
chocolate
christmas
circle
cite
civil
classic
clear
clearly
clip
close
closing
club
coach
coal
coast
coat
code
coding
coffee
cola
collaborate
collect
collected
Colour
column
comment
commercial
commit
company
complex
component
compound
concrete
confidential
constant
construction
consulting
contact
contract
control
cookie
cool
copy
core
corn
corrugated
cost
course
court
cover
covered

crack
craftsman
crane
create
criminal
crop
cup
current
custom
customize
customized
cut
dark
date
deep
deficiency
degree
deliver
deluxe
density
desert
design
despair
detail
develop
diameter
die
difficult
direction
director
dirt
disconnected
discuss
dish

display
disturbing
ditch
document
dog
dough
dove
down
draft
drain
draw
dress
Drew
drift
drill
drink
drive
driving
ear
ease
editorial
egg
eight
elderly
eldest
eleven
empty
engage
enjoy
enough
environment
al
equal
estimation

ethical
even
every
evidence
exact
executive
exit
explosion
export
extract
extreme
face
fair
fall
fan
fast
feeling
few
Figure
file
files
filing
film
final
find
fine
finish
fire
firm
first
fit
fitting
five
fix

flame
flare
flash
flat
flip
float
floor
flow
flush
fly
follow
foot
force
forest
form
foul
four
frame
fraud
freeze
French
fresh
fret
frighten
frightened
front
frozen
full
funding
gain
game
garage
garlic
gate

generation
genius
get
getting
glass
goal
gold
good
grade
grain
grave
gray
great
greed
green
grip
ground
guard
guide
gum
hack
hamper
hand
handle
hang
harbor
hard
hatch
head
heat
heave
hide
High
hire

hit
home
horn
hot
hunt
hurt
import
indoor
instruction
instructions
inter
iron
issue
jack
jail
jam
jerk
job
joint
joints
just
key
kick
kid
kind
kiss
knee
knees
knife
know
laid
land
language
lash

last
late
latest
launch
law
lay
lead
Leave
leaves
left
leg
letter
level
library
lie
life
lift
light
lighter
like
line
link
liquid
little
live
liver
load
loan
local
lock
locked
log
long
look

looking
loom
loop
lose
loss
lost
loud
love
lovely
luck
lucky
made
major
make
makes
making
man
march
mark
marriage
married
master
match
mate
matter
mean
measure
medical
meeting
migrant
mill
mind
mine
minor

mint
minute
miss
mix
mixture
model
mold
momentum
monitor
moon
mortgage
mouse
mouth
move
murder
murk
muslim
nail
navy
need
negative
net
network
new
news
next
nib
nine
node
nose
note
notice
nursing
nut
odd
off
offer
official
ok
old
olive
One
open
operation
order
out
outdoor
outlet
outline
outside
over
pack
package
pad
page
paint
pale
palm
paper
papers
park
particle
particles
pass
passage
pastry
patch
patient
peddling
pen
period
physical
pick
pie
pilot
pitch
plane
plant
plate
play
plot
plow
plumbing
point
polar
pole
polish
pool
pop
port
Positive
post
potato
pound
prescription
present
press
pressure
preventive
primary
prime
principal
print
printer
procedure
process
profile
progress
prompt
proof
property
protest
public
pudding
pull
pulse
punch
push
quiet
quite
race
rack
racket
radius
raft
railway
raise
rate
reaction
reading
real
record
recorded
referee
region
register
regret
relationship
relative
Religious
remote
rent
reply
report
rescuer
research
rest
resume
revenge
rice
ride
right
rock
role
roll
rollicking
roof
rose
rough
route
row
rubber
run
rupee
rust
rye
sad
safety
saw
scale
scare
scratched
screen
seal
seat
second
secretary
secure
see
selection
send
sense
service
set
seven
sewer
sheet
shelter
shift
shine
shoot
show
sick
side
sign
silver
six
smashing
smile
smiling
smoking
smooth
socket
sore
sound
sour
space
speech
spoon
spring
stable
stand
star
start
starts
state
statement
step
stick
stomach
stop
story
straight
strike
string
study
stuff
stumbled
successor
suddenly
suicide
suit
support
surface
surplus
swat
swim
swimmer
switch
table
tail
take
tanned
tea
teaching
tear
technology
television
ten
tenant
terminate
terminates
Terrible
terrorist
theoretical
three
throw
tie
tightly
tired
today
tonight
track
train
transmission
travel
treat
trouser
trousers
trust
turn
twelve

twinkling
two
unclear
understand
uneven
unlawful
use
very
video
virus
visit
volleyball
voltage
wait
waiting
walk
war
wash
waste
watch
watching
way
weight
well
wet
wheat
wire
withdraw
witness
work
worked
working
writing
wrong
yard

Appendix 2: Related Words in PolyWordNet

The related words can be both single-sense words as well as multi-sense words. The following are the sample related words used in PolyWordNet:

> Account, Loan, Deposited, Cash, Software, Balance, Interest, Rate, River, Sea, Walking, Boat, Pitch, Treble, Voice, String, Guitar, Play, Eat, Nutritious, Delicacy, Net, Caught, Detecting, Nocturnal, Diurnal, Hunt, Sleep, Day, Sensing, Cave, Live, Cricket, Played, Player, Beat, Shelf, Read, Writing, Author, Ticket, Room, Hotel, Sun, Dry, Face, See, Teeth, Child, Study, Mark, Future, Vegetable, Field, Farmer, Grows, Income, Photo, Hair, Nail, Picture, Skirt, Cheap, Sell, Goods, Won, Prize, Deal, Exam, Result, School, College, Seem, Stick Depth, Inch, Garden, Length, Width, Wide, Long Height, Dirty, Pain, Left, Right, Kick, Ball, Injured, Hill, Bed, Stair, Chair, Matter, Water, Job, Application, Fill, Filling, Cap, Helmet, Big, Small, Operation, Department, Company, Authority, Organization, Place City, Nepal, Pokhara, Tell Spoke, Habit Children, Hide, Mistake, Bulb, Car, Bag, Lift, Fire, Cigarette, Lighter, Stay, Hostel, Student, Be, People, Draw, Shape Polygon, Rectangle, Pencil, Ticket, Match, Exhibition, Class, Course, Subject, Player, Nice, Scored, Opponent, Game, Experimental, Scientist, Verify, Theory, Ground, Fast, Fell, Race, Slowly, Weight, Reduction, Horse, Sleeping, Ladder, Price, Circumstance, Market, Chemical, Reaction, Compounds Element, Stronger, Ordered, Cabin, Sitting, Placed, Office, Shirt, Wear, Wears, Mirror, Relation, Achieve, Goal, Pressure, Life, Victim, Thief, Asylum, Rope, Traveling, Waiting, Platform, Stop, Station, Robbery , Network, Traffic, Chef , Animal, Dog, Cramming, Good, Same, Living, Batting, Bus, GPS, Dhampus, Come, Go, Kathmandu, Munich

Appendix 3: Test Data (Test Sentences)

There are altogether 4105 Test sentences (Test Data) which are used to test the system in this research work. However, this appendix contains only 180 test sentences as a sample.

1. I have bank account.
2. Loan amount is approved by the bank.
3. He returned to office after he deposited cash in the bank.
4. They started using new software in their bank.
5. He went to bank balance inquiry.
6. I wonder why some bank have more interest rate than others.
7. Spending time on the bank of Kaligandaki river was his version of enjoying in his childhood.
8. She has always dreamed of spending a vacation on a bank of Caribbean sea.
9. He is waking along the river bank.
10. The red boat in the bank is already sold.

11. The ultrasonic sensor has its working principle similar to detecting obstacle by a bat.
12. Does diurnal bat exist or is it that each of them is nocturnal?
13. A bat hunts food and eats at night, but sleeps during the day.
14. I still have my first bat with which I played cricket.
15. Each player on the team has his own bat.
16. Cramming the night before exam is surely not a good way to study.
17. No two people have exactly the same way of living.
18. Each batsman has their own way of batting.
19. I have to change two bus in my way to college.
20. Turn on your GPS to find your way back to home.

21. I lost my phone on the way to Dhampus.
22. I used a paper form for job application.
23. We need to fill form to get sim card from telecom operator.
24. You can fill up the sign up form of the facebook page.
25. Filling this form is quite interesting.
26. We used to stay at line to take dinner at hostel.

27. Students in school are always advised to be in line wherever they go together within the school premises.
28. Try to make a polygon from these given lines.
29. Hari have drawn a line to make rectangle.
30. It is all about the bass, no treble.

31. The bass of female voice is different than that of male voice.
32. My bass string broke.
33. Most bass *guitar* strings are made of nickel-wrapped steel.
34. Some bass has Cobalt strings.
35. I quite like to eat bass.
36. The bass is very nutritious for the person suffering from heart diseases.
37. Bass is a North American delicacy.
38. That stick is two foot long.
39. The depth of well is about 12 foot.
40. She is five foot two inch.

41. Your foot looks so dirty.
42. Ouch! pain on my left foot.
43. Just use your foot while you kick a ball.
44. Just look at the foot of the hill.
45. The foot of the bed seem so weak.
46. She was found murdered at the foot of the stair.
47. Bikram scored 3 run.
48. She did so fast to make more runs and to lead the opponent.
49. In the experimental run, the scientist succeeded to verify his theory.
50. The Whiskers won by a single run.

51. They always run on ground in morning.
52. I hope this run gives a better result.
53. We need energy to run fast.
54. He fell down as he just started to run.
55. The light is so shining.
56. The newly brought light bulb is so bright.
57. One requires a bright day to dry it.

58. This child is exceptionally bright in study.
59. The marks that one secure in exams may not reflect how bright the child is.
60. Just look at face of that girl, so bright.

61. Our company has a bright future ahead.
62. The students of nanotechnology have very bright future in our country.
63. Future of that kid is bright.
64. I wear a cap on my head.
65. Protect your head by using helmet while driving two-wheeler.
66. Does my head look too big?
67. Who is the head of the operation here?
68. Our head of department just resigned.
69. The most authority and accountability of an organization is reside on its head.
70. The horse is in the stable.

71. Do not worry, the ladder is stable.
72. Most of the intermediate product in chemical reaction are not stable product.
73. They crop vegetable in their field.
74. I want to crop this photo.
75. She want her hair to crop.
76. Every Sunday, student crop their nail.
77. That black tie suits for this shirt.
78. They tie up with new relation.
79. The two Universities tie up for to achieve their common goals.
80. India tie with European Union to pressure Nepal on its recently released constitution.

81. Tie the victim so that he cannot escape.
82. I do not need pass ticket to watch the match.
83. Student need to pass final exam to be upgraded to upper class.
84. The player gave a nice pass to another.
85. I won first prize at the fair.
86. Course book might not be available in book **fair**.
87. Nepal wants fair deal on its hydropower project.
88. A horse is tied in the stable.

89. Although there is fluctuation in economy of Nepal, the price of sugar is stable over 2 years.
90. Some chemical compounds are very stable.

91. Farmer grows crop for survival.
92. We usually crop the pictures when we do not want the viewer to see the unwanted or undesirable part within the picture.
93. He usually wears tie on parties and meetings.
94. I feel awkward to wear a tie.
95. Yesterday's match got tie up with 2-2 score.
96. It is against the humanity that the patients are still tie up at asylum when they behave violent.
97. I like to visit book fair because we get cheap books there.
98. Country's fair is a forum for country's people to sell their goods and a entertainment.
99. Tribhuvan University is renowned for its fair exam.
100. The result does not seem fair.

101. Traveling by train is safe.
102. I am waiting for next train
103. Our train will come in this platform.
104. This train will stop in next station.
105. Train robbery has become a major problem in India.
106. Train him as a chef.
107. I train her to take over my job when I retire.
108. The shelf in his room are filled with book.
109. That one is the best book I have ever read.
110. Book the air ticket as soon as possible.

111. Book the 4 room in the hotel.
112. The glasses lie on the table.
113. A lie he speak proved to be a bottleneck for his relation.
114. Children sometimes lie her mother to hide their mistake.
115. Children never lie.
116. It is so bright to see.
117. The head **light** of the car was broken down.

118. This book is light than another.
119. The bag he is carrying is light.
120. Light up the fire.

121. Light up the cigarette with my lighter.
122. All the metals are organized in a periodic table.
123. The age of all the students was recorded in a table.
124. That table seems stronger.
125. A new table was ordered for the cabin.
126. He uses a table while he study.
127. A copy placed on the table is mine.
128. You have to deposit certain percentage of your salary in the bank.
129. Bank of a river is very pleasant place to enjoy.
130. Bat can identify the obstacles by sensing the reflected signals.

131. He gave his cricket bat to his friend.
132. The right way to come Kathmandu is from Muglin.
133. You can fill up the form online.
134. There is huge line of people along the road.
135. Draw the straight line with pencil.
136. He took loan from a Bank.
137. He was sitting on sea bank with his friend
138. Many bats are living in this little cave
139. Beat the thief with the bat
140. It is not the right way to do your home work

141. The way to Munich is not easy
142. Water can be found in different form.
143. You have to fill the form before you enter in room.
144. You have to wait long to get application form.
145. Please draw a line in your note book.
146. He can play the bass.
147. I caught a bass in my net.
148. His garden is 100 foot long and 50 foot wide.
149. His foot was injured.

150. The foot of the chair is broken.

151. If you run slowly, it does not affect on weight reduction.
152. By using a new tooth paste his teeth became bright.
153. He is a bright child not just in history but in all subjects of sciences
154. Just doing well in studies is not a surety of bright future.
155. His head and leg were injured in a car accident.
156. The head of my new company is a nice man
157. He owns a horse stable.
158. He is not stable after the new circumstance.
159. The price in the market is stable for the last year
160. Some elements are stable in chemical reaction.

161. Crop can be a good source of income for the villagers.
162. Crop her skirt.
163. It is difficult to wear a tie without using mirror.
164. Tie up your life with nice girl.
165. Tie with the rope even though it is not a good support.
166. Did u get the pass for the exhibition?
167. They pass the course.
168. Pass the ball to next player in an eye blink.
169. His horse won a Gold medal in city fair.
170. The result of 9 grade was not fair.

171. You have to train your horse before the race.
172. A big network of train is required to overcome traffic need in cities.
173. He is an author of this book.
174. Book a hotel for a week.
175. The river lie in the mid of the city.
176. Pokhara lie in Nepal.
177. Light is combination of seven different color.
178. it is light to lift
179. The data in the table.
180. The design of his new office table.

Appendix 4: List of Publications

1. Udaya Raj Dhungana and Subarna Shakya, "PolyWordNet: Analogous to Human Mind for Word Sense Disambiguation," in *International Journal of Computing and Digital System. Vol 9 No. 5*, September, 2020, pp 835-850.

2. Udaya Raj Dhungana and Subarna Shakya, "Word sense disambiguation using PolyWordNet," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 2, 2017, pp. 1-6.

3. Udaya Raj Dhungana and Subarna Shakya, "Polywordnet: A lexical database," in *Inventive Computation Technologies (ICICT), International Conference on*, vol. 2, 2017, pp. 1-7.

4. Udaya Raj Dhungana and Subarna Shakya, "Hypernymy in WordNet, Its Role in WSD and Its Limitations," *International Journal of Simulation, Systems, Science & Technology*, vol. 16, no. 6, December 2015.

5. Udaya Raj Dhungana and Subarna Shakya, "Hypernymy in WordNet, Its Role in WSD, and Its Limitations," in *The 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, Riga, 2015, pp. 15-19.

6. Udaya Raj Dhungana, Subarna Shakya, Kabita Baral, and Bharat Sharma, "Word Sense Disambiguation using WSD specific WordNet of polysemy words," in *Semantic Computing (ICSC), 2015 IEEE International Conference on*, California, USA, Feb, 2015, pp. 148-152.

7. Udaya Raj Dhungana and Subarna Shakya, "Word sense disambiguation in Nepali language," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on*, Bangkok, Thailand, May 2014, pp. 46-50.