Automatic Taxonomy Expansion in IndoWordNet

(With special reference to Assamese WordNet)

Thesis submitted in partial fulfilment of the requirements
for the award of the degree of

Doctor of Philosophy

in

Centre for Linguistic Science and Technology

by

Bornali Phukon

Under the supervision of

Dr. Sanasam Ranbir Singh and Prof. Priyankoo Sarmah

Centre for Linguistic Science and Technology
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

May , 2023

Dedicated to

My beloved family

Acknowledgment

First and foremost, I would like to express my sincere gratitude to my esteemed supervisors, Dr. Sanasam Ranbir Singh and Prof. Priyanoo Sarmah for their invaluable contributions to my doctoral research. Their unwavering support, patience, and guidance have been instrumental in the successful completion of my research work. Throughout the entire research journey, their continuous encouragement, positive feedback, and insightful suggestions have helped me to grow not only as a researcher but also as an individual. I am forever grateful for the opportunity I had working with Them and always be remain indebted.

I am thankful to my thesis doctoral committee members - Prof. Sukumar Nandi , Dr. Samit Bhattacharya Prof. SamudraVijaya K, and Dr. Abhishek Shrivastava for their percipient comments and suggestions that help in enhancing the quality and clarity of my work. I would like to acknowledge the heads of the centre of CLST during my Ph.D. journey at IITG - Prof. Sukumar Nandi and Prof Rohit Sinha for providing me with the facilities and resources. I appreciate the staff of the Centre For Linguistic Science and Technology - Souvik Chowdhury and Mr. Raktajit Pathak for their constant support on any engineering and official-related issues. I am obliged to all the faculty members, the staff, and security personnel for their constant help and support. I am thankful to all my colleagues and friends during my journey as a Ph.D. scholar. I am indeed thankful to my fellow lab mates at the CLST family - Porishmita, Dhrubojyoti, Shikha, Seema, Jennil, Hemanta, Supriya, Pankaj, Chaitanya, Lipsa, Moumita, Jayshree, Meghali and many more for creating a wonderful experience at my workplace. The thought-provoking debates, collaborative problem-solving, and teamwork have a big impact on how I develop as an independent and motivating researcher.

I am fortunate to have several wonderful friends within and outside IIT Guwahati who were instrumental in supporting me during my Ph.D. endeavor. Life never has a dull moment with them. I would especially like to thank Jennil, Seema, Hemanta, Shikha, Bidisha, Gyanendro, Anasua, Akash, Neelakshi, Prarthona, and Sujit. I have shared some indelible moments with you all. I am the one who wants to go out of the box and my comfort zone to explore more, and they are the people who always support me, especially Jennil, Seema, Anasua, Hemanta, and Akash. I am deeply grateful to Jennil and Seema, who has been my unwavering companions throughout my academic and personal life. Their constant presence and support have been a source of great comfort to me, particularly during the ups and downs of my Ph.D. journey. I could always count on them to be available whenever I needed them, and they have consistently offered me invaluable advice and guidance. I would like to express my sincere appreciation and gratitude towards all members of the CLST and OSINT family. The collective support and guidance from this community have been instrumental in my personal and academic growth. Your contributions have played a significant role in shaping me into the individual I am today, and I am forever grateful for your unwavering support. Thank you for being an integral part of my journey, and I look forward to continued collaborations with this community.

I would like to thank my favorite set of people, the pillars of my strength — my parents (Mrs. Renu raj kumari Phukon and Ajit Phukon), my in-laws(Mrs. Punya Phukon Gogoi and Rupen Gogoi), my brother and brothers-in-law (Dayananda Phukon, Aniruddha Gogoi, Parth Sarathi Gogoi, Ankur Dhekial Phukan, Prakritish Buragohain), my sister and sisters-in-law (Mridumoni Phukon, Sabita Borah, Uttara Buragohain, Manalima Borpattra, Trishna Saikia) and our little angles (Arlen, Anushree, Aradhya, Aaron,

Tejaswani, Aaralin, Abhigya, Iravaan and the newest addition to our family, little Driti). I am who I am today because of your boundless love, support, caring, warmth, and encouragement over all these years. I am truly indebted to them.

Finally, I would like to express my profound gratitude to the three pillars of support who have lifted me and kept me going through thick and thin. First and foremost, I am eternally grateful to my beloved husband, (Dr. Parikshit Gogoi), who has been the backbone of my support throughout this journey. Without his constant encouragement, this feat would not have been possible. I consider myself incredibly fortunate to have him in my life, and his unwavering support has been the driving force behind my success. To my little bundle of joy (Aaron), I owe an immeasurable debt of gratitude. He has been my source of inspiration and joy, and his infectious laughter and unconditional love have been the driving force behind every step I have taken. His presence has brought light to even the darkest of days, and I feel blessed beyond words to be his mother. Last but not least, I cannot thank my sister (Mridumoni), enough for being my constant source of support and encouragement. She has been my constant companion, always there for me through both the highs and lows of this journey. Her presence in my life has been invaluable, and I feel incredibly fortunate to have her as my sister.

I am grateful to the esteemed institution, IIT Guwahati, for providing a serene campus with high-quality facilities that facilitated my academic pursuits. Last but not least, I extend my gratitude to the medical staff, security personnel, and cleaning staff of IIT Guwahati for their valuable contributions towards making my journey here as comfortable and safe as possible.

February 23, 2023 Bornali Phukon

Declaration

I certify that

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisors.
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisors are not in a position to check for any possible instance of plagiarism within this submitted work.

February 23, 2023 Bornali Phukon

Centre for Linguistic Science and Technology
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

Certificate

This is to certify that this thesis entitled "Automatic Taxonomy Expansion in IndoWordNet(With special reference to Assamese WordNet)" submitted by Bornali Phukon, in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy, to the Indian Institute of Technology Guwahati, Assam, India, is a record of the bonafide research work carried out by her under my guidance and supervision at the Centre for Linguistic Science and Technology, Indian Institute of Technology Guwahati, Assam, India. To the best of my knowledge, no part of the work reported in this thesis has been presented for the award of any degree at any other institution.

Date: February 23, 2023
Place: Guwahati

Dr. Sanasam Ranbir Singh
(Coordinating supervisor)
Associate Professor
Dept. of C.S.E
IIT Guwahati

Prof. Priyankoo Sarmah
(Co-supervisor)
Professor
Dept. of HSS
IIT Guwahati

Automatic Taxonomy Expansion in IndoWordNet

(With special reference to Assamese WordNet)

Abstract

The task of automatic taxonomy expansion plays a significant role in natural language processing (NLP), as it helps to overcome the issue of low coverage in taxonomies. By effectively performing this task, various NLP applications like information retrieval, text classification, and natural language understanding can achieve better accuracy and efficacy. While numerous studies have explored the challenges of automatic taxonomic expansion, the methods and techniques used in these studies may be less effective for taxonomies like WordNet due to their unique structure and organization.

WordNet is a widely used lexical taxonomy of concepts in a language that comprises not only a hierarchical organization of concepts but also information regarding other semantic relations such as synonymy, meronymy, and troponymy among the concepts, which distinguish it from other taxonomies. The creation of WordNets typically involves manual methods; however, currently, a substantial number of WordNets are generated through the expansion approach, such as those included in Indo-WordNet. Despite its widespread usage, creating WordNet is challenging, with two significant problems being *limited coverage* and *missing relations*. The manual creation process of WordNets can result in limited coverage, while the use of the expansion approach for creating WordNets may result in missing relations between concepts and words. While previous studies have sought to address the issue of limited coverage, the problem of missing relations has yet to receive adequate attention. Furthermore, while automatic taxonomy expansion approaches have been proposed to resolve the issue of limited coverage, their effectiveness for WordNet expansion remains in question. The primary reason is that the expansion of WordNet requires not only inserting new concepts (*attach operation*) but also extending existing ones (*merge operation*) as shown in figure 1.4. However, most existing studies on taxonomy expansion only focus on the (*attach operation*). Furthermore, WordNet taxonomies, especially those in Indian languages, tend to have a multi-root structure. It makes it more challenging to utilize traditional methods for the expansion of WordNet taxonomy as these methods are not designed to handle the challenges of a multi-root structure, which may limit their usefulness in expanding WordNet taxonomies. In light of these challenges, this thesis work aims to address the problem of automatic taxonomy expansion by addressing the challenges in WordNet, especially in IndoWordNet. The objective is to develop a solution that can be extended to other taxonomies beyond WordNet.

This thesis first studies the problem of missing synonymy relations in WordNet taxonomy. It considers Assamese Wordnet as a case study. It investigates the effectiveness of Link prediction methods. As WordNets can be visualized as a network of unique words connected by synonymy relations, link prediction in complex network analysis is an effective way of predicting missing relations in a network. Hence, in order to predict the missing synonyms in the Assamese WordNet, link prediction methods were used in the current work that proved effective. It is also observed that for discovering missing relations in the Assamese WordNet, simple local proximity-based methods might be more effective as compared to global and complex supervised models using network

embedding. Second, a novel multi-task learning-based deep learning method known as *Taxonomy Expansion with Attach and Merge (TEAM)* is proposed, which performs both the *merge* and *attach* operations. This is the first study that integrates both the*merge* and *attach* operations in a single model to the best of our knowledge. The proposed models have been evaluated on three separate WordNet taxonomies, viz., Assamese, Bangla, and Hindi. From the various experimental setups, it is shown that TEAM outperforms its state-of-the-art counterparts for *attach* operation and also provides highly encouraging performance for the *merge* operation. Third, As TEAM considers local context, it faces challenges when it is applied to multi-root taxonomies. To address the limitations in TEAM, this thesis proposes another approach, LG-TEAM, which combines both the local and global context of taxonomy in an integrated *attach-merge* expansion environment, providing a more robust solution to the problem of taxonomy expansion. Extensive experiments on English, Assamese, Bengali, and Hindi WordNets demonstrate both the effectiveness and the efficiency of LG-TEAM for automatic taxonomy expansion.

Contents

List of Figures

List of Tables

List of Abbreviations

Terms	Abbreviations
AA	Adamic Adar
AUC	Area Under ROC Curve
CL	Classification
CN	Common Neighbors
CNN	Convolutional Neural Network
DGL	Deep Graph Library
DL	Deep Learning
DNN	Deep Neural Network
EAW	Extended Assamese WordNet
GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
HITS	Hyper-link induced topic search
JC	Jaccard Coefficient
LINE	Large scale Information Network Embedding
MAP	Mean Average Precision
MLP	Multi-Layer Perception
MR	Mean Rank
MRR	Mean Reciprocal Rank
MTL	Multi-Task Learning
NLP	Natural Language Processing
RN	Resource Allocation
RG	Regression
SDNE	Structural Deep Network Embedding
TEAM	Taxonomy Expansion approach for Attach and Merge
LG-TEAM	Local and Global context aware TEAM
OOV	Out-Of-Vocabulary
VERSE	Versatile Graph Embeddings from Similarity Measures

"You can't go back and change the beginning, but you can start where you are and change the ending."

— CS Lewis

1 Introduction

1.1 Overview

Lexical taxonomies are essential resources for natural language processing (NLP) as they provide a systematic and standardized method of organizing words, concepts, and phrases. These taxonomies offer an organized hierarchy of terms or concepts and can be used to enhance the performance of a wide range of NLP tasks. Some key applications of lexical taxonomies in NLP include –

- Dealing with unstructured text: Lexical taxonomies can be utilized to organize unstructured text data, resulting in more efficient and effective extraction of information from the text rather than working directly with raw unstructured data (as shown in Figure 1.1).

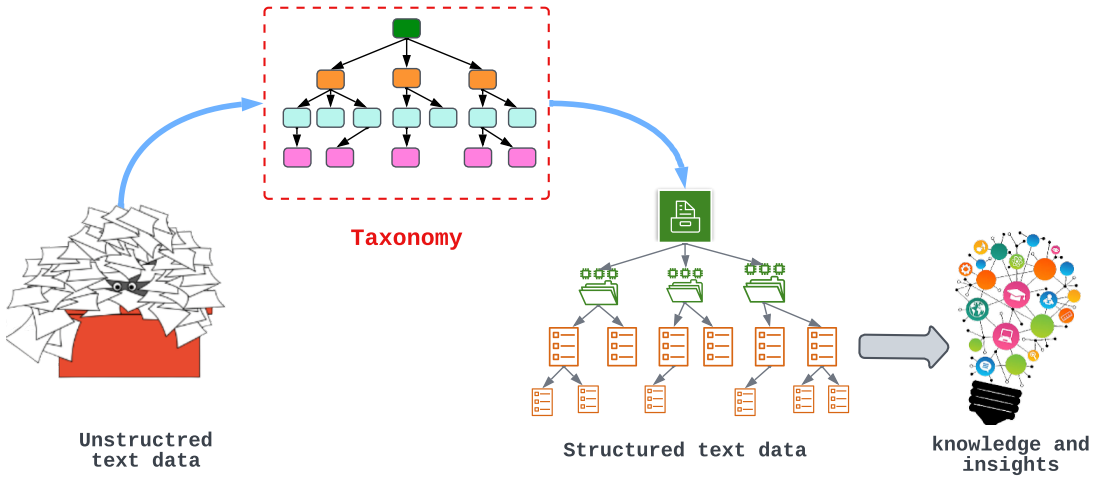


Figure 1.1: Utilizing taxonomy structure to turn unstructured text data into actionable knowledge

- Information retrieval : Lexical taxonomies can also be used in information retrieval, allowing for more precise and relevant search results by utilizing the structured hierarchy of terms and concepts.
- Text classification : By utilizing a lexical taxonomy, text can be organized and classified into specific categories or labels, improving the accuracy and efficiency of text classification tasks.

Manual creation of taxonomies is a time-consuming, expensive, and non-scalable process, which inherently limits their coverage. A low-coverage taxonomy can lead to a variety of problems in NLP tasks, including incomplete text understanding, limited usability, etc. As a result, NLP tasks such as information retrieval and text classification may perform poorly. Also, since new concepts are constantly emerging and growing, they may need to be added to existing taxonomies in order to keep them up-to-date and useful. Therefore, automated taxonomy expansion is crucial to effectively addressing these issues.

Many well-known taxonomies have been introduced by researchers in different domains, such as WordNet[51], MesH[43], Pinterest Taxonomy[21] (Figure 1.2). Among these taxonomies, WordNet is a widely used lexical taxonomy that provides a comprehensive and structured representation of the meanings of words in a language that could be exploited in more complex NLP research and applications. It organizes words into synsets (sets of synonyms) and defines relationships between them, such as hypernymy,

Figure 1.2: Few examples of taxonomy

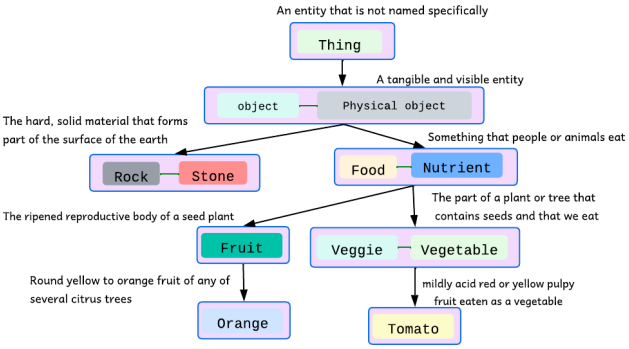


Figure 1.3: Structure of English WordNet taxonomy with synonymy and hypernymy relations

hyponymy, meronymy, and troponymy. Figure 1.3 shows a snippet of the English WordNet with its synonymy and hypernymy relations. This hierarchical structure allows for the representation of rich semantic information, making it a valuable resource for various NLP tasks. As mentioned above, the creation of WordNets typically involves manual methods; however, currently, a substantial number of WordNets are generated through the *expansion* approach[7]. This approach involves creating new WordNets through the mapping of a pre-existing WordNet, specifically those included in the Indo-WordNet. Figure 2.1 presents a high-level view of the *expansion* approach to WordNet creation from Hindi WordNet to Assamese WordNet.

Despite its widespread usage, creating WordNet is challenging, with two significant problems being *limited coverage* and *missing relations*. The manual creation process of WordNets can result in limited coverage, while the use of the *expansion* approach for creating WordNets may result in missing synonymy relations between words. A detailed analysis of these challenges are discussed in Section 3.1. While previous studies have sought to address the issue of limited coverage, the problem of missing relations has yet to receive adequate attention. Furthermore, while automatic taxonomy expansion approaches have been proposed to resolve the issue of limited coverage, their effectiveness for WordNet expansion remains in question. The primary reason is that the expansion of WordNet requires not only inserting new concepts (*attach operation*) but also extending existing ones (*merge operation*) as shown in figure 1.4. However, most existing studies on taxonomy expansion only focus on the *attach* operation [74,90,78,104,105,85,44]. Furthermore, WordNet taxonomies, especially those in Indian languages, tend to have

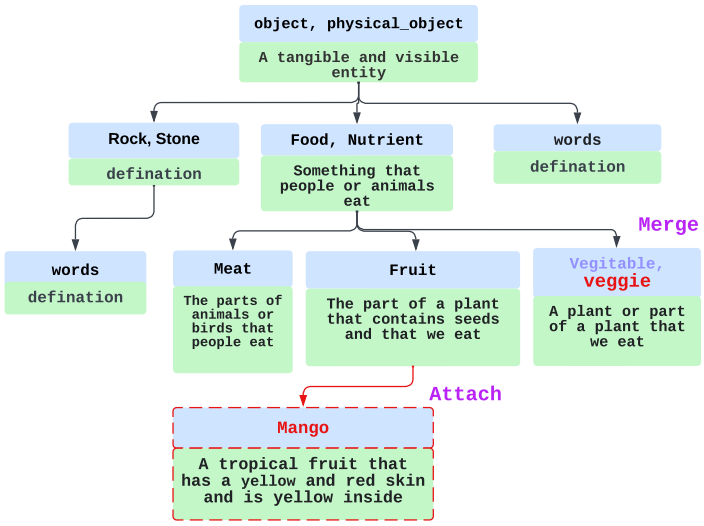


Figure 1.4: Expansion of English WordNet through Attach and Merge operations

a multi-root structure. It makes it more challenging to utilize traditional methods for expansion of WordNet taxonomy as these methods are not designed to handle the challenges of a multi-root structure, which may limit their usefulness in expanding WordNet taxonomies. These challenges highlight the ongoing need for further research to enhance the robustness and reliability of WordNet. Thus this thesis aims to address the problem of automatic taxonomy expansion, with a specific focus on the challenges inherent in WordNet, particularly in IndoWordNet. The objective is to devise a solution that can be adapted to other taxonomies beyond WordNet and enhance the overall accuracy of automatic taxonomy expansion. The research goal is to advance the field of automatic taxonomy expansion by addressing the limitations and challenges present in WordNet and to create a scalable and versatile solution for expanding taxonomies.

An overview of IndoWordNet : IndoWordNet is a comprehensive multilingual lexical resource covering 18 different Indian languages, which includes individual WordNets for each language. Among these WordNets, the Hindi WordNet is created manually using lexical knowledge from various dictionaries, whereas the other languages were generated through the *expansion* approach using Hindi as a pivot language. An overview of the *expansion* approach is shown in figure 2.1. This approach leverages the existence of universal concepts that are independent of the language and have consistent semantic relations across languages. As a result, the semantic relations for universal synsets are defined in Hindi and then borrowed by other languages. This approach has proven

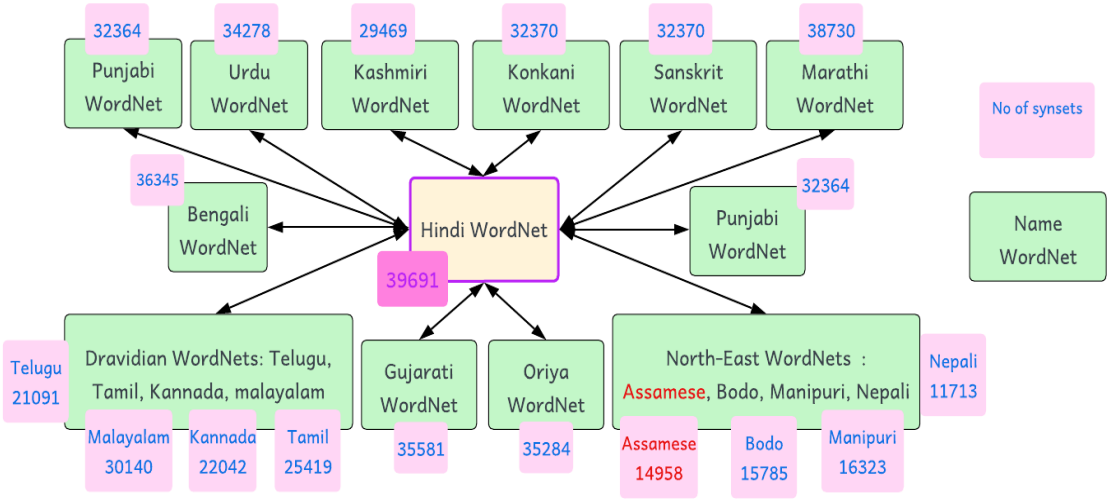


Figure 1.5: Linked IndoWordNet structure and current statistics Source*

effective for closely related languages, such as Hindi and Marathi. However, this approach may not be as effective for languages that are not closely related, such as Hindi and Assamese (a detailed study is given in chapter 3). The current statistics of the IndoWordnet are shown in figure 1.5.

1.2 Research gap

The field of automatic taxonomy expansion is an area of active research, with ongoing efforts to improve and expand existing taxonomies. Despite the progress made in this field, there remain challenges and limitations in expanding taxonomies, particularly in the case of WordNet. These limitations significantly impact the usefulness and applicability of these taxonomies for text-processing tasks. In order to address these limitations and challenges, there is a need for further research to create a scalable and versatile solution for expanding taxonomies. The following are the significant research gaps in the field of automatic taxonomy expansion:

1. **Lack of attention on missing synonymy relations in WordNet :** The development of taxonomies remains a persistent challenge, particularly in the case of the WordNet taxonomy. This taxonomy is unique due to its various lexical and semantic relations, with synonymy playing a crucial role in its creation. However, incomplete or missing synonymy relations can negatively impact the usefulness of WordNet and limit its usability in text-processing tasks. The occurrence of *miss-*

ing synonymy relations within the WordNet taxonomy refers to instances where two words have the potential to be synonyms, but their relationship as such is not explicitly defined within the taxonomy. The existence of missing synonymy relations within WordNet results in limited coverage of the taxonomy. This presents a major challenge for low-resource languages where the only available resource for synonymy is the WordNet synonymy thesaurus. The wordNets created by the *expansion approach* have a tendency to create missing or incomplete relations in the WordNet. This is evident in the case of the Assamese WordNet from IndoWordNet, which was created by mapping the Hindi WordNet to corresponding Assamese equivalent words. For example, the Assamese words ধীৰ *'Dhir'* (patience), শান্ত *'Xanto'* (calm),স্থিৰ*'Sthir'* (quiet) are synonyms with respect to the sense/concept *self composed*. As the sense or concept *self-composed* is not present in the Hindi WordNet, the synonymy relationship between these words is also missing in the Assamese WordNet. Additionally, the manual curation process can also cause missing or incomplete synonymy relations if the annotators' understanding does not accurately depict the associations between words. While previous studies have attempted to address the issue of limited coverage in WordNet, the problem of missing synonymy relations between words has yet to receive adequate attention. In the recent past, the problem of synonymy prediction (or synset expansion) has been studied in the context of extracting entity[69,77], attribute[26], and lexical synonymy words[89]. Though entity synonymy prediction received considerable attention, attribute, and lexical synonymy prediction have been less explored. This thesis aims to address this gap by developing a solution to discover missing relationships and enhance the usefulness of WordNet in various text processing tasks.

2. **Inadequate focus on both *attach* and *merge* operations:** For taxonomy expansion, WordNet, in particular, may need two types of operations; (i) *merge*, where a new concept is merged to an existing node, and (ii) *attach*, where a new concept is inserted as a new node. Figure 1.4 illustrates these two operations, where the word *Mango* is inserted as a new concept with the *attach* operation, and the word *veggie* is inserted as a new synonym in an existing concept with

the *merge* operation. Though both of these operations are integral parts of a WordNet taxonomy expansion, all of the existing studies on taxonomy expansion have considered expansion with either *attach* operation[74,90,78,104,105,85,44] or *merge* operation[55,57,55,69,11,94,19], but not together. Therefore, there is a need for further research to address this issue and develop a comprehensive solution for taxonomy expansion through *attach* and *merge* operations.

3. **Limited research on multi-root taxonomies:** There is a lack of research on expanding multi-root taxonomies, which are taxonomies that have multiple roots as opposed to a single root. However, in cases where WordNet has multiple roots, it poses challenges that need to be addressed when expanding it. This gap in research is being addressed by this thesis, which proposes a solution for the effective expansion of multi-root taxonomies such as the Indian languages WordNet.

1.3 Research Objectives

Taking into account the research gap outlined above, this thesis aims to address the following research objectives:

1. **Addressing the challenges of missing synonymy relations in WordNet taxonomy :** The objective of this research is to address the challenge of missing synonymy relations in WordNet, which restricts its usability in various text processing applications, including information retrieval, information extraction, text classification, and summarization. To resolve the issue of missing synonymy relations, the research focuses on visualizing WordNet as a network of unique words interconnected by synonymy relations. The study employs link prediction techniques in complex network analysis to empirically evaluate the effectiveness of state-of-the-art methods in predicting missing relations. The final goal is to enhance the usefulness of WordNet by resolving the issue of missing synonymy relations and expanding its applications in text processing.

2. **Incorporating both *attach* and *merge* operations for expansion of taxonomy:** The objective of this study is to address the limitations and challenges present

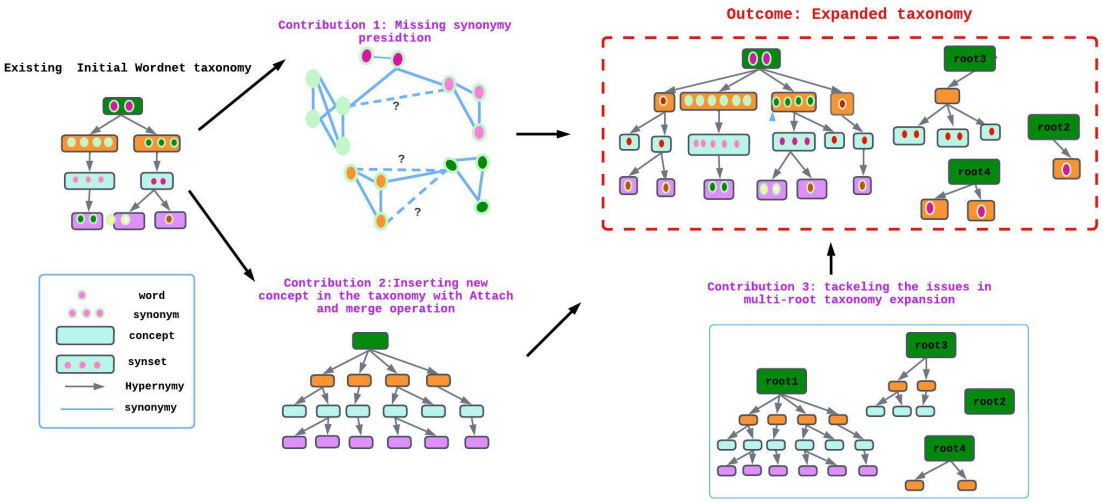


Figure 1.6: visual representation of the contributions made in this thesis and its outcome

in the automatic expansion of WordNet taxonomy by considering both the *attach* and *merge* operations. This research aims to fill the gap left by previous studies, which have only focused on one of these operations. The objective is to propose a comprehensive model that integrates both the *attach* and *merge* operations, resulting in a more comprehensive and effective approach to WordNet taxonomy expansion.

3. **Overcoming limitations in expanding multi-root taxonomies :** Expanding multi-root taxonomies, particularly WordNet for Indian languages, poses certain limitations that need to be addressed. This objective aims to overcome these limitations and provide a solution for expanding multi-root taxonomies in a scalable and flexible manner.

1.4 Contributions

This thesis presents the following significant contributions by addressing the identified gap in research and fulfilling the designated research objectives.

- The first contribution of this thesis is focused on addressing the issue of missing synonymy relations in the WordNet taxonomy, with a specific emphasis on the Assamese WordNet taxonomy. To address this gap, the study revisits the synonymy prediction problem for low-resource languages and carries out a thorough analysis of the characteristics of synonymy relations in the Assamese WordNet.

The empirical analysis uses complex network analysis methods to predict missing synonymy relations, providing a novel solution to this persistent challenge in the development of taxonomies. This contribution has the potential to make a significant impact in the field of automatic missing synonymy prediction in the WordNet and as a result, advance the field of automatic taxonomy expansion.

- The second contribution of this thesis is the development of a multi-task learning-based taxonomy expansion framework referred as TEAM. This framework is designed to perform both the *attach* and *merge* operations in a single model, making it the first integrated model of its kind. To achieve this, the framework is designed using two variants, namely TEAM-Regression (RG) and TEAM-Classification (CL). The integration of both operations in a single model provides a more scalable and versatile solution for taxonomy expansion, as compared to previous methods which only focused on either *attach* or *merge* operations. The multi-task learning approach used in TEAM enables it to learn from both *attach* and *merge* tasks simultaneously, providing a more comprehensive and effective solution for WordNet taxonomy expansion.

- The third contribution of this thesis involves proposing a novel approach, LG-TEAM, to enhance the taxonomy expansion framework, TEAM. Building upon the multi-task learning-based approach of TEAM, which performs both the *attach* and *merge* operations, LG-TEAM expands upon this concept by considering both local and global contexts in the taxonomy expansion process. This is achieved by considering the ego-net to capture the local context and the entire network to capture the global context. By adding a dummy root node that connects all sub-tree roots, LG-TEAM ensures message passing between the sub-trees while capturing the global context. The goal of LG-TEAM is to further improve the effectiveness of automatic taxonomy expansion by considering both local and global contexts, building upon the framework of TEAM.

- Overall, this thesis advances the field of automatic taxonomy expansion by providing comprehensive and practical solutions to expand WordNet taxonomy. The proposed solutions have the potential to be applied to other taxonomies, mak-

ing them a valuable contribution to the field. This will provide a foundation for further research and development in the field of automatic taxonomy expansion.

The schematic diagram presented in Figure 1.6 provides a visual representation of the contributions made in this thesis and the comprehensive outcome of the research effort.

1.5 Organization of the Thesis

The organization of this thesis is comprised of several distinct chapters, each of which serves a specific purpose in supporting the overall argument and structure of the work. The chapters are as follows:

- **Chapter 1. Introduction :** This chapter provides a comprehensive introduction to the concept of taxonomy and its relevance, with a particular focus on the WordNet taxonomy. The chapter then discusses the existing research gap and outlines the research objectives of the thesis. Additionally, the chapter highlights the contributions made by the thesis work.

- **Chapter 2. Background :** The Background study covers a variety of topics related to this thesis. Firstly, the dataset background is discussed, which includes WordNet and IndoWordNet. Following this, network preliminaries are discussed and various approaches to network embedding, including local and global network representation, are explored. Finally, the multi-task learning approach that has been employed in this thesis is presented.

- **chapter 3. Synonymy Expansion Using Link Prediction Methods: A Case Study of Assamese WordNet :** As our first significant contribution, this chapter first discusses the characteristics of synonymy relations in Assamese WordNet and its issues. Then it discusses the empirical analysis of predicting missing synonymy relations in Assamese Wordnet using Link Prediction methods. Further, it discusses the characteristics of the dataset used in this study. It is followed by evaluating the link prediction methods in predicting missing relations in Assamese WordNet.

- **Chapter 4. A multitask learning-based Taxonomy Expansion approach for *attach* and *merge* (TEAM)**. This chapter, introduces the second significant contribution of this thesis work, which is the proposed method of multitask learning-based Taxonomy Expansion approach for *attach* and *merge* (TEAM). The chapter outlines the TEAM approach in detail, including its implementation and evaluation.

- **Chapter 5. Local and Global context aware Multitask learning based Taxonomy Expansion for *attach* and *merge* operations(LG-TEAM) :** This chapter introduces the third significant contribution of this thesis, which is the proposed Local and Global context-aware Multitask learning based Taxonomy Expansion for *attach* and *merge* operations. The integrated taxonomy expansion approach, referred to as LG-TEAM, aims to address the challenges associated with multi-root disconnected taxonomies. The chapter outlines the LG-TEAM approach in detail, including its implementation and evaluation.

- **Chapter 6. Conclusion and Future Work**: This chapter concludes with possible future research directions of this thesis.

There are things known and there are things unknown, and in between are the doors of perception.

Aldous Huxley, English writer

2

Background Studies

This chapter provides a comprehensive overview of the background studies that are relevant to the research presented in this thesis. This thesis focuses on addressing the challenges inherent in the process of WordNet taxonomy expansion and explores various fields that are relevant to the proposed solutions. To provide a thorough understanding of the challenges associated with WordNet taxonomy expansion and to offer a well-supported and grounded solution, this chapter discusses the various fields relevant to this study. The background studies are presented in two directions. First, the background of WordNet and IndoWordNet is discussed, offering a brief overview of the datasets used in this research. Second, the technical background presenting an overview of the techniques, and the methods used in the field of automatic taxonomy expansion, are discussed.

2.1 Taxonomy Background

This thesis employs the IndoWordNet taxonomy as its primary dataset to investigate issues in the automatic expansion of WordNet taxonomy. To provide a foundation for this investigation, this section provides an overview of WordNet, its structure, and its features. Moreover, it discusses the creation of IndoWordNet and highlights the issues associated with it.

2.1.1 WordNet:

WordNet is a comprehensive lexical database consisting of synsets and semantic relations. The synsets in WordNet consist of a single word or a group of words that represent the same concept, and each synset contains definitions that describe the concept using sentences. WordNet includes synsets with words from four different parts of speech categories: nouns, adjectives, verbs, and adverbs. WordNet store various relations among words and synsets. These relations give important knowledge about language structure. These are categorized under two labels, viz., lexical relations and semantic relations. One of the most significant aspects of WordNet is the way in which it stores various relations between words and synsets, which provide valuable information about the structure of language. Lexical relations are those that exist between individual words, and semantic relations are those that exist between entire synsets.

Lexical Relations: Lexical relations refer to the relationships that exist between individual words. These relations can take several forms, including compounds (for nouns), conjunctions (for verbs), gradation (for all parts of speech, which refers to relationships based on factors such as state, size, light, gender, temperature, color, time, quality, action, and manner), and antonymy (for all parts-of-speech, which refers to relationships between words that are opposite in meaning with respect to factors such as action, amount, direction, gender, personality, place, quality, size, state, time, color, and manner).

Semantic Relations: Semantic relations, on the other hand, are the relationships that exist between the synsets themselves. These relations take different forms, such as hypernymy (for nouns and verbs), holonymy (for nouns), meronymy (for a component

object, member collection, feature, activity, place, area, face, state, portion, mass, resource, process, position, and area), troponymy (for verbs), similar attribute (between nouns and adjectives), function verb (between nouns and verbs), ability verb (between nouns and verbs), capability verb (between nouns and verbs), also-see, and adverb modifies verb (between adverbs and verbs).

WordNet is often considered a taxonomy because it organizes concepts based on their hypernymy and hyponymy relationships. Hypernymy refers to the relationship between a more general term (hypernym) and a more specific term (hyponym), while hyponymy refers to the inverse relationship between a specific term and a more general term. This allows WordNet to provide a systematic and structured way to classify and organize concepts and words based on their semantic relationships. This makes WordNet an essential tool for natural languages processing tasks such as information retrieval, text classification, and machine translation, as well as a valuable resource for linguists, lexicographers, and researchers in the field of language study. The first WordNet in the world was built for English at Princeton university ∗. This was followed by the development of WordNets for several European languages, known as EuroWordNet† [92]. Since 2000, there have been efforts to build WordNets for several Indian languages, with Hindi WordNet [29] being the first one to be developed. At present, WordNets have been created for 18 Indian languages on a shared platform, utilizing an expansion approach that considers Hindi WordNet as the source WordNet. The following section provides a brief overview of IndoWordNet.

2.1.2 IndoWordNet

The IndoWordNet [7,13] is a lexical database that encompasses multiple Indian languages, with Hindi WordNet at its core and connections to other language WordNets through expansion. IndoWordNet is the most useful multilingual lexical resource in Indian languages. Hindi WordNet is created manually using lexical knowledge from various dictionaries. Figure 2.1 shows how Assamese WordNet is created from HindiWordNet using the *expansion* approach. Currently, IndoWordNet includes 18 Indian languages, including Assamese, Bengali, Bodo, Gujarati, Kannada, Kashmiri, Konkani, Malayalam,

∗http://www.WordNet.princeton.edu
†http:// http://www.illc.uva.nl/EuroWordNet/

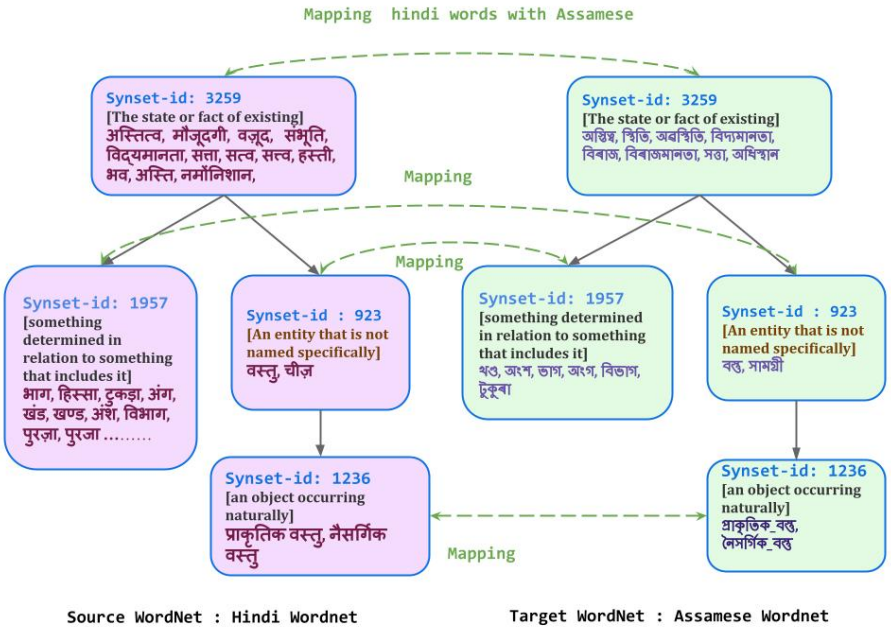


Figure 2.1: The process of WordNet creation by expansion approach

Manipuri, Marathi, Nepali, Odiya, Punjabi, Sanskrit, Tamil, Telugu, and Urdu.

2.2 Technical Background

This section presents an overview of the methods and approaches that are related to and employed in the present research. The trend towards utilizing network-based solutions in the field of complex systems modeling has become increasingly prevalent, and this research seeks to make a contribution to this trend by proposing a network-based approach to address the problems under investigation. This thesis studies network-based solutions to address the challenges inherent in WordNet taxonomy. This section provides an introduction to the concept of a network, including the definition of a network and the representation of WordNet as a network. A comprehensive overview of network representation is also presented, offering the necessary context and understanding of the proposed network-based solution and its significance

2.2.1 Network preliminaries:

Definition: A network (or graph) G is defined as an ordered pair (V, E), where V is a set of vertices (or nodes) and E is a set of edges. Each edge $e \in E$ is a pair of vertices

$(u, v) \in V \times V$, representing a connection between nodes u and v in the network. The set of neighbors $N(u)$ of a vertex u is defined as the set of vertices that are connected to u by an edge, i.e., $N(u) = \{v \in V \mid (u, v) \in E\}$.

Representing WordNet as a Network: WordNet can be represented as a network in several ways, depending on the relationship between words or concepts. Following are few common ways to represent WordNet as a Network.

1. WordNet as a Network of synsets: In this representation, Each synset S_i is represented as a node, and the relationships between synsets such as hypernymy (*hyper*) and hyponymy (*hypo*) are represented as directed edges. The resulting graph can be defined as $G = (V, E)$, where $V = S_1, S_2, \ldots, S_n$ is the set of nodes representing synsets, and $E = (S_i, S_j) \mid S_i \xrightarrow{hyper,hypo} S_j$ is the set of edges representing the directed relationships between synsets.

2. WordNet as a Network of words: In this representation, Each word w_i is represented as a node, and the relationships between words such as synonymy (s) or antonymy (a) are represented as edges. The resulting graph can be defined as $G = (V, E)$, where $V = w_1, w_2, \ldots, w_n$ is the set of nodes representing words, and $E = (w_i, w_j) \mid w_i \xrightarrow{s,a} w_j$ is the set of edges representing the relationships between words.

3. WordNet as a Homogeneous Network: In a homogeneous network, the graph $G = (V, E)$ focuses on a single semantic relation between concepts or words, such as hypernymy or hyponymy. For example, a hypernymy network would have nodes representing synsets and edges representing the hypernymy relationship between them: $E = (S_i, S_j) \mid s_i \xrightarrow{h} S_j$.

4. WordNet as a Heterogeneous Network: In a heterogeneous network, the graph $G = (V, E)$ incorporates multiple semantic relations to provide a more comprehensive view of the relationships between concepts or words. For example, a heterogeneous network could have nodes representing both synsets and words, and edges representing different semantic relations such as hypernymy, hyponymy, synonymy, or antonymy: $E = (S_i, S_j) \mid S_i \xrightarrow{hyper,hypo} S_j \cup (w_i, w_j) \mid w_i \xrightarrow{synonym,antonym} w_j$, where S_i and w_i denote the ith synset and word, respectively.

2.2.2 Network Embedding

Network embedding is a technique used to learn a low-dimensional vector representation of nodes in a network while preserving certain structural properties of the network. Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the goal of network embedding is to learn a function $f : \mathcal{V} \rightarrow \mathbb{R}^d$ that maps each node in the network to a d-dimensional vector representation. This is done by encoding information about the structure of the network in the vector representations. By reducing the dimensionality of the input data while retaining structural information, network embedding can be used for a variety of downstream tasks, such as node classification and link prediction in the network. Network embedding has become a popular tool in the field of network analysis and has been successfully applied in various domains, such as social network analysis, recommender systems, and natural language processing.

There are several network embedding techniques, each with its own strengths and weaknesses. Some of the most commonly used network embedding techniques are:

1. DeepWalk[63]: This random walk-based technique generates node sequences by sampling random walks from the network. The node sequences are then used as input to a Skip-gram model, which is a neural network language model, to learn node embeddings.

2. Large-scale Information Network Embedding (LINE)[86]: This is a graph-based embedding technique that models both first-order and second-order proximity between nodes. It minimizes the KL divergence between the observed node-node similarities and the similarities derived from the learned node embeddings.

3. node2vec[22]: This is a random walk-based embedding technique that allows for the exploration of diverse neighborhoods by controlling the random walk bias. It generates node sequences by sampling biased random walks from the network and learns node embeddings using a Skip-gram model.

4. Structural Deep Network Embedding (SDNE)[93]: This is a deep learning-based technique that uses autoencoders to learn node embeddings. It explicitly models the high-order proximity between nodes by reconstructing the adjacency matrix

in the embedding space.

5. Graph Convolutional Network (GCN)[34]: This is a graph neural network-based technique that applies convolutional operations on the graph structure to propagate information and learn node representations.

The network embedding techniques used to represent nodes in a graph or network utilize two commonly used approaches for embedding nodes: local embedding, which captures the structural information of a node's immediate neighborhood, and global embedding, which represents the overall structure of the entire graph.

Local embedding : Local network embedding can be thought of as a way to represent the "neighborhood" of a node in a lower-dimensional space. By preserving the relationships between a node and its neighbors, local network embedding can capture important information about the network that is relevant to a specific node. One popular local network embedding technique is node2vec. Node2vec is based on random walks, which explore the graph by taking a random path through it. It enables efficient exploration of diverse neighborhoods in the graph by controlling the balance between a breadth-first search and a depth-first search. The node sequences generated by node2vec are then used to learn node embeddings using a Skip-gram model.

Global embedding : Global network embedding, on the other hand, focuses on capturing the overall structure of the entire graph. One popular global network embedding technique is Graph Convolutional Networks (GCNs). GCNs apply convolutional operations on the graph structure to propagate information and learn node representations. It has been shown to be effective in capturing global structural features of the graph, such as the community structure and node centrality.

2.2.3 Graph Neural Network

A Graph Neural Network (GNN) is a type of neural network designed to operate on graph-structured data. GNNs can be used to learn representations of the nodes and edges in a graph by recursively aggregating information from the graph's local neighborhoods.

Unlike traditional neural networks, which are designed to work with inputs of a fixed

size and structure, GNNs can handle inputs that have a variable size and structure, such as graphs. This makes GNNs a powerful tool for modeling and analyzing complex data that can be represented as graphs, such as social networks, molecular structures, and communication networks.

GNNs encompass a broad category of models that include various architectures, such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs). The choice of which architecture to use may depend on the specific context and goals of the problem at hand Graph Neural Network (GNN) allows us to transform and propagate node features as messages to learn structure-aware node representations. The Extract() mechanism extracts the messages from a target node and its neighborhood, which is later combined based on a chosen Attention() mechanism by the Aggregate() operation. Next, the aggregated message is propagated to the rest of the graph. Studies apply various aggregation strategies to combine the propagated and extracted messages from the target node's neighborhood based on the importance of each node in the neighborhood towards that target node. GCN[34] and GAT[91] are popular GNN frameworks.

Graph Convolutional Network(GCN) uses $N(*)$ neighborhood-based normalization constant to calculate the importance ($att_{v\longrightarrow u}$) of node v towards the target node u without considering the participating nodes' features as follows.

$$H_u^l = \sigma\Big(\sum_{\forall v \in \tilde{N}(u)} att_{v\longrightarrow u}^{l-1} W^{l-1} H_v^{l-1} \Big) \tag{2.1}$$

$$\begin{aligned} \text{Attention}_{\text{GCN}}(v \rightarrow u) \quad &: \\ att_{v\longrightarrow u}^{l-1} \quad &= \frac{1}{\sqrt{|\tilde{N(u)}||\tilde{N(v)}|}} \\ \text{Extract}_{\text{GCN}}(v) \quad &: \quad W^{l-1} H_v^{l-1} \\ \text{Aggregate}_{\text{GCN}}(*) \quad &: \quad \sigma(*) \end{aligned}$$

where σ is a non-linear activation function, W^l is a projection matrix for a GNN layer l and $\tilde{N}(*)$ is a node's extended neighborhood structure including the node itself (i.e., including self-loop edges).

Graph Attention Network(GAT) uses the same message extraction and aggregation

strategies as above except for the fact that it uses attentive aggregation strategies that consider both the participating nodes' features as well as the neighborhood information, as follows.

$$\text{Attention}_{\text{GAT}}(v \rightarrow u) : att^{l-1}_{v \longrightarrow u} = \underset{\forall v \in \tilde{N}(u)}{\text{Softmax}}\Big(c^{l-1}(W^{l-1}H^{l-1}_{u} \oplus W^{l-1}H^{l-1}_{v})\Big)$$

where c^{l-1} is a learnable parameter to approximate the importance of node v towards u ($att^{l-1}_{v \longrightarrow u}$) based on their interaction in the latent space in l layer-wise manner, W is the layer-wise projection matrix, and $\oplus$ denotes concatenation.

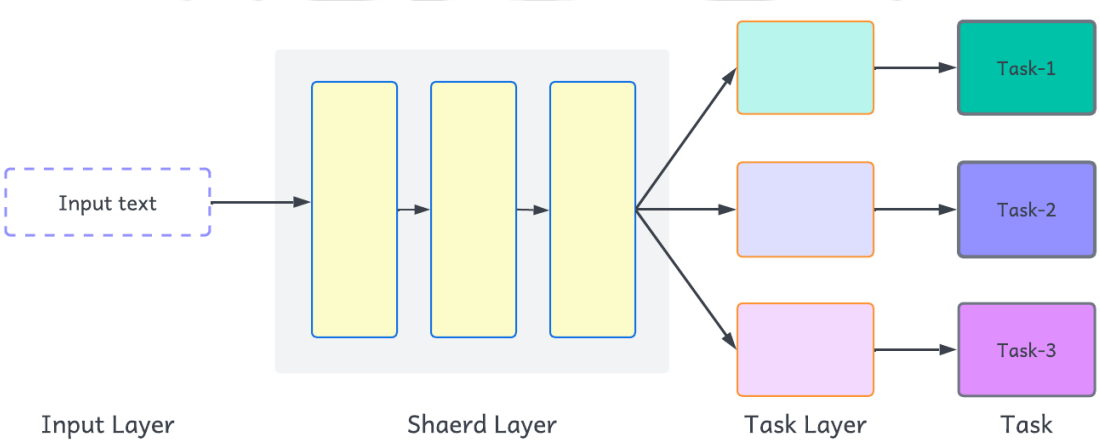


Figure 2.2: A Basic Multi-Tasking System with hard parameter Sharing

2.2.4 Multi Task Learning

While network analysis techniques can provide valuable insights into the structure and dynamics of complex systems, they may not always be sufficient for certain tasks or applications. In addition to the utilization of network analysis techniques, machine learning methods have also been employed in this thesis to address the problem of taxonomy expansion with attach and merge operations. To propose a novel taxonomy expansion model, multi-task learning has been employed, which enables the learning of multiple related tasks simultaneously. An overview of the multi-task learning approach used in this thesis is provided in the following section.

Multi-task learning is a machine learning technique in which a single model is trained to solve multiple tasks simultaneously. Unlike traditional machine learning, which fo-

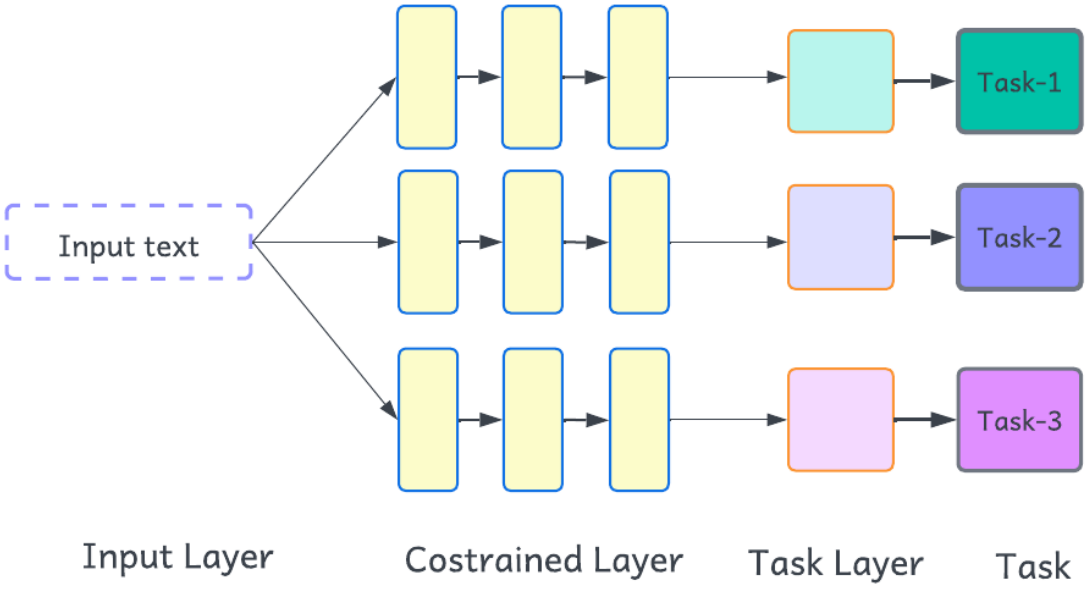


Figure 2.3: A Basic Multi-Tasking System with soft parameter Sharing

cuses on optimizing a single metric for a single task, multi-task learning allows for the optimization of multiple loss functions. This approach utilizes all available data across various tasks to learn generalized representations of the data that are useful for multiple purposes. Multi-task learning has been widely applied in domains like natural language processing, computer vision, and recommendation systems, and is widely used in industry due to its ability to effectively utilize large amounts of data to solve related tasks.

Multi-task learning methods for deep learning : In the context of Deep Learning, multi-task learning is commonly implemented through either hard or soft parameter sharing of hidden layers. Hard parameter sharing involves using a single set of parameters for multiple tasks, whereas soft parameter sharing involves having task-specific parameters that are initialized from a shared set of parameters.

- **Hard Parameter sharing** In Multi-Task Learning (MTL), Hard Parameter Sharing is a popular approach where the hidden layers are shared across all tasks and task-specific output layers are used. This approach reduces the risk of overfitting, as shown in[5], where the risk of overfitting the shared parameters is reduced by an order N compared to overfitting the task-specific parameters. This is due to the model finding a representation that captures all tasks, reducing the chance of

overfitting on a specific task.

- **Soft parameter sharing** In Soft Parameter Sharing, each task in Multi-Task Learning (MTL) has its own model with distinct parameters. To encourage the parameters to be similar, a distance metric between the parameters is regularized. For instance,[15] uses the 2 norm for regularization, while[102] employs the trace norm as the regularization method.

2.3 Summary

This chapter provides an overview of the relevant research domains that form the foundation for the work presented in this thesis. Both resource and technical aspects are included in this background. With this understanding, the first objective of the thesis, which is to automatically expand taxonomies by identifying missing synonymy relations, is studied in the following chapter.

There are things known and there are things unknown, and in between are the doors of perception.

Aldous Huxley, English writer

3

Synonymy Expansion using Link Prediction Methods: A case study of Assamese WordNet

This chapter focuses on the first objective of our research, which is to address the problem of missing synonymy relations in the WordNet taxonomy, with a special emphasis on the Assamese WordNet from IndoWordNet. The Assamese WordNet, created through the *expansion* method using the Hindi WordNet, also experiences a lack of synonymy relations. As WordNets can be visualized as a network of unique words connected by synonymy relations, link prediction in complex network analysis is a well-explored research field for predicting missing relations in a network. Hence, this chapter studies effectiveness of using state-of-the-art link prediction methods for automatically pre-

dicting missing synonymy relationships in Assamese WordNet. To perform the link prediction methods, we utilize only synonymy relationships from the Assamese WordNet taxonomy, and thus in this chapter, the Assamese WordNet taxonomy is referred to as the Assamese synonymy thesaurus, and the network created from it as the synonymy network.

3.1 Introduction

Assamese is an Indo-European language, spoken by almost 15.3 million speakers as the first language, in the province of Assam in the northeastern part of India[12]. Even though it is one of the scheduled languages of India, it is still considered an under-researched language in terms of the development of language processing tools and resource building[73]. Among many others, synonymy thesaurus in digital form for Assamese is still in a nascent stage. As noted in the studies, [61,3,80], synonymy thesaurus is one of the resources that play an important role in various text processing tasks such as information retrieval, information extraction, text classification, and summarization, etc. Further, for Assamese language, the Assamese WordNet from Indo WordNet $*$[8] is the only synonymy thesaurus available today. It should be noted that the Assamese WordNet is constructed using *expansion* approach, i.e., it is created by mapping Hindi WordNet to corresponding Assamese equivalent words. Here mapping implies the process of finding an Assamese word that shares the same meaning as a given Hindi word. By virtue of the way this resource is created, a few concerns are observed. Firstly, Assamese WordNet has low synonymy coverage (i.e., 4.79 synonymy per word) as compared to its source Hindi WordNet (9.35 synonymy per word). Examples of such missing relations are shown in 1 (a) - 1 (c) in the following paragraph. Secondly, the Assamese synonymy sets are constructed based on the ones available for Hindi. However, in spite of having the same Sanskrit roots, Assamese and Hindi have evolved separately as two different languages and hence, while two words may be phonologically very similar, the senses/concepts they convey may be very different. An example of such occurrences is provided in (2).

1. **Missing Synonymy Sets arising due to the use of Hindi synonymy relations**

$*$http://www.cfilt.iitb.ac.in/indoWordNet/

(a) Omission: The concept बहुत बोलने वाला व्यक्ति *'bahut bolane vaala vyakti'* (a very talkative person) in Hindi WordNet has two defined synonyms, i.e. वाचाल *'vaachaal'* (talkative) and अतिभाषी *'atibhashi'* (bilingual, talkative). Thus, following *expansion* approach, the Assamese WordNet has কথকী *'kothoki'* (talkative) and অতিভাষী *'Ativashi'* (talkative, bilingual) as synonymous words representing the underlying concept. However, words like কথা-চহকী *'kotha-sohoki'* (talkative) ,অতিবক্তা *'Atibokta'* (talkative), etc. are also potential synonyms which can closely define the concept.

(b) Novel senses/concepts : The Assamese words [ধীৰ *'Dhir'* (patience), শান্ত *'Xanto'* (calm) ,স্থিৰ*'Sthir'* (quiet) are synonyms with respect to the sense/concept *self composed*. As the sense/concept *selfcomposed* is not present in Hindi WordNet, the synonymy relationship between these words are also missing in Assamese WordNet.

(c) Human errors: The relations, such as, [চিলাই-কর্ম *'silai-karma'* (tailoring) , চিলাই_কাম *'silai_kam'* (tailoring)], [সূর্য_ৰশ্মি *'surjya_rashmi'* (sun-rays), সূর্যৰশ্মি *'surjyarashmi'* (sun-rays)], [তাম্বুল *'tamul'* (betel_nut), তাম্বোল*'tamul'* ('betel nut)] are few examples of absolute synonym relations, where each lexical item has a few morphological or phonological variations. While all these phonological variants are present in Assamese WordNet,they are not defined as a synonymy relation. Such issues may result from language specific features and inability to capture such features in manual efforts.

2. **Noise incurred from Hindi WordNet:** In Assamese WordNet, the words like ঝুঠা *'Jhootha'* (liar), ঝুট ' *Jhooth'* (lie), জখমী *'Jakhmi'* (Injury), ঝগড়া *'Jhagada'* (fight), গুণাহ *'gunaah'* (crime) are borrowed from Hindi WordNet, which are valid words in Hindi but not in Assamese. Since these words are not potential Assamese words, they introduce noisy words and noisy synonymy relations in the Assamese WordNet. Further the concepts in Assamese and Hindi are not exactly translatable. For example, the word মোটা*'mota'* (big) in Assamese is equivalent to the word मोटा*'motaa'* (fat/big) in Hindi. Though both the words share the same concept and phonology, the Hindi मोटा is used with both animate and inanimate

nouns, whereas the Assamese মোটা is used only with inanimate nouns.

As mentioned before, from (1) and (2) above, motivations for two research objectives stand out; (a) mining the missing set of synonymy relations and (b) handling the Hindi bias-induced noise in the Assamese WordNet. However, we found that 1 (a) - 1 (c) type of missing relations are more prevalent in the Assamese WordNet. Hence, in this work, we focus on addressing the issue of finding missing synonymy relations in the Assamese WordNet. As for the noisy words in (2), the Hindi-induced words are manually identified from the Assamese WordNet and removed from the experimental analysis in this study. The automatic identification of such noisy words is not considered within the scope of the study.

One obvious way of synonymy expansion is to expand manually, i.e., missing relations are manually checked and added to the network. Though manual expansion may provide precise relations, it is an expensive and time-consuming task. Further, finding linguistics experts for such manual effort is another challenge. On the other hand, thesaurus, like WordNet, can be visualized as a network of unique words connected by synonymy relations. Link prediction[42] in complex network analysis is a well-explored research field for predicting missing relations in a network. Motivated by the above, this chapter studies the effectiveness of using state-of-the-art link prediction methods for automatically predicting missing synonymy relationships in Assamese WordNet. Though other online Assamese dictionaries like Xobdo ∗ and Chandrakanta † also provide synonymy information; these resources could not be considered for our study as the underlying networks are not publicly available. The Xobdo and Chandrakanta do not provide information about the associated concepts. Further, some of the cohort words are also included in the synonymy list without distinction.

Previous studies on lexical synonymy prediction can be classified into two broad categories, (i) Dictionary based[70,10] and (ii) Corpus based [69,98]. Among dictionary-based methods, it has been reported that graph representations capture several latent properties of relations between words and help in better synonymy predictions[56]. Further,[83], and [56] report that resource like WordNet follows the scale-free characteristics of real-

∗http://www.xobdo.org/
†https://dsal.uchicago.edu/dictionaries/candrakanta/

world complex networks. As shown in Section 3.5.1, Assamese WordNet follows scale free[4] and small-world properties[99] of complex network. Link prediction methods[38,41] are generally used to discover future or missing relationships in complex networks such as, friendship networks[16], co-authorship networks[20], protein-protein interaction network[76] etc. Considering its simplicity, applicability, and efficiency of applying link prediction methods in a complex network, this study investigates the effect of various link prediction methods on predicting missing synonymy relations over partially completed Indo Assamese WordNet[8]. Though there have been few studies that use similarity between two nodes in a network to expand synonymy network[9,89], to the best of our knowledge, this is the first study which systematically investigates the effectiveness of different state-of-the-art link prediction methods in a complex network (Indo Assamese in particular). Popularly used link prediction methods in literature may be broadly classified into three; (i) network topological-based node proximity approach[42], (ii) representation learning (embedding) approach[67], and (iii) classification approach. This study considers various approaches covering the above three classes and analyzes their performances. To investigate the effectiveness of using link prediction methods for detecting missing synonyms, we organize the experimental setups in two ways. First, we verify the performance of each link prediction method by dividing experimental data into training and testing, i.e., *using the training edges, predict the test edges*. Like traditional supervised setups, both training and testing data are present in the original network. These experiments verify the suitability of using link prediction methods. From various experimental results, it is evident that many of these methods can effectively predict edges in the test set.

Further, we deploy these methods for predicting actual missing relations by applying to the original network. From various analyses, it is evident that link prediction methods in complex network analysis can effectively be used to predict missing synonymy relations. Noisy words are not considered within the scope of the thesis.

The major contributions of this chapter are summarized as follows:

1. Revisiting synonymy prediction problem for low resource language, namely, Assamese using network-science and graph-theoretic methods (See Section 3.4).

2. Studying characteristics of synonymy relations in Assamese WordNet (See Section

Table 3.1: Indo-WordNet Synonymy statistics

	Synonymy per concept		Synonymy per word	
	Average	Standard deviation	Average	Standard deviation
Assamese WordNet	2.30	3.00	4.79	11.48
Bodo WordNet	2.31	1.22	4.82	11.74
Oriya WordNet	2.08	1.25	4.03	9.96
Sanskrit WordNet	2.27	3.10	5.55	13.03
Nepali WordNet	2.33	2.92	5.01	12.11
Telugu WordNet	2.31	2.90	3.87	9.60
Punjabi WordNet	2.28	2.67	4.33	10.59
Bengali WordNet	2.16	2.47	5.08	12.27
Kannada WordNet	2.33	2.75	4.95	10.03
Gujarati WordNet	2.38	2.73	3.54	7.73
Kashmiri WordNet	2.25	2.62	4.54	11.12
Manipuri WordNet	2.22	2.56	4.23	10.40
Malayalam WordNet	2.17	2.54	4.05	9.81
Tamil WordNet	2.15	2.54	4.16	10.98
Urdu WordNet	2.05	2.45	2.66	8.13
Konkani WordNet	2.02	2.37	4.48	9.39
HINDI WordNet	3.75	4.83	9.35	14.64

3.3).

3. Empirical analysis of predicting missing synonymy relations in Assamese Wordnet using complex network analysis methods (See Section 3.5.3).

3.2 Related Studies

In the recent past, the problem of synonymy prediction (or synset expansion) has been studied in the context of extracting entity[69,77], attribute[26], and lexical synonymy words[89]. Though entity synonymy prediction received considerable attention, attribute, and lexical synonymy prediction have been less explored. Since this study focuses on lexical synonymy prediction, we present a brief overview of previous studies solving the lexical synonymy prediction problem.

Majority of the previous studies focusing on predicting lexical synonymy relations exploit either (i) Corpus-based approach[69,98] or (ii) Dictionary-based approach[70,10]. Corpus-based approaches exploit a large corpus of the underlying language and use distributional or pattern-based methods to predict synonymy relations[69]. Distributional approaches[42,100,62] assume that two words are synonymous if they frequently co-occur

in similar contexts. On the other hand, the pattern-based approaches[68,81] consider the local contexts and predict the synonymy relation between two words by the number of sentences mentioning both of them. However, corpus-based approaches for synonymy prediction are popular but not suitable for a resource-poor language like the Assamese, where limited amount of digitized information is available.

Among dictionary-based synonymy prediction approaches, graph-theoretic methods are very popular[27,10,9,89]. These approaches leverage a synonymy network/graph where nodes are represented by words, and an edge between two nodes defines the synonymy relation. Thereafter, social network analysis methods such as graph clustering and graph similarity are used on synonymy network to predict the missing synonymy relations. For example, ArcRank[27] uses a graph built over dictionary definition and employs a variant of PageRank[60] algorithm to compute the similarity between two words.[10] proposed a method for automatic synonym extraction from a dictionary. This method uses a graph constructed from the dictionary, where the nodes are defined by the words in the dictionary. Two nodes are assumed to be connected if there are common words between the definitions of the nodes. Given a word, it uses HITS ranking algorithm[36] to generate a topic-driven sub-graph using two hops neighbors of query word and rank the nodes in the sub-graph. The word associated with the highest ranked node is considered as the most similar word of the query word, and so on. While the above studies attempt to find similarity between words by exploiting the topological structure of the network using random walk, the studies like[69],[40],[19] exploits neural-based embedding models to capture the similarity between words.

In earlier studies, link prediction methods are generally used to discover missing links in complex networks such as friendship networks[16], co-authorship networks[20], protein-protein interaction network[76] etc. This study investigates the effectiveness of various link prediction methods for predicting missing synonymy relations in a synonymy network. The majority of the earlier studies[42],[2],[59] on link prediction mainly focus on topological structure and properties of the network. A good survey on link prediction methods using topological structure can be found in[95]. Recently, studies on link prediction using network representation learning methods are also evident. Node2Vec[22], VERSE[88], DeepWalk[63], Graph Convolutional Networks[35], Relational Graph Convolu-

tional Networks[75] are some of the recent network representation learning methods which can be used for link prediction. Once representations of nodes or edges are generated, classification methods are then applied to perform the link prediction task.

3.3 Synonymy relations in Assamese WordNet and its issues

Synonymy is a semantic relation between words. It is very difficult to give a clear and precise definition of synonymy. Linguistically, two or more words are called synonyms in the same language with the same or very closely related meaning in some or all senses/concepts[53]. A replaceable test can determine the relation of sameness in meaning, i.,e. two words are similar if and only if they can be replaceable by one another in a sentence without changing the truth value[54]. In general, there are two categories of synonymy namely absolute synonymy and near synonymy [17]. Absolute synonymy can be replaceable by one another in all contexts, while near synonymy can be replaceable in some but not all contexts. Both absolute synonymy and near synonymy relations are reflexive and symmetric, while only absolute synonymy relations are transitive[25]. Generally, it is recognized that absolute synonymy is quite rare, and lexicographers have always treated synonymy as near synonymy [18]. Near synonyms are almost synonyms, but not quite; very similar, but not identical, in meaning; not fully inter-substitutable. They can differ in terms of their shades of denotation, connotation,[14].[18] classifies near synonymy variation into 35 subcategories within the four broad categories. In Assamese, all these near-synonyms variations are extensively found[6]. However, in this study, we are not focusing on these subcategories of synonymy rather focusing sameness in meaning and replaceability of a word in the same context.

WordNet is a lexical database that consists of synset and semantic relations (hypernymy, hyponymy, meronymy, troponymy)[52]. Synsets are the fundamental building blocks of WordNet. Synset is a set of synonymous words that are replaceable in some or all contexts[52]. Therefore words that share a synset must be either absolute synonymy or near synonymy. Absolute synonymy generally includes various spellings such as litre and liter, abbreviations such as kg and kilograms, etc. All absolute synonymy should be present in the same synset[25]. However, in the case of near synonymy, the occurrence of near synonymy in the same synset is based on the underlying concept and synset

creation principle, namely, minimality, coverage, and replaceability[13].

As mentioned in[13], Assamese WordNet is created from Hindi WordNet by mapping Hindi synonymy words with Assamese words of similar meaning. For example, the concept for *bad omen* has only two representing words অশুভ লক্ষণ *'Axuvo lakhyon'* (bad omen) and কুলক্ষণ *'kulakhyan'* (ill omen), in Assamese WordNet. These two words are mapped directly from Hindi WordNet and translated into Assamese. As a result of that several other meanings, representative of the concept remained excluded. At the same time, a word like অমঙ্গল *'amangal'* (bad omen or evil omen) which is a more commonly used word is not included in the synset. Further, we have observed that synonymy relations in Assamese WordNet are noisy and not complete. While the related words may be present in the database, relationships with other words in the database may not be defined. There are several reasons for these shortcomings. Firstly, in the case of the Assamese WordNet, Hindi WordNet is considered to be the source, based on which concepts and individual words are added in the Assamese WordNet. However, as Assamese and Hindi are two different languages, the concepts and words may not be exactly translatable, in spite of phonological similarity between words. For example, word চিকমিকাই থকা *'sikmikai thoka'* (sparkling or twinkling) in Assamese is comparable to the Hindi word चमकना*'chamakna'* (shine). In spite of the phonological similarity between the two, the Hindi word may be used with both animate and inanimate nouns. However, the Assamese correspondence can be used only for inanimate, metallic objects. This is a major shortcoming resulting from the direct transfer of concepts and words from a source language to a target language.

Considering the discussion above, Assamese WordNet suffers from several weaknesses, such as (i) incomplete relational structure and (ii) noisy relationships. This implies that they have a lexical relationship that should not be present(Insertion error); on the other hand, they have not defined some common lexical relations that should be present(deletion error). Thus discovering missing synonyms are important for effective utilization of Assamese WordNet.

3.4 Link Prediction in Complex Network

Given a network $G(V, E)$, where V is the set of nodes and E is the set of edges, the link prediction problem estimates likelihood of introducing new edges in the network[42]. Moreover, link prediction has been studied in two perspectives, namely, (i) given the snapshot of a network at time t, predict edges which will appear in future, and (ii) given an incomplete network, predict the missing edges. Since this chapter focuses on predicting new synonymy relations from a given incomplete synonymy network, this study relates to the second definition of link prediction.

From earlier studies, the majority of the link prediction methods broadly follow two taxonomies, namely, (i) topology-based (exploit the topological structure of the underlying network using proximity measures defined for capturing local/global similarity) and (ii) learning-based (generate network features from the underlying network and train a classification model). Since the objective of this study is to systematically examine the applicability of link prediction methods in synonymy prediction, we consider the popular link prediction approaches from both of the taxonomies. The remaining part of this section briefly discusses the link prediction methods considered in this study.

3.4.1 Topology-Based Link Prediction

Local Similarity-Based

The link prediction methods based on local similarity exploit local neighborhood proximity to estimate similarity between two disconnected nodes. We consider the following four popularly used link prediction methods. For a given network $G(V, E)$, let $\mathcal{N}(v)$ represents the set of neighbors of node v. The different link prediction methods are defined as below.

1. **Common Neighbor(CN)**[42]: It estimates the similarity score by counting the total number of common neighbors between the underlying pair of nodes.

$$\mathcal{S}_{CN}(u, v) = |\mathcal{N}(u) \cap \mathcal{N}(v)| \tag{3.1}$$

2. **Jaccard Co-efficient**[42]: It is the normalized version of above-defined common neighbor measure. The similarity score is defined as

$$\mathcal{S}_{JC}(u,v) = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v)|} \tag{3.2}$$

3. **Adamic Adar**[1]:

   This similarity measure is based on giving more weight to less common nodes or nodes having less number of neighbors. It is defined as

$$\mathcal{S}_{AA}(u,v) = \sum_{z \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{log(|\mathcal{N}(z)|)} \tag{3.3}$$

4. **Resource Allocation**[59]: Like AA, this similarity measure also weights more to those nodes having less number of neighbors and defined as

$$\mathcal{S}_{RA}(u,v) = \sum_{z \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{|\mathcal{N}(z)|} \tag{3.4}$$

Global Similarity-Based:

The local-proximity-based link predictors defined above capture only the two-hop proximity between the disconnected nodes. However, in real-world complex and social networks, the new edges may appear between two nodes connected via a path of length greater than two. Thus, capturing global proximity is an essential requirement in estimating the likelihood of new edges. We consider two popular methods exploiting global proximity using paths between nodes to estimate the similarity between two nodes, namely, (i) Katz index[42] and (ii) Friendtns[84]

1. **Katz Index**[42]: Katz index estimates the similarity between two nodes by counting total number of paths of all lengths. Further, the paths are damped exponentially to give shorter paths more weights. The similarity score between two nodes using Katz index is defined as

$$\mathcal{S}_{katz}(u,v) = \sum_{l=1}^{\infty} \beta^l . |paths_{u,v}^{(l)}| \tag{3.5}$$

where $|paths_{u,v}^{(l)}|$ is the set of all paths of length-l from nodes u to v.

2. **FriendTNS**[84]: This similarity measure exploits a hybrid approach capable of capturing both local and global graph characteristics to measure the similarity between two nodes. For an undirected and simple network $G(V,E)$ (with no loops or multiple edges), the similarity score between two nodes using FriendTNS can be defined as

$$\mathcal{S}_{FrendTNS}(u,v) = \begin{cases} 1, & if \quad u == v \\ 0, & if \quad u \neq v \wedge (u,v) \notin E \wedge (v,u) \notin E \\ \frac{1}{deg(u)+deg(v)-1}, & otherwise \end{cases} \tag{3.6}$$

Random Walk-Based:

As observed above, both local proximity and global proximity-based similarity measures can be used for the link prediction task. Further, it is difficult to say that capturing which proximity (local or global) would lead to better performance in predicting links. Thus, balancing the trade-off between local and global similarity measures is vital because it harnesses the local as well as global characteristics of the underlying network. In the past, random walk-based similarity measures have achieved significant considerations because of its capability in capturing both local and global characteristics. Thus, in this study, we consider a popular random walk-based similarity measures, namely, Rooted PageRank .

Rooted PageRank[42]: This similarity measure is a variant of popular PageRank centrality proposed by Brin and Page in the study[60]. PageRank estimates the centrality of the nodes in the underlying network using the stationary probability distribution of random walks subjected to follow the network structure for a given probabilistic value $d \in (0,1)$ (i.e., damping parameter) and jumps to any random node by probability $1-d$. Unlike traditional PageRank, Rooted PageRank allows the random walker to jump to a particular node or root. Thus, the centrality estimates are relative to the given root node and can be used as a similarity score.

For the given graph $G(V,E)$, let $\mathbf{A}$ is the adjacency matrix representation of G, and $\mathbf{W}$ represents the transition probability matrix such that

$$\mathbf{W}[u,v] = \begin{cases} \frac{\mathbf{A}[u,v]}{\sum_{w \in V} \mathbf{A}[u,w]}, & if \quad (u,v) \in E \\ 0, & Otherwise \end{cases}$$

where $\sum_{v \in V} \mathbf{W}[u,v] = 1$. Let the vector $\mathbf{r}$ give the centrality score for all the nodes and d be the damping parameter, then the PageRank of node u can be defined as

$$\mathbf{r}[u] = d \sum_{v \in V} \mathbf{r}[v]\mathbf{W}[v,u] + (1-d)\frac{1}{n}$$

where n is the total number of nodes in the network. In matrix form, we can write it as below.

$$\mathbf{r} = [d\mathbf{W} + (1-d)\mathbf{I}\mathbf{p}^T]\mathbf{r} \tag{3.7}$$

where $\mathbf{I}$ is the identity vector i.e., $\mathbf{I}^T = \{1,1,1.....1\}$ and $\mathbf{p}^T = \{\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}\}$. It should be noted that PageRank allows the random surfer to jump to any random node with equal probability (i.e., $1-d$) given by vector $\mathbf{p}$ in Equation (3.7). As discussed above, the Rooted PageRank allows the random surfer to jump to a root node only. Thus, Equation (3.7) can be used to formulate Rooted PageRank by modifying the vector $\mathbf{p}$ such that it sets probability 1 to the root node and 0 to other nodes in vector $\mathbf{p}$. For example, let $\mathbf{r_u}$ denotes the Rooted PageRank score rooted at node u then $\forall_{v \in V \wedge (v \neq u)} \mathbf{p[v]} = 0$ and $\mathbf{p[u]} = 1$. Further, each centrality value in vector r_u gives the similarity score of the corresponding node to the root node. For example, $\mathbf{r_u[v]}$ gives the Rooted PageRank similarity for an edge from node u to node v. As evident, the above formulation of Rooted PageRank similarity score is not symmetric, i.e., $\mathbf{r_u[v]} \neq \mathbf{r_v[u]}$. Thus, the symmetric Rooted PageRank similarity score can be defined as

$$\mathcal{S}_{rpr}(u,v) = \frac{1}{2}(\mathbf{r_u[v]} + \mathbf{r_v[u]}) \tag{3.8}$$

3.4.2 Learning-Based Link Prediction

Inspired by the efficiency of machine learning models such as classification and regression, a large volume of previous studies on link prediction employs a two-step process, i.e., (i) extract the edge features using the underlying network structure and (ii) train a classification model over these features for link prediction [67]. Further, extracting

edge features can be done by exploiting various network characteristics such as node centrality and the proximity among the nodes[33]. We refer these types of features as explicit node/edge features because they consist of explicit node/edge characteristics. Alternatively, some of the early and recent studies consider the latent representation of the network using matrix factorization, neural network, etc., to extract network features corresponding to node or edge[37,69,22]. Therefore, this study considers both types of feature extraction approaches (i.e., explicit and latent representation) and exploits Naive-Bayes classifier to predict missing synonymy relations in the above-discussed synonymy networks.

Explicit feature-based:

As discussed above, explicit features can be extracted corresponding to both nodes and edges; we exploit node and edge characteristics to extract two types of explicit features. The first type of explicit features considers various node centrality measures, namely, (i) degree, (ii) eigenvector, (iii) closeness, (iv) average neighbour degree, and (v) clustering coefficient to extract features corresponding to all the nodes. Thereafter, similar to studies[22,88], using Hadamard product over node features, we extract the edge features. The second type of explicit features considers various edge similarity measures, namely, (i) Jaccard Coefficient, (ii) Adamic Adar, (iii) Resource allocation, (iv) Preferential attachment, and (v) Total neighbours to extract the edge features.

Latent feature-based:

In the past, the majority of the studies focusing on automatic feature generation using latent representations have exploited matrix factorization-based approaches[49]. However, matrix factorization-based methods are non-scalable to the real-world large social and information networks. Thus the majority of the recent studies exploit neural network-based methods for embedding the given network in latent dimensions. These approaches are inspired from the popular Word Embedding (Word2Vec) model[50] and are often referred to as network embedding methods. Furthermore, it is evident from recent studies that neural-based network embedding is scalable and suitable for large real-world networks.

Network embedding refers to representing the underlying network into low dimensional network features corresponding to node, edge, sub-structure, or whole network. As this study focuses on predicting missing synonymy words, we consider the network embeddings corresponding to nodes (i.e., words) which have been referred to as node embeddings in the subsequent parts. In general, node embedding methods exploit single-layer neural network, namely skip-gram, which maximizes the log probability of neighboring nodes $\mathcal{N}(v)$ for a given node v. For a given network $G(V, E)$, let $f: V \rightarrow R^d$ be a function representing a node $v \in V$ to a d-dimensional feature vector; then the objective is to maximize the following objective function.

$$\sum_{v \in V} \sum_{n \in \mathcal{N}(v)} log\ P(n|f(v)) \tag{3.9}$$

where $P(n|f(v)$ is the conditional probability of observing neighbor node n for the given node v.

In general, unsupervised network embedding using neural networks is often a two-step framework. At first, extract the node neighborhood such that it captures the structural and similarity characteristics. After that, train a neural network-based model to generate a node vector such that two similar nodes in original network space are represented closely in embedding space. In other words, maximize the Equation (3.9) to enhance the probability of neighboring nodes for a given node. Most unsupervised node embedding models differ in consideration of the sampling approaches for generating the node neighborhood $\mathcal{N}$. This study considers two recently proposed and efficient node embedding models differ in terms of node neighborhood, namely, Node2Vec[22] and VERSE[88].

1. **Node2Vec:** As discussed above, capturing the neighborhood information of nodes is one of the important process in network embedding. Node2Vec network embedding model attempts to preserve neighborhood information by maintaining two properties of network namely structural equivalence and homophily. It uses two hyper-parameters p and q where p is probability of returning to the node visited earlier and q is the probability of visiting unexplored neighbor. Node2Vec visits nodes in the network using a 2^{nd} order random walk with a transition probability

estimated using the parameters p and q. Suppose a random walker just traverses the edge $(u, v) \in E$ and is resting at the node v. Now the walk has to estimate the transition probability to visit the next node x originating from v. Here Node2Vec sets the unnormalized transition probability w_{vx} to $\pi_{vx} = \alpha_{pq}(u, v).w_{vx}$, where

$$\alpha_{pq}(u, v) = \begin{cases} \frac{1}{p} & if d_{ux} = 0 \\ 1 & if d_{tx} = 1 \\ \frac{1}{q} & if d_{ux} = 2 \end{cases} \tag{3.10}$$

and d_{ux} is the shortest distance between nodes u and x. Node2Vec tunes the parameters p and q over a small sample of the underlying network in semi-supervised way and generates a collection of node sequences visited in subsequent walks. The collection of walks defined by node sequence is then used to train a skip-gram model defined in Equation (3.9) to generate the node embedding vectors.

2. **VERSE:** This node embedding model exploits similarity measures such as personalized PageRank[60], SimRank[28], etc. to capture neighborhood information of a given node. Thereafter, it trains a skip-gram-based neural network model for generating node embeddings which preserve the vertex-to-vertex distribution in embedding space.

   Let $Sim_G(v, .)$ and $Sim_E(v, .)$ represent the similarity distribution in the original network space and similarity distribution in the embedding space, respectively. Then the objective of VERSE embedding model is to minimize the Kullback-Leibler (KL) divergence from $Sim_G(v, .)$ and $Sim_E(v, .)$. In other words, VERSE aims at minimizing the distortion of original network in embedding space which is given by:

$$\sum_{v \in V} \mathbf{KL}(Sim_G(v, .) \;||\; Sim_E(v, .)) \tag{3.11}$$

   VERSE defines the unnormalized distance between two nodes u and v as the dot product of their embeddings i.e., $\bar{u}^T \cdot \bar{v}$. Now, normalizing the similarity distribution with softmax

$$Sim_E(v, u) = \frac{exp(\bar{v}^T \cdot \bar{u})}{\sum_x exp(\bar{v} \cdot \bar{x})} \tag{3.12}$$

From Equation (3.11), the objective is to minimize the KL divergence between Sim_G and Sim_E.

$$\mathcal{L} = -\sum_{v \in V} Sim_G(v, .) log(Sim_E(v, .) \tag{3.13}$$

Graph Convolution Network :

The above discussed latent feature-based methods, namely Node2Vec and VERSE, exploit a random walk-based paradigm to generate node embedding. Recent trends employ graph neural network-based paradigm to generate node embedding. Graph convolution network (GCN) is one of the popularly used methods to generate node embedding using Graph Auto Encoder (GAE) model[35]. Given a graph G, the encoder transforms the input graph to a stochastic matrix Z of $n \times m$ dimensions, where n is the number of nodes and m is the number of output units in GCN. GCN takes two input matrices A and X where A is the adjacency matrix of the network and X is a feature matrix of dimension $n \times d$ *. GCN can be mathematically represented as follows:

$$\hat{X} = GCN(X, A) = \sigma(\tilde{A}XW)$$
$$\tilde{A} = D^{(-\frac{1}{2})}AD^{(-\frac{1}{2})}$$

where $\tilde{A}$ is the symmetrically normalized adjacency matrix, D is the degree matrix, W is the weight parameter of the neural network, and σ is the activation function. In this study, we use ReLu activation function and employ two-layer GCN defined as follows:

$$GCN(X, A) = \sigma(\tilde{A}\sigma(\tilde{A}XW_1)W_2)$$

where W_1 and W_2 are the weight matrices for the first and second layers of the GCN. The matrix Z is then generated using linear combination of two GCNs sharing the weight of first layer.

$$\mu = GCN_{\mu}(X, A) = \sigma(\tilde{A}\sigma(\tilde{A}XW_1)W_2)$$
$$\delta = GCN_{\delta}(X, A) = \sigma(\tilde{A}\sigma(\tilde{A}XW_1)W_3)$$

*Since we have no feature matrix, we consider X as the identity matrix.

Table 3.2: Network characteristics of Assamese synonymy network

properties	Assamese WordNet
Nodes	24533
Edges	68031
Average degree	5.5461
Clustering coefficient	0.905
Connected components	4894
Average path length	11.338
Min degree	1
Max degree	100
Network diameter	34
% edges and nodes in the giant component	53%edges and 28% nodes

$$Z = \mu + \delta * \varepsilon \tag{3.14}$$

where $\varepsilon \sim N(0,1)$.

3.5 Experimental Setup and Discussions

3.5.1 Dataset

In this section, we discuss the characteristics of the dataset used in this study. The synonymy network is extracted from Assamese WordNet, where the nodes are the words, and the edges are the synonymy relations. It should be noted that synonymy relations in this network are not bounded by concepts. Even if a word is present under different concepts, its relations are considered without distinction. The Assamese WordNet consists of 14958 synsets, out of which 5641 synsets are singleton. Here singleton synset implies single word synset (the only word which uniquely represents a concept and has no relation to any other words). We have removed singleton synset as they have no connection with the graph and they will increase the number of components only. We are considering only words that have at least one connection with other words.

Table 3.2 shows the characteristics of the synonymy network. It has 24533 nodes and 68031 edges with an average path length of 11.34 and network diameter of 34 edges. There are 4897 number of connected components with component sizes ranging from 1 node to 7027 nodes. The giant component covers 53% of the edges and 28% of the nodes. Figure 3.3 shows distribution of components against component size.

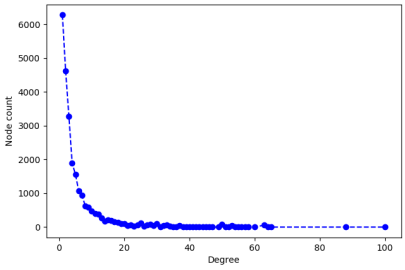


Figure 3.1: Degree distribution of synonymy network.

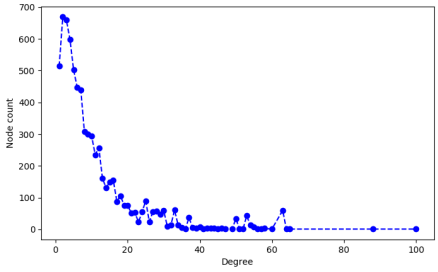


Figure 3.2: Degree distribution of the giant component.

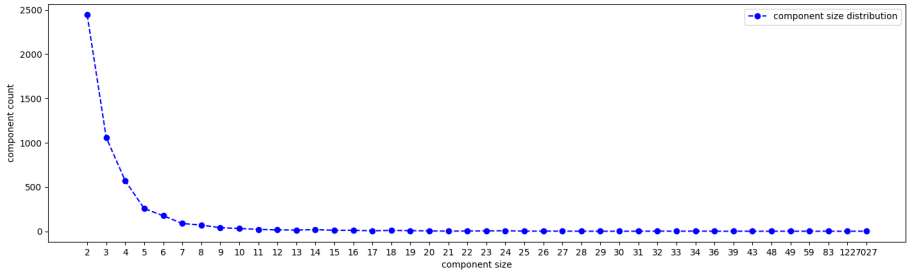


Figure 3.3: Component size distribution of synonymy network

Figure 3.1 and Figure 3.2 show degree distribution of the entire network and the biggest component of the network. It clearly shows that synonymy networks satisfy power low distribution, and exhibits properties of complex network (social network). Thus, the rich literature of social network analysis may be effectively used for mining missing relations in Assamese synonymy network.

3.5.2 Hyper-parameter settings

All the link prediction methods based on network embedding require multiple hyper-parameter setting. In this study, to compare various methods, we consider the length of the final embedding vector as 128. Furthermore, we consider the values of hyper-parameters as reported in the original paper for all the models. For example, the walk length in Node2Vec is set to 80, and number of walks is equal to 80 with window size 10. Further, the learning rate in GCN is set to 0.01 with two hidden layers of size 32 and 128. Similarly, we consider Personalized PageRank as the similarity measure in VERSE to capture the node neighborhood and exploit NCE to train the Skip-Gram-based model.

3.5.3 Evaluating link prediction performance

We conduct two sets of experimental analysis; (i) *from the synonymy network defined in Table 3.2, some of the edges are randomly removed, and the removed edges are again discovered using link prediction methods*, and (ii) *consider the entire network and discovers actual missing relations*. The first experiment is important to assess the suitability of applying different link prediction methods for predicting actual missing relations in the network. The second experiment will validate the performance of discovering actual missing synonymy relations in the Assamese WordNet.

Are link prediction methods capable of discovering missing relations in synonymy network?

To understand the suitability of applying state-of-the-art link predictions to discover missing relations in a network, we first investigate the performance of different link prediction methods in the traditional experiment setup, *i.e., a small set of edges are randomly removed from the network and then the ability to discover the removed edges are investigated*. For fair evaluations, the edges in the network are randomly divided into 5-partitions. For each partition of existing edges (samples under the positive class), an equal number of non-existing edges (samples under the negative class) between the nodes present in the partition are randomly identified. Thus, the experimental dataset with five folds is generated. The experimental results shown in this section are obtained using 5-fold cross-validation.

As discussed in Section 3.4, we group the link prediction methods into two; (i) unsupervised approach (referred to as *topological*), and (ii) supervised approach (referred to as *learning based*). The local estimates (RA, CN, AA, JC), the global estimates (Katz, FriendTNS), and random walk-based estimates are unsupervised methods. In the supervised approach, we build a classifier (using Naive Bayes) between positive edges and negative edges. We apply different methods of generating vector representation for each edge, as discussed in section 3.4. In one approach, the edge vector is generated using explicit edge features. In another approach, edge vectors are generated using Hadamard product[22] between node vectors (using explicit node features or node embedding).

Table 3.3: Results of Link Prediction Methods

			WordNet	Giant Component	WordNet without Noise
Approach	Environment	Method	AUC	AUC	AUC
Topological	Local	RA	0.988470	0.993191	0.988495
		CN	0.988420	0.993130	0.988444
		AA	0.988455	0.993162	0.988479
		JC	0.988486	0.993212	0.988511
	Global	katz	0.993106	0.996018	0.993133
		FriendTNS	0.897115	0.865231	0.897144
	Random walk	Rooted page rank	0.717132	0.709423	0.717132
Learning based	Explicit feature	Node feature	0.623411	0.623385	0.623413
		Edge feature	0.543111	0.541261	0.543111
	Latent feature	node2vec	0.979651	0.966532	0.979651
		VERSE	0.979711	0.966108	0.979712
		GCN	0.882314	0.883480	0.882317

As this study attempts to predict missing synonymy relations using link prediction method, we evaluate the efficiency for synonymy prediction using popularly used link prediction evaluation metric namely, Area Under ROC Curve (AUC)[24]. The ROC curve gives the ratio between true positive rate to the false positive rate over predicted synonyms. The AUC score shows the likelihood of predicting a positive synonymy relation whenever a random missing synonymy relation is predicted. In other words, higher the AUC score signifies the link predictor model's efficiency in correctly predicting the missing synonymy relations. Let p be the number of existing links (positive synonymy relations) and n be the number of non-existing links (negative synonymy relations). The AUC score can be defined as follows[24].

$$AUC = \frac{\sum_i c_i + 0.5 \sum_i d_i}{p.n} \tag{3.15}$$

where c_i denotes a quantity equal to number of times an existing link gets link prediction score greater than the non-existing link prediction scores. Further, d_i denotes a quantity equal to number of times an existing link prediction score is equal to non-existing link prediction scores. AUC score (Equation (3.15)) gives higher score to the existing links having higher link prediction score than non-existing link prediction score and an equal score for the cases where link prediction score for existing and non-existing links are equal. Thus, as described above, higher the AUC score we have a better and robust model for predicting missing synonymy relations.

Table 3.3 shows the performance of different link prediction methods using AUC

scores. The column number 4 shows the AUC scores of different methods over the entire network. Whereas the column number 5 shows the AUC scores of different methods over the giant component. As discussed in section 3.5.1, Assamese WordNet has a giant component and many other small components. Further, column number 6 shows the AUC score of different methods over the noise removed network. As discuss in section 3.1, Assamese WordNet has Hindi induced noisy words. To examine the possible risk in modeling link prediction methods from the existence of noise in the original network, we have removed noisy words from the original network and calculated the AUC score over the noise removed network. However, in our manual investigation, We randomly selected a sample of 168 words that were identified as noisy in the dataset. We have not observed any significant change in the AUC score. From Table 3.3, it is evident that 63% (7 out of 11) of the link predictors achieve a very high AUC score (i.e., closed to 0.99 in discovering the manually removed synonymy edges from the network). It is interesting to observe that all the local proximity measures provide an AUC score of 0.99 for both the entire network or giant component. High AUC scores of local proximity measures can be justified by the nature of the underlying synonymy network. As shown in Table 3.2, a high average clustering coefficient of nodes with 0.905 means the formation of triads among the neighbors of a node. The formation of triads further leads to the formation of cliques among the neighbors of a node. As a result, a randomly removed edge from the network has a very high probability of belonging to a clique. Such a triadic edge is likely to have a higher prediction score than non-triadic edges. It is also interesting to observe that classifiers built using node embedding (Node2Vec and VERSE) also achieve high AUC scores (0.99 over the entire network and 0.97 over giant component). It also indicates the ability of the node embedding methods to capture both structural and proximal characteristics of the network. However the classifier built using GCN gives poor performance in comparison to classifier built using Node2Vec and VERSE. With the parameter tuning of the feature matrix, GCN performance might be improved. Furthermore, classifiers built using explicit features perform poorly. It is because the explicit features that we consider in this study (though identified from the studies[95]) fail to capture proximity distance between two nodes. As for example, two nodes in different network components may have very similar structural properties;

Table 3.4: Example predicted relations

Target Word	New relations	Categories
চিলাই-কর্ম *'silai-karma'* (tailoring) তাম্বুল *'tamul'* (betel-nut) হাৰামখোৰ *'haramkhor'* (dishonest, bastard)	চিলাই_কাম*'silai_kam'* (tailoring) তাম্বোল *'tamul'* (betel-nut) হাৰামী *'harami'* (dishonest, bastard)	Absolute synonymy
জুঠা *'jutha'* (left-over) নিকটতম *'nikattam'* (adjacent, side) ভালদৰে *'valdore'* (properly)	এৰেহা *'ereha'* (left-over) কাষৰ *'kaxor'* (adjacent, side) ঠিককৈ *'thikkoi'* (properly)	Near synonymy
দিয়াচলাই_কাঠী *'diaasolai_kathi'* (match_box stick) নাপ *'naap'* (measure) ফালতু *'phaltu'* (nonsense, useless)	মেচ_বক্স*'mes boks'* (match box) ইস্কেল*'scale'* (scale) অদৰকাৰী *'adarkari'* (useless)	Semantic cohort

they are physically not connected to each other. Therefore, automatic embedding using representation learning may be encouraged as compared to manual feature selection approaches.

From the above observations, it is evident that a missing synonymy relation in a dense region of the network can be effectively discovered using state-of-the-art link prediction methods. By virtue of the property of synonymy relation in WordNet, words under a *concept* are absolute synonymy and form an equivalent class (i.e., satisfy reflexive, symmetric, and transitive relations). If there are missing relations while creating WordNet (which is highly possible for the target Assamese WordNet, as it is created by expanding from Hindi WordNet), it can be effectively identified using the above link prediction methods. Further, it is also observed that simple local proximity-based methods are effective enough to identify such relations.

As described above, topology-based link prediction methods are simple yet powerful in capturing the relational structure of the given Assamese WordNet. Thus, we further evaluate the capability of link prediction by an ensembling framework exploiting the collecting efficacy of all the topological link predictors given in Table 3.3. We consider the link similarity score for an edge from all the individual link predictors and estimate an average similarity score. Since the topological link predictors considered in this study capture different characteristics, the link scores are normalised before estimating the average score. We record 0.8157 as the AUC score for link prediction using the ensembling framework.

Discovering actual missing synonymy relation in the Assamese WordNet:

From the above section, we have observed that link prediction methods are capable of discovering missing relations. In this section, we apply the link prediction methods and attempt to discover actual missing synonymy relations in the original Assamese WordNet and evaluate the performances. For this study, we now transform the task as synonymy retrieval task, i.e., for a given word, retrieve a list of synonymy words that are not actually present in the existing network using link predictors.

There are about 24,500 words in the Assamese WordNet, and there is no ground truth dataset. Therefore, the only way to evaluate the performance of discovering missing synonymy words is subjective evaluation through manual annotation. Manual evaluation of all 24,500 words for different methods is an expensive task. To reduced the evaluation cost, we randomly sampled only 500 words from the WordNet. The top ten results obtained from different predictors are given to human annotators to evaluate the quality of the results. The annotators are asked to perform the replaceability test, as mentioned in Section 3.3, and identify synonymy categories, i.e., absolute synonymy, near synonymy and cohort word.

We have employed five human annotators to evaluate the newly predicted synonymy by the above-discussed link prediction methods. Annotators are native Assamese speakers and have a clear knowledge of the replaceability test that we discussed in Section 3.3. For each pair of synonymy relations, the inter-annotator agreement is 80% in each case, i.e., absolute synonymy, near synonymy, and cohort word. However, most of the disagreement arises when an annotator encounters a word that is not known to his knowledge. For the words which have multiple concepts in Wordnet, we consider only one concept from those, i.e., the first occurring concept in resultant synonymy list. The resultant list of the human annotation process is considered as the ground truth data, which is used to evaluate the retrieval performance. Table 3.4 shows some of the discovered absolute synonyms, near synonymy, and semantic cohorts.

We use Mean average precision at position k(MAP@k) to evaluate the retrieval performance. For a given query word, q, we calculate its corresponding Average Precision at k, and then the mean of the all queries gives Mean average precision. The metric MAP quantifies how good our model is at performing the query. The mean average

Table 3.5: Mean average precision score of the predictors in predicting top 10 actual synonymy words

Method	method	map@1	map@1	map@3	map@4	map@5	map@6	map@7	map@8	map@9	map@10
Topological (Local)	CN	0.59	0.64	0.62	0.62	0.65	0.65	0.66	0.69	0.7	0.68
	RA	0.55	0.61	0.6	0.61	0.64	0.64	0.63	0.66	0.67	0.66
	AA	0.51	0.55	0.53	0.51	0.53	0.53	0.54	0.57	0.56	0.54
	JC	0.57	0.62	0.61	0.62	0.65	0.64	0.64	0.68	0.7	0.7
Topological (Global)	Katz	0.57	0.63	0.63	0.65	0.66	0.65	0.64	0.63	0.62	0.6
Learning based	Node2vec	0.52	0.56	0.57	0.58	0.57	0.57	0.57	0.57	0.58	0.58

Table 3.6: Mean average precision score of the predictors in predicting top 10 semantic cohorts

Method	method	map@1	map@1	map@3	map@4	map@5	map@6	map@7	map@8	map@9	map@10
Topological (Local)	CN	0.82	0.85	0.84	0.83	0.82	0.82	0.82	0.85	0.85	0.86
	RA	0.8	0.84	0.83	0.82	0.82	0.81	0.81	0.83	0.83	0.85
	AA	0.6	0.65	0.63	0.63	0.66	0.65	0.65	0.68	0.68	0.67
	JC	0.82	0.85	0.84	0.83	0.82	0.81	0.81	0.84	0.85	0.86
Topological (Global)	Katz	0.83	0.86	0.87	0.88	0.88	0.87	0.86	0.85	0.85	0.83
Learning based	Node2vec	0.78	0.82	0.82	0.82	0.82	0.81	0.81	0.81	0.82	0.82

precision score is given by the following formula

$$MAP = \frac{1}{N_q} \sum_{i=1}^{N_q} AP_i \tag{3.16}$$

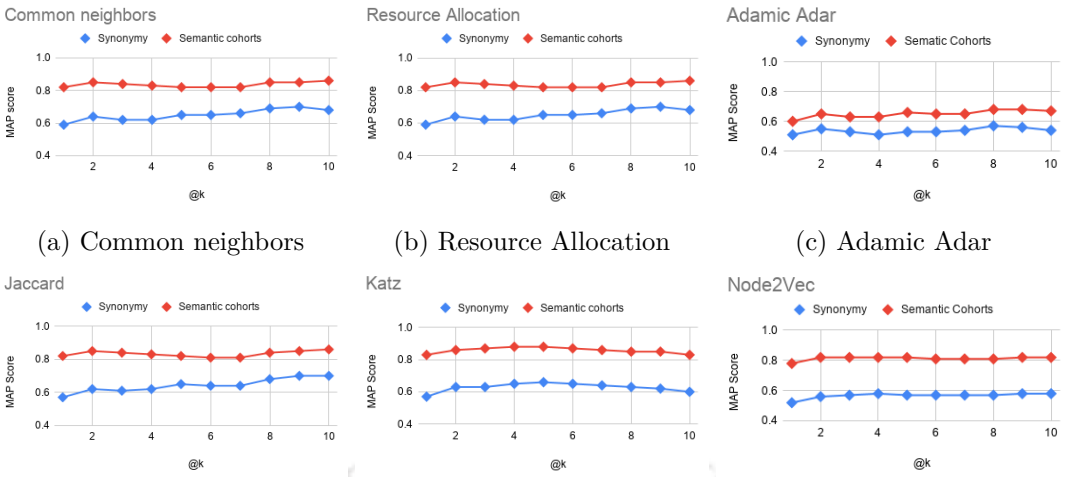
Here, N_q is the number of queries and the average precision of a query at the position k is given by the following formula

$$AP@k = \frac{1}{N_{GTP}} \sum_{i}^{k} precision@i * recall@i \tag{3.17}$$

Where $N_{GTP}$is refers total number of ground truth positives, and i refers the positions.

Table 3.5, Table 3.6 and Figure 3.4 present MAP scores of different predictors for retrieving synonymy (absolute and near synonymy) and semantic cohort words over 500 queries. From Table 3.5, it is observed that all link predictors could predict new synonymy relations in the WordNet with more than 50% minimum accuracy rate, which indicates the suitability of the Link prediction method to predict missing synonymy relations in a synonymy network. It also confirms that Assamese WordNet has missing synonymy relations.

In Table 3.5, we observed that among all link predictors, local topology-based predictors other than Adamic Adar performs better compared to global predictors, namely Katz and Node2Vec (Learning-based) in top 10 predictions. An important observation is that in the case of Katz, newly predicted words having a path length of 3 with the



(a) Common neighbors (b) Resource Allocation (c) Adamic Adar

(d) Jaccard (e) Katz (f) Node2vec

Figure 3.4: Line chart comparison of MAP score of the predictors in predicting top 10 synonymy and semantic cohort relations

target word are mostly semantic cohorts rather synonymous, which also may be a cause of local predictors performing better than a global predictors. One interpretation of this could be that the quality of synonymy decreases when path length between two words increases. Another important observation is that in the case of learning based link predictors(Node2vec), all newly predicted words which are not connected via a path with target word are noisy. It implies that if two words are synonymous in a synonymy network, they generally share at least one common relation, or they are connected via a path.

Further, we see that among local similarity based link predictors, the common neighbor gives the best accuracy in predicting new synonymy relations. This infers that the synonymy relations generally exist in two-hop proximity, and synonymy relations between two words can be better defined by the total number of common words between the two words.

From Table 3.5, it is evident that link predictors can achieve a MAP@10 score up to 0.7. It means that for every top 10 missing relations in the existing Assamese WordNet, 7 of them are synonymy. It is interesting to see that the MAP scores are comparable across the top 10 ranks. It means that correctly predicted synonyms are either at the top ranks or proportionately distributed. Further, in Table 3.6, we show the MAP scores for the relations, including the semantic cohort. It shows that the MAP score improves up to 0.86. It means that even is the retrieved results are not synonymy, they

Table 3.7: Mean Average Precision for WordNet and EAW

	WordNet	Extended Assamese WordNet
Mean Average Precision	0.99	0.95
Coverage	7.03	13.19

are semantically related∗. Interestingly, the MAP@1 is also improved up to 0.82 from 0.59.

From the above observations, a few important points are noted. (i) The creation of a WordNet (Assamese) by expanding another wordnet (Hindi) may result in missing a considerable number of relations. (ii) Link prediction methods can be effectively used to expand the synonymy network. (iii) Related words are closely connected. (iv) For discovering missing relation from synonymy network, simple local proximity based methods might be more effective.

3.6 Potential Application- Sentiment lexicon generation

To evaluate the effectiveness of new synonymy relations predicted by link prediction methods in this study, we attempt to assess the performance in sentiment lexicon generation task. We first expanded the original Assamese WordNet by adding new synonymy relations and named it Extended Assamese WordNet (EAW). Here we exploit the best performing link predictor, namely Jaccard-Coefficient, to predict the new synonymy relations. We consider the top 20 relations derived from this method, which we added to EAW. After that, we compare the performance of EAW against the original Assamese WordNet by applying it in the Sentiment lexicon generation task. As we have not found any automated tool for generating sentiment lexicons, we performed this task manually.

Sentiment lexicon is a collection of words associated with their sentiment orientation. Sentiment lexicon is an essential tool for identifying the sentiment polarity of words and texts. A straightforward approach for generating sentiment lexicon is to exploit the dictionary based method. This method generally integrates predefined resources, such as the WordNet, to construct sentiment lexicon. It first labels a set of seed words by

∗Semantically related words are words that are connected in meaning but may not be identical in meaning. For example, "car" and "vehicle" are semantically related because they are both means of transportation and share a similar concept, but they are not synonyms as they have slightly different meanings and contexts of use.

their polarity, and then extends the sets by adding synonyms to each word. There are several state of the art methods that used synonymy obtained from WordNet as a feature in different learning-based models to generate sentiment lexicon. As synonyms play an important role in sentiment lexicon generation task, WordNet is a crucial resource. In this evaluation, we have applied EAW and WordNet individually to generate a sentiment lexicon. Our objective is to compare the effectiveness of Extended Assamese WordNet and the original Assamese WordNet.

To perform this task, we randomly select 100 seed adjectives from the WordNet dataset, with known sentiment orientation. Following that the lexicon was enhanced by adding new sets of synonymy by searching with the original Assamese WordNet and EAW. It first searched synset of the target word and added all synonyms in the synset to the seed list. Table 3.8 shows a few examples of sentiment lexicon generated from the original Assamese WordNet and the Extended Assamese WordNet. We manually evaluate the sentiment lexicon generated by the original Assamese WordNet and EAW. Then we calculate the mean average precision for each WordNet. Table 3.7 present the MAP scores of WordNet and EAW in sentiment lexicon generation Finally, the following observations are derived.

1. Coverage is increased: From the manual evaluation, it has been observed that the average coverage given by original Assamese WordNet is 7.03 for generating sentiment lexicon. However coverage improves up to 13.19 while using EAW as a resource.

2. Sensitivity to sentiment orientation: We observed that new synonymy relations are sensitive to the sentiment orientation of the seed word. That is, the new synonymy relations derived using EAW conform to the sentiment orientation of seed. While the generated set may not be absolute synonymy, they are words of the same sentiment orientation.

From the above observation, it is evident that Extended Assamese WordNet is able to generate more synonymy relations of same sentiment polarity.

Table 3.8: Sentiment lexicon generated from WordNet and EWA

Seed word	Sentiment lexicon propageted from WordNet	Sentiment lexicon propageted from EWA	Increasing coverage
ফেইল	অনুত্তীর্ণ, অকৃতকার্য, অসফল	অনুত্তীর্ণ, অকৃতকার্য, অসফল বিফল , ব্যর্থ , নিষ্ফল , অকৃতার্থ , অকৃতকার্য্য , অকৃতী	6
অনৈতিক	অনৈতিক, নৈতিকতাহীন, অনুচিত, নীতিবিৰুদ্ধ, অনীতিপূর্ণ	অনৈতিক, নৈতিকতাহীন, অনুচিত, নীতিবিৰুদ্ধ, অনীতিপূর্ণ অন্যায় , অযথার্থ , গর্হিত , অসৎ , অযুক্তত , অসংগত , অসমীচীন ন্যায়বিৰুদ্ধ , অনৈতিকতা , অনীতি , নীতিহীনতা , অধম , নীচ	13
দন্দুৰী	উগ্রা, দন্দুৰী, কাজিয়াখোৰ	উগ্রা, দন্দুৰী, কাজিয়াখোৰ কন্দলীয়া , দন্দখোৰ , কলহপ্রিয় , ্কর্কশ , কুতর্কী , বিতণ্ডাবাদী	6
মিছা	মিছা, অসত্য, মিথ্যা, অপ্রকৃত, ফাঁকি, ফাকি-ফুকা, ফাকতি, অনৃত	মনে-সজা , মনে-গঢ়া মিছা, অসত্য, মিথ্যা, অপ্রকৃত, ফাঁকি, ফাকি-ফুকা, ফাকতি, অনৃত	3
সহজ	সহজ, সৰল, সুগম, অজটিল, সহজ-সৰল, উজু, সহজসাধ্য, অনায়াসসাধ্য, ঢিলা, সুকৰ, সুখসাধ্য	সামান্যতা , বাধাহীন সহজ, সৰল, সুগম, অজটিল, সহজ-সৰল, উজু, সহজসাধ্য, অনায়াসসাধ্য, ঢিলা, সুকৰ, সুখসাধ্য	6
জ্ঞানী	পণ্ডিত, জ্ঞানী, বিদ্বান	অভিজ্ঞ , শিক্ষিত , সাক্ষৰ , লিখা পঢ়াজনা- , পঢ়া লিখা জনা পণ্ডিত, জ্ঞানী, বিদ্বান	5
পূজনীয়	পূজনীয়, পূজ্য, উপাস্য, আৰাধ্য, বন্দনীয়	পূজ্যতা , পূজনীয়তা , পূজাৰ্হ পূজনীয়, পূজ্য, উপাস্য, আৰাধ্য, বন্দনীয়	3
আশ্রয়হীন	নিৰাশ্রয়ী, আশ্রয়হীন, অনাশ্রিত	অনাশ্রয় , নাথহীন নিৰাশ্রয়ী, আশ্রয়হীন, অনাশ্রিত	2
ভাল	সজ্জন, সৎ, সাধু, সুজন, ভাল	সৎগুণী , সদ্গুণী , গুণশালী , গুনৱন্ত , সৎপুৰুষ , সজ , সদাচাৰী , সত্যাচাৰী , সভ্য , সজ্জন, সৎ, সাধু, সুজন, ভাল	9

3.7 Summary

Assamese WordNet created as a part of IndoWordNet[13] is the only Wordnet available for Assamese language. It is generated by mapping Hindi WordNet to corresponding Assamese equivalent words. As a result, Assamese WordNet not only has low synonymy coverage (it has 4.79 synonyms per words, while its parent Hindi WordNet has 9.34 synonyms per word), but also influenced by Hindi WordNet. This study investigates the effectiveness of applying different link prediction methods for discovering missing synonymy relationships. The study investigates the performance of eleven link prediction methods ranging from the simplest local proximity-based common neighbor (CN) to the recent node embedding approaches. As synonymy relations tend to satisfy triadic property of a complex network, from various experimental observation, it is observed that the simple local proximity-based methods such as CN, RA, AA, JC are effective enough, sometimes superior, to discover missing synonymy as compared to global and complex model supervised models using network embedding. We observed the above finding from two sets of experimental setups. First, we remove edges from the synonymy network and predict the removed network using link predictors. We are able to predict up to almost 99% accuracy using simple predictors like CN. Second, we apply link predictors over the entire network and predict actually missing synonymy relations using 500 randomly selected Assamese words. We are able to achieve a *MAP*@10 score of 0.68 with CN. It is interesting to observe that *MAP*@1 score of CN is 0.59. It means that about 60% of the predictions using CN are actual missing synonyms. It is further observed that the majority of the predicted non-synonymy words are also actually se-

mantically related cohort words (with *MAP*@1 score of 0.82 and *MAP*@10 score of 0.86 with CN).

From the study reported in this chapter, it is evident that synonymy network can be effectively expanded using link prediction methods. However, relationships may be of different types; *absolute, near, cohort, polysemy*, etc., and it has not been considered within the scope of the thesis.

In this chapter, we covered the problem of expanding synonymy network of WordNet to include missing relationship in existing WordNet. Although this method can effectively expand new synonymy links, it does not address the issue of adding new words to the WordNet taxonomy. To fully expand the WordNet taxonomy, it is necessary to consider not only new relationships, but also new words. In the next chapter, we will explore the problem of adding new words to the WordNet taxonomy

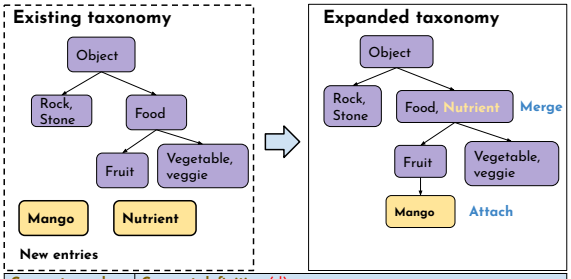
"Never doubt, when you begin with something that it will end in failure. Our thought is transformed as picture in our mind."

Napz Cherub Pellazo

4

TEAM: A multitask learning based Taxonomy Expansion approach for Attach and Merge

The previous chapter asses the effectiveness of link prediction method to perform synonymy expansion in WordNet with a special focus to Assamese WordNet. The objective of this research is to identify any potential missing relationships in WordNet. From various experiment it is evident that synonymy network can be effectively expanded using link prediction methods. However, this method can not expand the taxonomy with new word . As new concepts continually emerge and evolve, they may need to be incorporated into existing taxonomies to keep them relevant and useful. Automatic taxonomy expansion is therefore a crucial task. Most of the taxonomy expansion ap-



Concept- words Synset (s)	Concept definition (d)
Object	A tangible and visible entity.
Rock, Stone	A lump or mass of hard consolidated mineral matter.
Food	Any substance that can be metabolized by an organism to give energy and build tissue.
Vegetable,Veggie	Edible seeds or roots or stems or leaves or bulbs or tubers or non sweet fruits of any of numerous herbaceous plant
Fruit	The ripened reproductive body of a seed plant.
Nutrient	A substance used by an organism to survive, grow, and reproduce.
Mango	A mango is an edible stone fruit.

Figure 4.1: Example of WordNet taxonomy expansion with *attach* and *merge* operations to include new terms *"Mango"* and *"Nutrient"*.
"Mango" is a specific concept of *Fruit* not present in the existing WordNet. Hence, a new concept node is created in the taxonomy by attaching it to its generic concept *Fruit* . As *"Nutrient"* refers to the same concept as *"Food"*, no new concept is created. *"Nutrient"* is merged with the existing concept *"Food"*.

proaches are of two types, *attach* and *merge*. In a taxonomy like WordNet, both merge and attach are integral parts of the expansion operations, but the majority of studies consider them separately. With a keen eye towards the significance of this issue, this chapter explores into a novel approach called *"multitask learning-based taxonomy expansion approach for Attach and Merge" (TEAM)* to integrate new concepts into WordNet taxonomy. In this chapter, the scope of the study has been expanded to include not only the Assamese WordNet but also the Bengali and Hindi WordNets from IndoWordNet.

4.1 Introduction

Taxonomy, such as the WordNet, is a crucial resource for developing NLP related technologies, as it plays a vital role in various text processing tasks such as information retrieval, information extraction, text classification, summarization, etc.[61,3,80 51]. As most of the WordNets are manually curated, it often suffers from the problem of limited coverage. Therefore, an automatic taxonomy expansion is a crucial problem to handle the above issue. For taxonomy expansion, WordNet in particular, may need two types of operations; (i) *merge*, where a new concept ∗ is merged to an existing node, and (ii) *attach*, where a new concept is inserted as a new node. Figure 4.1 illustrates

∗Concept is a basic building block of WordNet, which refers a definition with associated synonym words

these two operations where the word *Mango* is inserted as a new concept with the *attach* operation, and the word *Nutrient* is inserted as a new synonymy in an existing concept with the *merge* operation.

Though both of these operations are integral parts of a WordNet taxonomy expansion, all of the existing studies on taxonomy expansion have considered expansion with either *attach* operation[74,90,78,104,105,85,44] or *merge* operation[55,57,55,69,11,94,19], but not together. Realizing the need to apply both the operation, SemEval-2016:task 14 (Semantic taxonomy enrichment) Jurgens and Pilehvar[32] includes a call for expansion with both *attach* and *merge* operations. However, none of the submissions incorporate both operations in a single model.

Motivated by the above observations, in this study, we propose an integrated deep learning-based method, namely, *Taxonomy Expansion with Attach and Merge* (TEAM), which performs both the *attach* and *merge* operations in a multitask-learning framework. Though most of the existing studies consider the expansion a regression problem[78,104,105], considering that our method performs both the *attach* and *merge* operation in a single model, it can also be considered a classification task. As a result, we propose two versions of TEAM, namely, TEAM-RG: *Regression*, and TEAM-CL: *Classification* to perform with explicit and implicit rankings. The proposed models have been evaluated on three different WordNet taxonomies, viz., Assamese, Bangla, and Hindi. From the various experimental setups, it is observed that the proposed TEAM-RG and TEAM-CL outperform their baselines counterparts for *attach* operation, and also obtained encouraging performance for *merge* operation as well. The major contributions of this chapter are summarized as follows:

- A multi-task learning based taxonomy expansion framework TEAM is jointly trained to perform both the Attach and Merge operations. To the best of our knowledge, it is the first integrated model to perform both the *attach* and Merge operations in a single model.

- Two variants of TEAM, namely TEAM-Regression (RG) and TEAM-Classification (CL) are proposed.

4.2 Related studies

Existing methods for taxonomy expansion can be divided into two categories: relying on alignment between multiple taxonomies [Ruiz-Casado et al.[72], Toral et al.[87], Ponzetto and Navigli[65], and Yamada et al.[101]] or relying on machine learning-based rating subgraphs. Further, the latter category can be divided into two sub-categories (1) by expanding synonymy relations/Merge (2) by expanding hypernymy relations/Attach. Synonymy-based taxonomy expansion leverages synonymy relations of the taxonomy. Given a seed taxonomy, the distributional approach discovers synonyms by representing strings with their distributional feature and learning a classifier to predict the relation between strings [[55], [94], [19]].

Expansion by resource alignment: In the first category of studies, Poprat et al.[66] first attempted to automatically expand a WordNet with biomedical terminology; however, they were unable in developing the resource. Ruiz-Casado et al.[72], Toral et al.[87], Ponzetto and Navigli[65], and Yamada et al.[101] exploit structured information in Wikipedia to expand WordNet with new synsets. Snow et al.[82] leverage distributional similarity techniques for WordNet expansion. Jurgens and Pilehvar[31] enrich the existing WordNet taxonomy using an additional resource, Wiktionary, to extract sense data based on information in the term concepts.

Synonymy Expansion: Synonymy expansion in a taxonomy leverages synonymy relations to enrich a taxonomy with new concepts. Approaches for synonymy expansion can be divided in to two categories: (1) Distributional based approach[94], [19] (2) Pattern-based approach[57], [55]. Given a seed taxonomy, the distributional approach discovers synonyms by representing strings with their distributional feature and learning a classifier to predict the relation between strings. However, in the pattern-based approach, consider the sentences mentioning a pair of synonymous strings and learn some textual patterns from these sentences, which are further used to discover more synonyms. Qu et al.[69] proposed an approach that integrates both the categories. Boteanu et al.[11] focus on the problem of expanding taxonomies with synonyms for applications in which entities are complex concepts arranged into taxonomies designed to facilitate browsing the product catalog on amazon.com. They first generate synonymy candidates for each

node in the taxonomy and then filter synonymy candidates using a binary classifier. Yu et al.[103] study a task of synonym expansion using transitivity named SYNET, which leverages both the contexts of two synonymy pairs.

Hypernymy expansion : Jurgens and Pilehvar[32] formulated a task of synonymy expansion, where it is proposed to enrich the WordNet taxonomy by performing two operations for each new concept. The first action is *attach*, where a new concept is treated as a new synset and is attached as a hyponym of one existing synset in WordNet, and the second action is Merge, where a new concept is merged into an existing synset. The best solution proposed by Schlichtkrull and Alonso[74] included only the *attach* operation. Later solutions for attaching, as in Shen et al.[78], adopted self-supervision and tried to exploit the information of nodes in the seed taxonomy to perform node pair matching. On the other hand, Yu et al.[104] resorted to classification along mini paths in the taxonomy. In contrast, in our current approach, we have incorporated both the *attach* and *merge* operations.

Most of the recent taxonomy expansion approaches are based on hypernymy expansion. These methods attempt to determine the attachment position by scoring between several nodes. Recently numerous methods have been proposed to solve this problem[79],[78],[104],[105],[44]. [78]. Hence, all the existing taxonomy expansion approaches expand a taxonomy either by *merge* operation(synonymy expansion) or by *attach* operation(hypernymy expansion). However, particular to WordNet expansion it is an integrated task of *merge* and *attach* operation. We are the first to study the problem of taxonomy expansion using both the Attach and Merge taxonomy expansion operations in a single model.

4.3 Taxonomy Expansion - Attach and Merge

In this study, we have considered WordNets as our target taxonomies. A WordNet may be defined by a collection of concepts connected by various semantic relationships such as *hypernymy, hyponymy, troponymy*, etc., where each concept is further defined by a set of attributes such as *definition, synonyms, examples*, etc[7]. In this study, we have considered only the hypernymy relation and the definition and synonymy attributes.

In order to be able to apply the proposed model, we first transform the original

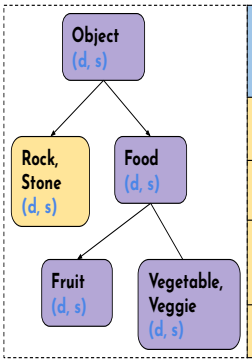
WordNet taxonomy into an experimental intermediate taxonomy (directed unweighted acyclic graph) $\mathcal{T} = (V, E)$ where V represents the set of concepts and E represents the set of hypernymy relations between the concepts. A concept $v \in V$ is further defined by a tuple $v = (d_v, s_v)$ where d_v represents the *definition* of the concept, and s_v represents the set of associated synonyms. An edge $e \in E$ represents a hypernymy relation from a parent concept v_p to its child concept v_{ch} and is denoted as $e : (v_p \xrightarrow{\text{hyper}} v_{ch})$. The taxonomy $\mathcal{T}$ is arranged in a hierarchical manner with directed edges in E, as shown in Figure 4.1. Given the taxonomy $\mathcal{T}$ and a query concept $q = (d_q, s_q)$, the *attach* and the *merge* expansion operations are defined below.

Attach (A) — An *attach* operation is performed when the concept q is not present in $\mathcal{T}$. The objective of the *attach* operation is to identify the best matching parent node in taxonomy network known as anchor concept $a = (d_a, s_a)$, and insert a new concept q with an edge $e : (a \xrightarrow{\text{hyper}} q)$. In a taxonomy network, a parent node represents a more generic concept of its children. After an *attach* operation i.e., insertion of q in $\mathcal{T}$ under the anchor a, the expanded taxonomy is updated as follows.

$$\mathcal{T} = (V \cup \{q\}, E \cup \{e\}) \tag{4.1}$$

Merge (M) — A *merge* operation is performed when an equivalent concept $a = (d_a, s_a)$ of the query q (i.e., $d_a \equiv d_q$) is already present in $\mathcal{T}$, but the synset s_q is not present in a (i.e., $s_q \cap s_a = \emptyset$). The objective of the *merge* operation is to identify the best matching concept $a = (d_a, s_a)$, known as the anchor concept, in the taxonomy network $\mathcal{T}$ and add the synset s_q to s_a. It neither creates a new node nor adds a new edge. It only updates the synset of the anchor concept. After the *merge* operation, the updated anchor concept in the expanded taxonomy can be expressed as follows.

$$a = (d_a, s_a \cup s_q) : a \in V \tag{4.2}$$



Query	Anchor	Label	Operation
Rock(d, "rock")	Object (d, s)	True	Attach
Rock(d, "rock")	Fruit (d,s)	False	
Rock(d, "rock")	Stone (parallel_d, s)	True	Merge
Rock(d, "rock")	Fruit(d, s)	False	

Figure 4.2: Example of training dataset generation.
The table shows positive and negative training instances corresponding to the query concept *"Rock"* for both operations *Attach* and *Merge*.

4.4 Proposed Methods

Our objective is to develop an integrated model that performs both *attach* and *merge* operations for taxonomy expansion. Since we have two tasks to unify in a single model, we resort to a multi-task learning framework known as **T**axonomy **E**xpansion Framework with **A**ttach and **M**erge (TEAM). This joint learning objective facilitates information flow so that the two tasks can aid each other. Also, we are interested in deciding which expansion operation to perform given a triplet (expansion task classification) and retrieving the ranked list of candidates (ranking) as prospective anchors to associate the query with. For this first-of-its-kind novel taxonomy expansion task, we propose two versions of TEAM, namely *TEAM–Regression* (TEAM-RG) and *TEAM–Classification* (TEAM-CL) — where we show that using either regression or classification learning objectives, this task can be accomplished.

4.4.1 Training dataset generation

Given a transformed taxonomy $\mathcal{T}$ (as described in Section 4.3), we generate a training dataset for building the model as follows. The training samples are defined by a 3-tuple $< q, a,$ label $>$, where q is the query, a is the potential anchor, and label is associated class, i.e., true/false (1/0). We randomly select a set of nodes in $\mathcal{T}$ as a set of queries $*$, and generate the training samples for the *attach* and the *merge* operations separately as follows.

$*$As we consider the same query set for both *attach* and *merge* experiments, nodes with at least two synonyms are considered.

Attach (A) — We first remove the query nodes from the $\mathcal{T}$. For each query $q = (d_q, s_q)$, we consider its parent as anchor node $a = (d_a, s_a)$ and generate positive sample $< q, a, TRUE >$. We then randomly pick up $\mathcal{N}$ number other nodes $a' = (d_{a'}, s_{a'})$, and generate N negative samples $< q, a', FALSE >$. Thus, for a given query node q, we extract one positive and $\mathcal{N}$ negative samples.

Merge (M) — For each of the randomly selected query node $x = (d_x, s_x >)$ in $\mathcal{T}$, we generate the following positive training sample $< q, x, True >$ where $q = (d_x, s_q), s_q \subset s_x$ is the query and $x = (d_x, s_x - s_q)$ is the anchor.The s_q is a randomly selected synonym in s_x. Unlike *attach*, for generating the training sample for the query q, we only remove the query synset s_q from the anchor synset s_x i.e., $s_x = s_x - s_q$, and, not the node. Like *attach*, we randomly pick up $\mathcal{N}$ number other nodes a', and generate N negative samples $< q, a', FALSE >$. Figure 4.2 illustrates the generation of the training samples from a taxonomy.

4.4.2 TEAM-Regression (TEAM-RG)

The proposed TEAM-RG works in two tiers process. Given a training input sample $< q, a, c >$, it first generates encoding of the query q and the anchor a. It then merges to a shared layer to produce two different multi-tasking dense networks; one for *merge* and another for *attach*, as shown in figure 4.4.D.

For learning embedding of the anchor concept from the taxonomy network and the query concept from the associated attributes, we consider the publicly available Fasttext pre-trained embedding available at https://fasttext.cc/docs/en/crawl-vectors.html.

Processing of the query concept: As mentioned in Section 4.3, a query concept consists of its definition and the associated synset i.e., $q = (d_q, s_q)$. The definition is a piece of text describing the concept, and the synset is a synonym associated with the query concept. The two embeddings are then concatenated to represent the query.

Processing of the anchor concept: For generating the encoding of the anchor concept, we exploit the proximity structure of the nodes in the taxonomy $\mathcal{T}$. For a given anchor node $a \in \mathcal{T}$, we first extract its ego-tree from the taxonomy. An ego tree $\mathcal{T}_a : (V_a, E_a)$ of a node a in the taxonomy $\mathcal{T}$ is a sub-tree that comprises the node a and its k-hop neighborhood nodes. In this study, we considered $k = 1$, i.e., the anchor

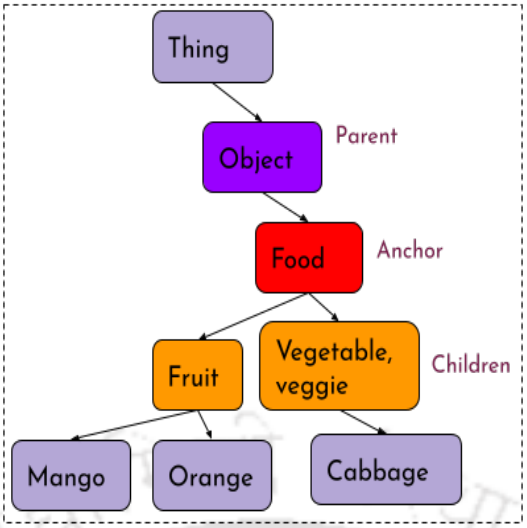


Figure 4.3: Ego tree of the anchor node *"Food"*. 1-hop ego-tree is extracted around the anchor *"Food"*. The color-codes distinguish various roles w.r.t the anchor node *"Food"*, eg., Deep Purple: Grand-parent, Red: Anchor/ Parent, Orange: Childrens

node, its parent node, and all its children nodes. Figure 4.3 illustrates an example of an ego tree. A similar approach has also been used in [96,104,105,78] studies. To obtain the embedding of the anchor concept, we further apply graph embedding as described below.

Embedding Ego-tree

Ideally, we should be able to use any graph embedding method to obtain the embedding of the anchor node. As the objective is to incorporate the positional information of the parent and children node in the ego tree, we use the Graph Attention Network (GAT) proposed in Taxo-Expan[78]. This GAT is a special type of graph neural network (GNN)[34] with a neighborhood-based attention mechanism. The details of GAT and its difference from GNN are given in Section 2.2.3 of Background study. Thus we used position enhanced GAT to obtain the node embeddings of an anchor's ego tree.

We summarize the tree by applying an activation function over the average of the embedding vectors of all nodes in the ego-tree as given in equation 5.1 to define the encoding of the anchor node.

$$\bar{\mathcal{T}}_a = \sigma\left(\frac{1}{|V_a|}\sum_{x \in V_a} \bar{x}\right) \tag{4.3}$$

where $\sigma(.)$ is an activation function. We have considered *Sigmoid* function in this study.

Multi-task Learning

Once we obtain the embeddings of the anchor and query concepts, the concatenated vector is subjected to a shared dense layer and then build two multi-task layers to perform the *merge* and *attach* operations as shown in Figure 4.4.D. Given a query concept and its true anchor concept with $\mathcal{N}$ false anchor concepts, the task is to design a regression-based ranking model such that the true concept is ranked higher than the $\mathcal{N}$ false concepts. This objective should be realized for all the queries in the training dataset.

Given the embedding vectors of anchor $\bar{a}$ and query $\bar{q}$ as learned above, we first estimate similarity between the two using a bi-linear model proposed in[23]. It learns the discrimination between q and a through a learnable bi-linear scoring matrix $B \in \mathbb{R}^{|\bar{q}|\times|\bar{a}|}$ via a function $\mathcal{D} : \mathbb{R}^{|\bar{q}|\times|\bar{a}|} \mapsto \mathbb{R}$ as follows.

$$\mathcal{D}(q, a) = \sigma(\bar{q}^T B \bar{a}) \tag{4.4}$$

Here σ is sigmoid non-linearity. The output of this matching module is a probability estimate indicating the strength of association between the query and anchor. Now, considering the query concept q and its associated $\mathcal{N}+1$ anchor concepts, we estimate the probability of being the correct anchor using InfoNCE loss proposed in[58]. Let $\mathcal{X}$ be a set of query concepts and their respective $\mathcal{N}+1$ anchor nodes (one positive and $\mathcal{N}$ negative). An element of $x_q \in \mathcal{X}$ for a given query q consists of $\{(q, a, 1), (q, a'_1, a'_2, ..., a'_{\mathcal{N}}, 0)\}$, where a is the positive anchor, and a' are the negative anchors of q. InfoNCE estimates loss function using an average probability of being true anchor node across the dataset $\mathcal{X}$ as follows.

$$\mathcal{L}_{A/M} = -\frac{1}{|\mathcal{X}|} \sum_{x_q \in \mathcal{X}} log \frac{\mathcal{D}(\bar{q}, \bar{a})}{\sum_{v \in \mathcal{M}(q)} \mathcal{D}(\bar{q}, \bar{v})} \tag{4.5}$$

where $\mathcal{M}(q)$ denotes the set of both positive and negative anchors of q. As mentioned earlier, the loss defined in Equation 5.6 is estimated separately for *attach* and *merge* operations. Therefore, we generate two different training datasets for *attach* and *merge*, and estimate $\mathcal{L}_A$ and $\mathcal{L}_M$ separately using respective datasets, The final model loss is

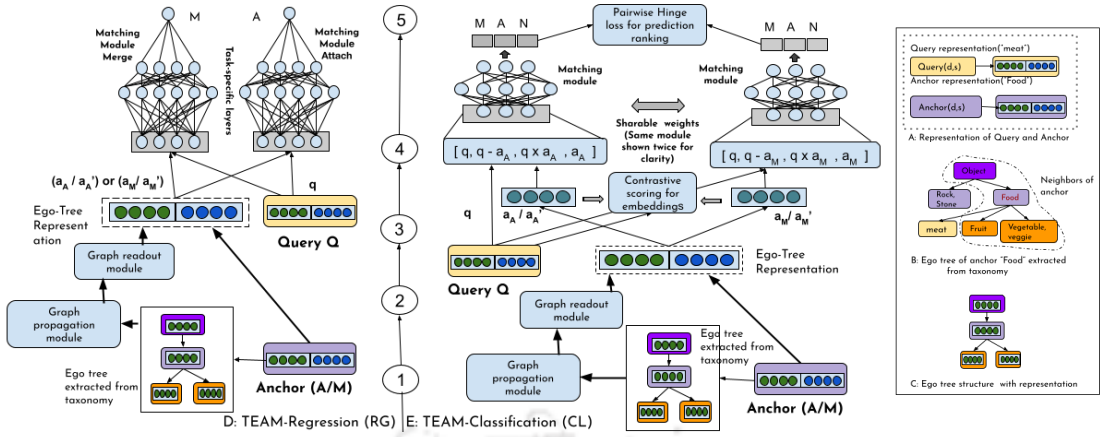


Figure 4.4: Taxonomy Expansion framework with Attach and Merge (TEAM) **D: TEAM-Regression-RG** — • 1. (Query Q, Anchor, (N+1) Negative Anchors) are fed to the model, • 2. Representation learning via shared graph propagation and readout modules, • 3. Projection to shared hidden layers, • 4. Task-specific matching modules with non-shareable weights, • 5. Task-specific regression outputs. **E: TEAM-Classification (CL)** — • 1. (Query Q, Anchor-A, Anchor-M, (N+1) Negative Anchors) are fed to the model, • 2. Representation learning via shared graph propagation and readout modules, • 3. Projection to shared hidden layers, • 4. Simultaneous optimization of classification and ranking losses. • 5.Three-way *merge*, *attach*, no-operation (M, A, N) prediction.
Explanation of used color-codes. Light-Purple: Anchor (A/M) nodes, Orange: Children of the anchor, Purple: Parent of the anchor. Green: Definition representation of anchors, Blue: Synset representation of anchors, Yellow: Query representation.

defined as $\mathcal{L} = \mathcal{L}_A + \mathcal{L}_M$ — considering both the operations *attach* and *merge*.

4.4.3 TEAM-Classification (TEAM-CL)

Figure 4.4E shows the schematic diagram of the TEAM-CL. We use the identical representations for query q and candidate anchors a_A, a_M as described for TEAM-RG. We also adopt the same position-enhanced graph propagation and read-out modules as described in Section 4.4.2 for learning anchor $a = (d_a, s_a)$ concept representation. Once we obtain the query and anchor representations, we model the strength of association of an input query and the candidate anchors based on their features to predict the expansion task i.e., *merge* M or *attach* A. The matching module, a multi-layer perceptron (MLP) based classifier, takes the features of query $\bar{q} \in \mathbb{R}^{|\bar{q}|}$ and anchor $\bar{a} \in \mathbb{R}^{|\bar{a}|}$, and generates a contextualized pair representation $\bar{k} = [\bar{q} \oplus (\bar{q} - \bar{a}) \oplus (\bar{q} \times \bar{a}) \oplus \bar{a}]$ (assuming $|\bar{q}| = |\bar{a}|$). Here, $\oplus$ denotes concatenation. The anchor a can be any of the *attach* or *merge* candidates (a_A/a_M). A three-way classifier is learned to produce the categorical probability distribution over the training samples for Merge (M), Attach (A) and No-operation (N) — three classes ($|\mathcal{Z}| = 3$) of operations. If $\theta \in \mathbb{R}^{|\bar{k}| \times |\bar{\mathcal{Z}}|}$ be a learnable projection matrix that projects the contextualized pair embedding $\bar{k}$ to the label space $\mathcal{Z} \in \mathbb{R}^3$. The predictions

are obtained as below,

$$\hat{Y} = \text{softmax}(\text{MLP}(\bar{k}; \theta)) \tag{4.6}$$

For two versions of TEAM, we chose two different kinds of matching models based on empirical performances to capture different kinds of embedding interaction in the latent space.

Multi-task Learning

Classification. Unlike in TEAM-RG, where we posit taxonomy expansion as a regression task with implicit ranking viz. discriminating true and false examples via InfoNCE loss, in TEAM-CL, we simultaneously optimize for classification and explicit ranking objectives. We obtain classification predictions from the matching module as described before. Given a training set $\mathcal{X}$, and a set of classes $\mathcal{Z}$ (M: Merge, A: Attach, N: No-operation), we optimize for the self-supervised cross-entropy loss over the task predictions $\hat{Y}$ given the ground-truth task-classes Y for an input query-anchor pair.

$$\mathcal{L}_C = -\frac{1}{|\mathcal{X}|} \sum_{i \in \mathcal{X}} \sum_{z \in \mathcal{Z}} Y_{iz} ln \hat{Y}_{iz} \tag{4.7}$$

Ranking. The classification objective can only learn and infer the confidence score of an operation (M/ A/ N) for a training sample. It fails to give us a reliable ranked list of prospective anchors-(A/ M) given a query — since it does not learn the relative ranks of positive and negative anchors for a query. As illustrated in Figure 4.4, for a query q, (i) the ego-tree of anchor-A comprises of that query's parent's hierarchical neighborhood, and (ii) the ego-tree of anchor-M comprises of that query's replica's (same/similar definition with a missing portion of synset) hierarchical neighborhood. Since a query q is very similar to both of its anchor-A and anchor-M's ego-trees – these operations are hardly distinguishable. Thus, a model must accommodate a provision for directly comparing the prediction scores of M and A operations and learning a margin of separation between the scores. Here, we introduce two ranking objectives in the framework — (i) a contrastive objective to compare and contrast among a positive

anchor and $\mathcal{N}$ negative anchors, (ii) a pair-wise hinge loss to learn a maximum margin between the M and A prediction scores.

Let, *dist*(.) be a function to measure the distance between a query $\bar{q}$ and its true/ false anchor-(A/ M) representations $(\bar{a_A}, \bar{a_A}')$, $(\bar{a_M}, \bar{a_M}')$. We use "slash" (/) to denote either. We intend to rank a positive query-anchor pair $(q, a_A/a_M)$ higher than $\mathcal{N}$ no of negative pairs $(q, a'_A/a'_M)$ by enforcing a group-wise contrastive loss using a margin λ as,

$$\mathcal{L}_{R1_A} = \frac{1}{\mathcal{X}} \sum_{i \in \mathcal{X}} \frac{1}{|\mathcal{N}(\bar{q}_i)|} \sum_{\bar{a_{A_i}'} \in \mathcal{N}(\bar{q}_i)} \max(0, \lambda - m + m') \tag{4.8}$$
$$m = -\operatorname{dist}(\bar{q}_i - \bar{a}_{A_i}),\ m' = -\operatorname{dist}(\bar{q}_i - \bar{a_{A_i}'})$$

We can similarly compute the margin-based group-wise contrastive loss $\mathcal{L}_{R1_M}$ for the Merge (M).

Now, to distinguish between M and A operations, let, $f(\bar{k})$ be a function that projects the contextualized (q, a) embedding $\bar{k}$ in Equation 4.6, to a hidden space $\mathbb{R}^b$. Here we introduce a margin-based hinge-loss on sample anchor pairs *attach-merge* $\langle a_A, a_M \rangle$ for a given query q via their contextualized vectors $\langle \bar{k_A}, \bar{k_M} \rangle$. If class labels of *merge* and *attach* are $M = 2, A = 1$, we ensure the prediction scores $\hat{Y}_{A/M} = f(\bar{k}_{A/M})$ for M and A are separated by a margin of λ.

$$\mathcal{L}_{R2} = \sum_{Y(\bar{k_A}) > Y(\bar{k_M})} \max(0, \lambda - f(\bar{k_M}) + f(\bar{k_A}))$$

Therefore, the final loss is, $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_{R1_A} + \mathcal{L}_{R1_M} + \mathcal{L}_{R2}$ — considering both margin-based group-wise contrastive loss and pairwise hinge loss comprising the overall ranking loss.

4.4.4 Model Inference

We follow Taxo-Expan's[78] evaluation strategy for inferring the best candidate anchor a given a query q. We use our classification objective to decide which operation among *merge*, *attach*, or no-operation (M, A, N) to perform when q is given. *i)* For TEAM-RG, we augment a classification layer on top of the task-specific regression layers. Given a

Table 4.1: Dataset Statistics

	Nodes	Edges	Max-in degree	Max-out degree	leaf nodes
Assamese WordNet	8466	8363	1	525	7072
Bengali WordNet	26007	25815	1	924	22847
Hindi Wordnet	28242	28016	1	951	24737

query q and a set of candidate anchors a, we obtain the *merge* and *attach* regression scores and choose the best value along with the corresponding operation as the apt operation to perform. *ii)* For TEAM-CL choosing which operation to perform is obtained based on the three-way prediction scores, given $< q, a, (0/1) >$ as input. Since both of our proposed frameworks optimize for ranking loss, i.e., discriminates true candidate pairs from the negative ones — we get a ranked list of candidate anchors a while matching each of them with q via respective matching modules.

4.5 Experiments

Datasets. Table 4.1 shows the basic statistics of three WordNet taxonomies used in this study. The taxonomy networks are extracted from Assamese, Bengali, and Hindi WordNets, respectively.

Metrics. We use an array of performance metrics from the domain of classification and ranking to evaluate the competing methods' performances. Among the ranking metrics, we use Mean Rank (MR), Hit@k (k=1, 3), and Mean Reciprocal Rank (MRR) to judge how well a competing method performs in producing a ranked list of candidate anchors given a test query and a taxonomy expansion operation to carry-out – *merge* or *attach*. Like Taxo-Expan[78] evaluation strategy, we scale the MRR score by a factor of 10 to highlight the discrepancy of the performances among different methods.

- **Mean Rank:** It calculates the average rank of true anchors among all the candidate anchors with respect to the matching scores, given a query.
- **Hit@k:** It calculates the number of times a true anchor appears in the top k positions when matched with a test query.
- **Mean Reciprocal Rank(MRR):** The Mean Reciprocal Rank is used to assess the

ranking quality of the true anchor. The reciprocal rank can be computed by finding and inversing the rank of a true anchor in the predicted anchors' list of each query. MRR is averaged over all queries.

Further, we use Accuracy, Micro/ Macro F1, Precision, Recall, and F-Scores as classification metrics for deciding given a test query and an initial taxonomy tree, which operation among *merge* (M), *attach* (A), and no-operation (N) is to be performed.

- **Accuracy:** It summarizes the performance of the classification model as the fraction of the number of true tasks predicted over the total number of ground-truth tasks for a set of queries.
- **Precision:** It calculates the fraction of true-positive predicted expansion task classes among the total number of true-positive and false-negative task classes.
- **Recall:** It calculates the fraction of true-positive predicted expansion task classes among all the relevant ground-truth task classes.
- **F-Score:** The harmonic mean of precision and recall. It is also known as F1-Score.
- **Micro/ Macro F1 :** The Macro F1 computes F1-Score for each class (*merge* M/ *attach* A) independently but averages the final score by treating each expansion task-class as equally contributing. However, Micro F1 computes the F1-Score for each query sample in the training set and therefore aggregates the contributions of all expansion task classes to compute the final average metric.

Baselines. We choose two most recent benchmark SOTA taxonomy-expansion frameworks TaxoExpan[78] and Triplet Matching Network(TMN)[105] as the competing methods. As Taxo-Expan and TMN outperform SemEval-2016[78,105], we have not included SemEval-2016 as baseline in this study. In terms of learning objective, Taxo-Expan is similar to ours. It uses ego-tree-based anchor features for matching query features in a regression-based setting. TMN captures fine-grained relationship dynamics of query and anchor concepts using channel-wise gating mechanism-based attention learning.

Evaluation Strategy. We obtain the initial feature vector for train and test concepts using pre-trained subword-aware Fasttext embeddings. For each concept, we generate its definition embedding by averaging the embedding of each word in its textual definition.

Table 4.2: Ranking results for test queries

	Assamese WordNet-Noun				Bengali WordNet-Noun				Hindi WordNet-Noun			
Methods	Micro_MR	Hit@1	Hit@3	MRR	Micro_MR	Hit@1	Hit@3	MRR	Micro_MR	Hit@1	Hit@3	MRR
TEAM-RG(A)	**144.92**	0.27	**0.42**	**0.67**	**191.51**	**0.17**	**0.36**	**0.86**	**177.85**	0.28	**0.43**	**0.67**
TEAM-CL(A)	189.75	0.16	0.28	0.57	277.01	0.05	0.18	0.50	220.98	0.13	0.29	0.54
Taxo-Expan(A)	341.81	0.07	0.11	0.29	679.26	0.03	0.04	0.10	648.72	0.04	0.08	0.14
TMN(A)	203.28	**0.28**	0.41	0.63	319.36	0.10	0.15	0.69	246.45	**0.31**	0.25	0.61
TEAM-RG(M)	**1.27**	**0.95**	**0.98**	**1.00**	**2.04**	**0.92**	**0.98**	**1.00**	**5.38**	**0.83**	**0.88**	**0.95**
TEAM-CL(M)	6.44	0.71	0.82	0.93	11.06	0.61	0.75	0.89	9.80	0.64	0.71	0.91
TEAM-RG(MA)	**73.34**	**0.61**	**0.70**	**0.83**	**59.81**	**0.69**	**0.80**	**0.85**	**91.62**	**0.63**	**0.71**	**0.81**
TEAM-CL(MA)	99.00	0.37	0.50	0.74	144.38	0.33	0.46	0.70	113.39	0.32	0.47	0.73

We employ PyTorch and DGL framework ∗ to load and train embeddings. In TEAM, we use a two-layer position-enhanced GAT where the first layer (of size 300) has four attention heads and the second layer (of size 600) has one attention head. We use 50-dimension position embeddings for both layers and apply dropout with the rate of 0.1 on the input feature vectors. We use Adam optimizer with an initial learning rate of 0.001.

4.6 Results

Here we report the classification and ranking results of the competing methods. We also compare and contrast among the variants of our TEAM framework. Apart from the two versions of the TEAM, namely, TEAM-RG and TEAM-CL, we have task-specific model variants specified as — *attach*–A, *merge*–M and *merge+attach*–MA. Here, (*attach+merge*) means simultaneously optimizing for both the tasks.

4.6.1 Ranking Results

In Table [4.2], we show the performance of the competing methods in terms of (best) ranking scores. We see similar trends for all taxonomies in the sub-tables. When considering only *attach* operation and the test ranking scores, we see TEAM-RG clearly beats Taxo-Expan by a large margin of $(196.87, 487.75, 470.87)$ in MR, by a margin of $(0.2, 0.14, 0.24)$ in Hit@1 and $(0.31, 0.32, 0.37)$ in Hit@3 for Assamese, Bengali and Hindi WordNet respectively. We see TEAM-CL though performing competitively but is outperformed by TEAM-RG by a margin of $(44.82, 86.01, 43.04)$ in MR, by a margin of $(0.11, 0.12, 0.28)$ in Hit@1 and $(0.14, 0.18, 0.43)$ in Hit@3 respectively for Assamese, Ben-

∗https://github.com/dmlc/dgl

Table 4.3: Classification results for test queries

	Assamese WordNet						Bengali WordNet						Hindi WordNet					
Methods	Acc	Mi-F1	Ma-F1	Prec.	Recl	F-Sc	Acc	Mi-F1	Ma-F1	Prec.	Rec.l	F-Sc	Acc.	Mi-F1	Ma-F1	Prec.	Recl	F-Sc
TEAM-RG(A)	0.97	0.97	0.49	0.95	0.97	0.96	0.98	0.98	**0.50**	0.97	0.98	0.97	0.90	0.90	0.47	0.81	0.90	0.85
TEAM-RG(M)	0.81	0.81	0.45	0.66	0.82	0.73	0.29	0.29	0.29	0.57	0.29	0.48	0.55	0.55	0.30	0.23	0.15	0.39
TEAM-RG(MA)	0.88	0.88	0.88	0.89	0.88	0.88	0.51	0.51	0.36	0.96	0.71	0.85	0.53	0.53	0.43	0.82	0.53	0.62
TEAM-CL(MA)	**1.00**	**1.00**	**0.50**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.48	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.50**	**1.00**	**1.00**	**1.00**

gali and Hindi WordNet. In TEAM-RG(M), we obtain near-perfect MRR scores. This is because the definitions are already present in training set for the query concepts with known definitions (test sample drawn from the base taxonomy). The score of 1 indicates the ability of the proposed method TEAM-RG(M) to correctly identify the appropriate anchor nodes for the *merge* operation. TMN gives better performance than Taxo-Expan owing to its useful attention mechanism. But, Team-RG(A) outperforms TMN in all the metrics except Hits@1. We only compare Taxo-Expan results for the *attach* since it is originally proposed for the *attach* operation. In the (*merge*-M) and (*merge+attach*-MA) section of the tables also, we see that TEAM-RG outperforms TEAM-CL on all three WordNet taxonomies. We attribute this huge performance improvement of TEAM-RG to InfoNCE based training — as it simultaneously provides pseudo-supervision from the negative examples while optimizing for the task-specific regression layers.

4.6.2 Classification Results

In Table [4.3], We observe similar trends on all three WordNet taxonomies. Since Taxo-Expan is a regression-based algorithm proposed for only *attach* operation in taxonomy expansion task — we could not obtain its classification performance. Therefore, we only consider variants of our frameworks as competing methods. As described in the sub-section 4.4.4, using a classification layer on top of the regression layer in TEAM-RG, we obtain classification performances for the *attach*, *merge* operations along with both (*attach+merge*) operations. Whereas obtaining classification performance for TEAM-CL is straightforward since this is already a classification framework.

When comparing TEAM-RG (*attach*) and (*merge*) variants — we see, unlike ranking results where ranking results of *merge* operation were always better than the *attach* operation, here the classification results of *merge* operation are inferior to *attach* operation. It means that the RG variant learns better ranking as compared to CL variants, but they fail to distinguish M and A – the operation to perform. This is expected since we

Table 4.4: Ranking result for out-of-vocabulary words

Assamese WordNet-Noun				
Methods	Micro_MR	Hit@1	Hit@3	MRR
TEAM-RG(A)	**65.30**	**0.33**	**0.53**	**0.80**
TEAM-CL(A)	240.47	0.02	0.16	0.06
Taxo_Expan(A)	386.30	0.05	0.05	0.11
TEAM-RG(M)	**170.79**	**0.07**	**0.14**	**0.38**
TEAM-CL(M)	331.93	0.03	0.12	0.29

do not provide a scheme here to contrast M and A operations — which is the motivation for our CL variant framework.

When comparing TEAM-RG and TEAM-CL for (*merge+attach*), we see TEAM-CL gives better classification scores using test queries except for Macro-F1 scores. TEAM-RG gives the best performance for the Macro-F1 score for the test cases. This essentially means that class-wise prediction performances are inferior for TEAM-CL. This is expected behavior since, in each batch of the training sample, we include a substantially large number ($\mathcal{N}$) of negative examples with class-label (N–No operation). We design our training samples like this so that the contrastive loss is better approximated. Nevertheless, it leads to a class-imbalance issue in our three-way classification setup, i.e., a large number of samples with N class labels as compared to the other M/ A class labels. Thus, TEAM-CL biases its prediction towards the N class, leading to poorer Macro-F1 scores than TEAM-RG.

To summarize, we observe that TEAM-RG gives the best ranking performances, whereas TEAM-CL gives the best classification performances. TEAM-CL performs poorly in Macro-F1 since it presumably suffers from class-imbalance issues owing to the style of training sample generation. Frameworks with multi-task learning strategy (TEAM-RG and TEAM-CL) outperform frameworks (Taxo-Expan) designed to perform a single task — which is motivated by the fact that simultaneously optimizing for multiple tasks provides self-supervision to each other, resulting in better performances.

4.6.3 Expansion of Assamese WordNet Taxonomy with Out-Of-Vocabulary (OOV) words

To investigate the effectiveness of the proposed models, we employ the models for expanding a WordNet with OOV words. For this, first, we find out-of-vocabulary words, i.e., words that are not present in Assamese WordNet. Second, we manually identify true

anchors of respective out-of-vocabulary words with associated operations (*attach*/*merge*) in Assamese WordNet. We evaluate the predicted results of the proposed model against the manually identified true anchors. Since we can either perform an A or M operation with OOV words and not both, we do not predict expansion tasks for OOV words using any of the MA variants of our proposed frameworks. Table 4.4 shows the ranking performance of the model in predicting true anchors for *attach* and *merge* expansion operations. We see a similar trend of prediction ranking as seen with the test set in our earlier experiments. TEAM-RG gives the best performance in both expansion operations. In the case of *attach* expansion, TEAM-RG beats state-of-the-art Taxo-expan by a large margin of $(321, 0.31, 0.48, 0.69)$ in MR, Hit@1, Hit@3, MRR, respectively. However, our proposed frameworks are seen not to perform so well in *merge* M operation as compared to the *attach* A operation. Intuitively, this is because, for OOV words, we use a set of manually collected paraphrase definitions of the OOV words to match them with the candidate anchor concepts in the existing taxonomy. Whereas for actually training our model, we have used the same definitions in the replica nodes. That is, we have used the same definition in the original anchor concept and in the input query-concept with mutually exclusive synset information. Thus, in this case-study, the paraphrase-based definition matching deems challenging for our learning model resulting in poorer results for M operation. We believe we can always eliminate this drawback by using a description generation tool[96] to generate different definitions of the same concept nodes and train our learning model in a more powerful way.

4.7 Summary

In this study, we proposed an integrated framework called *Taxonomy Expansion with attach and merge (TEAM)* for expanding taxonomy with *attach* and *merge* operations together. We built two multi-task learning-based variants of TEAM, namely, TEAM-Regression and TEAM-Classification, which solve the taxonomy expansion problem as regression and classification, respectively. Our proposed methods learned to predict the taxonomy expansion operation (*merge*, *attach*, or no-operation) to perform and provided a ranked list of candidates. We evaluated the effectiveness of TEAM on WordNet taxonomies of three distinct languages, viz., Assamese, Bangla, and Hindi. In various experimental

setups, the proposed TEAM-RG and TEAM-CL outperformed its state of the art for *attach* operation and provided a highly encouraging performance on *merge* operation. We had also investigated the performance of the proposed model with out-of-vocabulary concepts.

Though TEAM works efficiently for integrated taxonomy expansion, it focuses on the local taxonomic context. A taxonomy can be of two types: single-root taxonomies (such as English WordNet) and multi-root taxonomies (such as Assamese WordNet). As TEAM considers local context, it faces challenges when it is applied to multi-root taxonomies. In the next chapter, a new approach, named LG-TEAM, is proposed to address these limitations by combining both the local and global context of a taxonomy in an integrated *attach-merge* expansion environment, providing a more robust solution to the problem of taxonomy expansion.

"Learn from the mistakes of others. You can't live long enough to make them all yourself."

Eleanor Roosevelt

5

LG-TEAM: Local and Global context aware multitask learning based Taxonomy Expansion approach for Attach and Merge

Automatic taxonomy expansion is a crucial problem in natural language processing (NLP) because it helps solve the problem of low coverage in a taxonomy. This can improve the accuracy and effectiveness of NLP tasks like information retrieval, information extraction, text classification, summarization, etc. The majority of the prior studies have addressed the expansion problem by performing Attach operation, whereas both Attach and Merge operations are integral parts of the taxonomy expansion task.

In the previous chapter we have proposed, an approach called TEAM to solve this issue. Though TEAM works efficiently for integrated taxonomy expansion, it focuses on the local taxonomic context. A taxonomy can be of two types: single-root taxonomies (such as English WordNet) and multi-root taxonomies (such as Assamese WordNet). As TEAM considers local context, it faces challenges when it is applied to multi-root taxonomies. To address the limitations in TEAM, this chapter proposes a new approach, LG-TEAM, which combines both the local and global context of a taxonomy in an integrated *attach-merge* expansion environment, providing a more robust solution to the problem of taxonomy expansion.

5.1 Introduction

Lexical taxonomy is a system of organizing and categorizing words and concepts based on their meanings. It is employed to understand the relationships between different concepts and clarify the meanings of the words within a particular domain of knowledge. There are many existing taxonomies in different domains, such as WordNet[51], MesH[43] and Pinterest Taxonomy[21]. These taxonomies play an important role in many Natural Language Processing (NLP) tasks, such as information retrieval, information extraction, text classification, summarization,[61,3,80,51] etc. Taxonomies are usually curated manually, which is time-consuming, expensive, and not scalable, resulting in limited coverage. As new concepts are constantly emerging and growing, they may need to be added to the existing taxonomies to keep them up-to-date and useful.

Though automatic taxonomy expansion involves two operations, attach- inserting a new concept and merge- extending existing concepts (see Figure 1.4), most of the existing studies on taxonomy expansion have focused only on the attach operation. To combine both attach and merge operations in a single model, Phukon et al.[64] proposed a multi-task learning-based taxonomy expansion method, TEAM that relied on learning *local* taxonomic information similar to the studies that focused only on the *attach* operation [78,105,96]. These studies focus on utilizing the neighborhood information of a concept-node to learn the taxonomic relatedness between concept-nodes, such as anchor-parent. Despite achieving encouraging results, TEAM has issues when applying to multi-root taxonomies as it only considers *local* taxonomic information. In a multi-

root taxonomy, there are multiple independent sub-trees that are not connected to each other. As a result, the taxonomic information of a concept-node is limited to a specific subtree and may not provide enough context to understand the overall structure of the taxonomy. Because of this, determining the best-fit position for a concept within the taxonomy may require additional information, such as the relationships between concept-nodes in different sub-trees, or external information about the concept itself. In such cases, a more *global* perspective is needed to accurately distinguish between the two concept-nodes, as it would provide a more comprehensive view of their positions in the overall taxonomy.

Motivated by the above observations, we propose LG-TEAM : *Local and Global context aware Multitask learning based Taxonomy Expansion for attach and merge operations*, a method to enhance the effectiveness of automatic taxonomy expansion. Like in TEAM, LG-TEAM considers ego-net (explain in Sec 5.5.3) to capture local context and entire network to capture the global context. To ensure message passing between the sub-trees while capturing global context, one dummy root node is added, connecting to all the sub-tree roots. We evaluate LG-TEAM on four WordNet taxonomies, viz., Assamese, Bengali, Hindi (multi-root taxonomies), and English (single-root taxonomy). From experiments, it is observed that the proposed model outperforms all the baseline counterparts over the entire dataset for *attach* operation. For merge operation, both TEAM and LG-TEAM provide comparable, near-perfect performance. The study also includes ablation analysis to support the need to incorporate global context.

5.2 Related Studies

Existing studies on taxonomy expansion have primarily focused on expanding existing taxonomies through the attachment of new terms or concepts. These studies have employed a variety of techniques, including the use of lexical features such as lexical patterns (e.g. [55,81]), the use of distributional representations from a resource corpus to construct a taxonomy from scratch (e.g. [45,71]), and methods expanding generic taxonomy without using external resources (e.g.[11,47,78,104,48,105,44,46,30,39]). Some of these works have also employed advanced techniques such as graph neural networks (e.g.[78],[104,105,46,30,39,97]) and pre-trained language models (e.g., [96]) to improve the accu-

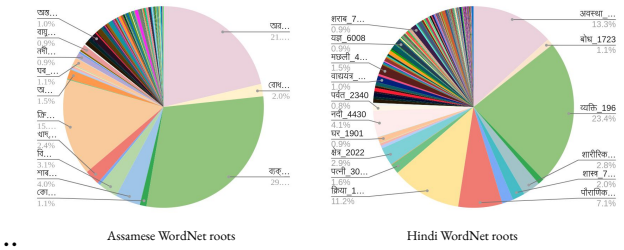


Figure 5.1: Roots in Assamese and Hindi WordNet

racy and efficiency of taxonomy expansion. However, taxonomy expansion also involves the merge operation, which is an important aspect of expanding taxonomies such as Wordnet or making them more precise. Realizing the need to apply both operations, SemEval-2016:task 14 (Semantic taxonomy enrichment) Jurgens and Pilehvar[32] includes a call for expansion with both *attach* and *merge* operations. However, none of the submissions incorporate both operations in a single model.[64] proposed a multi-task learning based taxonomy expansion framework TEAM is jointly trained to perform both the *attach* and merge operations. While TEAM has been shown to be effective for integrated taxonomy expansion, it has limitations when applied to multi-root taxonomies. To address this limitation, we propose a new approach called LG-TEAM, which enhances the effectiveness of automatic taxonomy expansion by combining local and global taxonomic information in a multi-task learning (attach/merge) platform.

5.3 TEAM and multi-root taxonomies

While English Wordnet has only one root taxonomy, Wordnets of Indian languages are found to have multiple roots, i.e., a forest formed by several trees. This is due to the fact that etymologically, Indian languages have roots emerging from different language families. Figure 5.1 shows root nodes and the size of the corresponding trees for Assamese and Hindi WordNet.

In a taxonomic system, the root represents the basic starting point of the entire database. In the context of WordNet taxonomy, the root synset represents the most basic concept of the language, and all other synsets are organized in relation to it. Typically, a traditional taxonomic system comprises of a single root; this is true for English WordNet as well, where the root of the hierarchy is the concept of 'entity', which includes all other concepts in the language. However, in cases where the system

classifies objects or concepts that belong to multiple distinct categories, it is possible for the system to have multiple roots.

With respect to Indian languages, it has been observed that WordNet taxonomies can have multiple roots. This is due to the fact that etymologically Indian languages have roots emerging from different language families. It is important to note that the determination of the roots of a word may not always be straightforward, and in some cases, the origins of words in Indian languages may not be fully understood. As these WordNets are created manually, annotators may be affected by the complexities in tracing the roots of words, and this can lead to the need for creating new roots in the WordNet. Figure 5.1 shows the available root synset in Assamese and Hindi WordNet.

Empirically, it has been observed in the literature that existing studies on taxonomy expansion that consider only local context tend to be more effective in the context of single-root taxonomies than multi-root taxonomies. This discrepancy in performance may be attributed to the lack of consideration of information flow between disconnected components. This concern highlights the need for a global view of a node concept within the taxonomy.

In this concern, we examine the model TEAM [64], as it supports this observation and highlights the challenges in multi-root taxonomy expansion. Though TEAM shows an encouraging performance as an integrated model of attach and merge expansion operations, we have observed a few concerns that limit TEAM from being used in multi-root taxonomy expansion. One of the key concerns is that TEAM uses the summarized representation of the local neighborhood of a node concept to learn the relatedness between concepts. The summarized neighborhood representation of two concepts across different sub-trees may be similar, potentially leading to confusion or inaccuracies in the determination of relatedness between the concepts. This concern highlights the need for a global view of a node concept within the taxonomy. A global view of a sub-tree includes all of the concepts within the sub-tree and how they are related to one another, as well as their relationships with concepts in other sub-trees rooted at different root concepts. Therefore, this study proposes a model that considers both local and global embedding for understanding the similarity and differences between concepts in any taxonomy, as it provides visibility into how concepts relate to one another across

different sub-trees and how they fit into the overall structure of the taxonomy.

5.4 Problem Description

A Taxonomy is a hierarchical organization of concepts. A real-world taxonomy can be either single-root or multi-root in nature. In this study, we have modelled WordNets as taxonomy-trees. A Taxonomy Tree is a directed unweighted acyclic graph represented as $\mathcal{T} = (C, R)$. C represents the set of concepts and R represents the set of hypernymy relations between the concepts. A concept $c \in C$ is further defined by a tuple $c = (def_c, synset_c)$ where def_c represents the *definition* of the concept, and $synset_c$ represents the set of associated synonyms. An edge $r \in R$ represents a hypernymy relation from a parent concept c_{parent} to its child concept c_{child} . The taxonomy $\mathcal{T}$ is arranged in a hierarchical manner with directed edges in R, as shown in Figure 1.4. Given the taxonomy $\mathcal{T}$ and a query concept $q = (def_q, synset_q)$, the *attach* and the *merge* expansion operations are defined below.

Operation Attach and Merge : As defined in TEAM [64], the attach and merge operation can be represented as a function, denoted as $f_{attach}(q, a, \mathcal{T})$, $f_{merge}(q, a, \mathcal{T})$ respectively, where; q is the new concept to be inserted, represented as a tuple $(def_q, synset_q)$; a, stands for anchor, is the best matching parent concept for Attach operation and the best matching concept for Merge operation. It is represented as a tuple $(def_a, synset_a)$;$\mathcal{T}$ is the current taxonomy network, represented as a tuple (C, R)

The function $f_{attach}(q, a, \mathcal{T})$ returns the updated taxonomy network $\mathcal{T}'$, represented as a tuple (C', R'):

$$C' = C \cup q \qquad R' = R \cup (r)$$

$$r : (a \xrightarrow{\text{parent}} q)$$

The function $f_{merge}(q, a, \mathcal{T})$ returns the updated taxonomy network $\mathcal{T}'$, represented as a tuple (C', R'):

$$C' = C \qquad R' = R$$

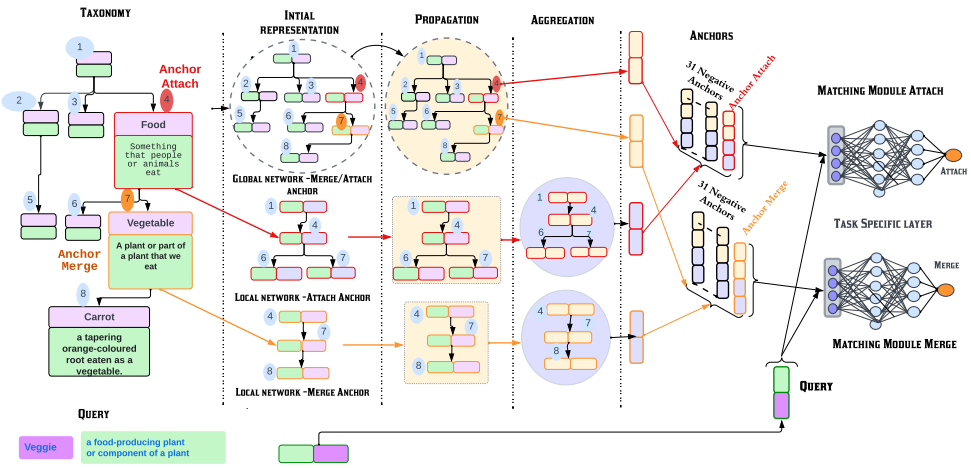


Figure 5.2: Framework of LG-TEAM • **Taxonomy**. query, anchor-merge, anchor-attach, negative anchors in the sample taxonomy • **Initial representation**. initial representation of the sample taxonomy, merge anchors local structure,attach anchors local structure • **Propagation**. graph propagation on the sample taxonomy to generate a global representation of anchors, and propagation on merge and attach anchor's local structure to generate local representations of anchors. • **Aggregation** read out summary of local structure for final representation of anchors • query, anchors, negative anchors are fed to matching module • Projection to shared hidden layers, Task-specific matching modules with non-shareable weights. • Task-specific regression outputs.

$$synset'_a = synset_a \cup synset_q$$

$$a' = (def_a, synset'_a)$$

Where a' is the updated anchor concept, the synset of the anchor concept a has been updated by adding the new synset $synset_q$ to it, and the set of vertices C and edges R remain the same.

5.5 Proposed Method

Our proposed taxonomy expansion framework, LG-TEAM is based on the multi-tasking objective to simultaneously learn from both attach and merge operations. To this end, we enrich the learned node representations with a combination of *local* and *global* neighborhood information. We aim to augment the capabilities of the model by addressing the challenges associated with multi-root taxonomies through the incorporation of enhanced node representations. The illustration of the LG-TEAM model is shown in Figure 5.2. We choose TEAM-Regression (RG) from the original paper[64] and show how TEAM-RG can be improvised to solve the multi-root as well as single root taxonomy expansion problem.

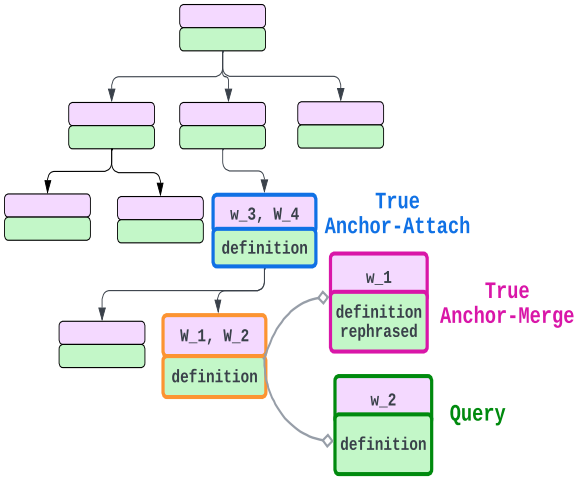


Figure 5.3: Generation of training data from the taxonomy :In this sample taxonomy orange node is the query. Final query and true anchor *merge* is created from orange node , blue node is true *anchor attach*, any random black node is the negative anchor for *attach* and *merge*

5.5.1 Generation of training data

The training dataset is generated from the transformed WordNet taxonomy $\mathcal{T}$, where each sample is a 3-tuple $[q, a,$ label$]$, q is the query, a is the potential anchor, and label is the associated class, i.e., True/False (1/0). A set of nodes in $\mathcal{T}$ are randomly selected as queries. The samples for the *attach* and *merge* operations are generated separately. For attach, the parent of the query node is considered as positive anchor, and other randomly selected nodes are considered the negative anchor. With these positive and negative anchors, positive and negative samples are generated. For merge, a positive sample is generated by removing a randomly selected synonym from the query node, and the modified query node is considered as the positive anchor and a positive sample is created. Negative samples are generated by randomly selecting other nodes. Figure 5.3 illustrates the generation of training samples from the taxonomy. Our proposed framework consists of the following subsections.

5.5.2 Representing the query concept

In the approach, a query concept is represented by a tuple $q = (def_q, synst_q)$, where def_q is a text description of the concept and $synset_q$ is a set of synonyms that closely relate to the concept. We utilize fasttext embeddings to represent the candidate words. The

embedding vectors of all candidate words are then combined to form a single embedding vector $\bar{q}$ that represents the query q.

5.5.3 Representing the anchor concept

An anchor $a = (def_a, synset_a)$ in a taxonomy tree $\mathcal{T} : (C_a, R_a)$ is a prospective node with which an incoming query node q should be attached or merged. Apart from the definition and synset associated with the anchor a, we exploit the neighborhood structure, aka ego-tree surrounding a to represent the node a. Past works[96,104,105,78] justified this contextualized representation of the anchor nodes to help the learning algorithm infer better-matching query-anchor pairs.

Here, we used neighborhood contexts (ego-trees) at multiple granularities to define a local and a global neighborhood for an anchor. We use a suitable Graph Neural Network (GNN) that uses – 1) an attention mechanism to weigh the neighboring nodes according to their roles and, 2) a position-encoding strategy to incorporate the relative positions (i.e., parent, grand-parent, children, siblings) of the candidate nodes in the ego-tree to better discriminate their roles in enriching the anchor node a. We adopt position-enhanced Gated Attention Network (GAT) from Taxo-Expan[78] for embedding the local and global ego-trees of an anchor a. Once the node embeddings $\bar{x}$ for each participant node $x \in C_a$ are generated, an activation function is employed on them to obtain a collective ego-tree representation $\bar{\mathcal{T}}_a$ as below.

$$\bar{\mathcal{T}}_a = \sigma\Big(\frac{1}{|C_a|}\sum_{x \in C_a} \bar{x}\Big) \tag{5.1}$$

where $\sigma(.)$ is the *Sigmoid* activation function.

5.5.4 Generating the local and global views

Local view A precise $k-$hop ego-tree is extracted from the taxonomy tree for a prospective merge/ attach anchor node. We consider $k = 1$ and extract direct relations, i.e., parent, sibling and child nodes surrounding the anchor. A suitable GNN is employed to obtain a summarized local ego-tree representation $\bar{\mathcal{T}}_{l_a}$ that only considers the roles of an anchor's direct relations to infer its semantic meaning. Fasttext embeddings are

used to initialize each participating node's features based on its associated definition and synset.

Global view Unlike TEAM-RG, which only considers the microscopic view of a node's ego-tree for automatic taxonomy expansion, we enrich the microscopic view with the node's global neighborhood information for better discrimination.

To obtain a global view, a multi-layer ($L = 5$) attentive GNN module is implemented on the taxonomy tree to extract a larger neighborhood-based coarser-level context for each anchor node. The final representation of the anchor a gives a global view that incorporates the roles of nodes beyond that anchor's direct relations. It gives a high-level contextualized anchor node representation $\bar{\mathcal{T}}_{g_a}$ for better discrimination of closely related concepts. We experimented with a number of layers L and chose a suitable value that gave us the best performance.

Combining the local and global views Finally, we combine the local and global views for each anchor node to get a unified anchor node representation based on its ego-tree structure. Attach $\bar{\mathcal{T}}_{a_A}$ and Merge $\bar{\mathcal{T}}_{a_M}$ anchor node representations are obtained via concatenation operation as,

$$\bar{\mathcal{T}}_{a_A} = w_1 * \bar{\mathcal{T}}_{l_{a_A}} \oplus w_2 * \bar{\mathcal{T}}_{g_{a_A}} \tag{5.2}$$

$$\bar{\mathcal{T}}_{a_M} = w_1 * \bar{\mathcal{T}}_{l_{a_M}} \oplus w_2 * \bar{\mathcal{T}}_{g_{a_M}} \tag{5.3}$$

Here, A, M denote types of operations – attach and merge, respectively. Subscript a denotes anchor. l, g denote types of views – local and global. $w_1, w_2 \in \mathbb{R}$ are real-valued weights to measure the contributions of two views, and $\oplus$ refers to the concatenation operation. In the Ablation study, we show that this simple weighted combination of two different views helps improve the ranking performance.

5.5.5 Matching Module

The anchor embedding obtained from Eqn 5.2 and Eqn 3 are passed through a shared dense layer, which facilitates information sharing between the two end-tasks. It is followed by task-specific matching modules for attach and merge operations. Similar to

TEAM-RG, we use a bi-linear matching module that captures the latent interaction of anchor $\bar{a}$ and query $\bar{q}$ embeddings in the projection space via a bi-linear scoring matrix $B \in \mathbb{R}^{|\bar{q}| \times |\bar{a}|}$. A real-valued score $\mathbb{R} \in \{0,1\}$ is learned as a probability estimate of the anchor-query association as,

$$\mathcal{D}(q, a_A) = \sigma(\bar{q}^T B \bar{a_A}) \tag{5.4}$$

$$\mathcal{D}(q, a_M) = \sigma(\bar{q}^T B \bar{a_M}) \tag{5.5}$$

Here σ is the Sigmoid non-linearity. Subscript A, M denote types of operations – attach and merge.

5.5.6 Efficient representation of a multi-root taxonomy via augmenting a dummy root

GNNs are capable of learning global features in multiple-rooted taxonomies. However, GNNs may face difficulties in discriminating similar concepts in disconnected subtrees due to the absence of message-passing across the differently rooted subtrees and the need for more contextual information from each subtree to discriminate among a set of similar concepts

In this study, we introduce a dummy root node that provides a generic abstraction to connect all the root nodes in the taxonomy via hypernymy relations. This dummy root connects all the disconnected trees. Applying GNN to this connected taxonomy tree facilitates the following : 1) It helps in information dissemination via a message-passing mechanism across the differently rooted trees, 2) Closely related concepts can be discriminated based on better-contextualized representations, which helps the learning algorithm decide the most suitable concept for matching with a query.

In the Experiments section, we show taxonomy expansion results for taxonomy trees with and without a dummy root and analyze the outcomes to draw important insights.

5.5.7 Optimization

We sample $\mathcal{N}$ number of negative examples for each positive merge/ attach anchor $a_{A/M}$ given a query q. Our task-specific training data are denoted as $\mathcal{Z}_A : \{(q, a_A, 1), (q, a'_{A_1}, 0),$

Table 5.1: Dataset Statistics

	Nodes	Edges	Roots	Leaf
Assamese WordNet	8466	8363	103	7072
Bengali WordNet	26007	25815	226	22847
Hindi Wordnet	28242	28016	192	24737
English Wordnet	74401	75850	1	57708

$(q, a'_{A_2}, 0), ..., (q, a'_{A_\mathcal{N}}, 0)\}$ and $\mathcal{Z}_M : \{(q, a_M, 1), (q, a'_{M_1}, 0), (q, a'_{M_2}, 0), ..., (q, a'_{M_\mathcal{N}}, 0)\}$ for each query $q \in Q$. Here true and false associations are labeled with 1 and 0 respectively. An InfoNCE loss is estimated over the query set Q as an average probability estimate of the occurrence of a true anchor given a set of true-false attach anchors, $\mathcal{M}(q_A)$ and merge anchors $\mathcal{M}(q_M)$,

$$\begin{aligned} \mathcal{L} = & - \frac{1}{|Q|} \sum_{q \in Q} log \frac{\mathcal{D}(\bar{q}, \bar{a_A})}{\sum_{v \in \mathcal{M}(q_A)} \mathcal{D}(\bar{q}, \bar{v})} \\ & - \frac{1}{|Q|} \sum_{q \in Q} log \frac{\mathcal{D}(\bar{q}, \bar{a_M})}{\sum_{v \in \mathcal{M}(q_M)} \mathcal{D}(\bar{q}, \bar{v})} \end{aligned} \tag{5.6}$$

5.5.8 Inference

At the time of inference, when a query-anchor is given as input, a classification strategy is implemented based on the task-specific matching module scores. The merge and attach real-valued matching scores are compared. The input anchor is categorized as Merge or Attach anchor, i.e., the operation is classified as Merge or Attach operation based on the task whose matching module has given the best score. Given a query q and a list of corrupted/ true candidate anchors $a_{A/M}$, a ranked list of anchors is obtained by optimizing the loss function in Eqn 5.6. The best-ranked anchor becomes the best matching candidate for a query, and the corresponding operation is classified using the above-mentioned strategy.

Table 5.2: Overall experimental results in multi-root settings

Method	Assamese WordNet-Noun(multi-root)									
	Ranking				Classification					
	Macro_mr	Hit_at_1	Hit_at_3	MRR_scaled_10	Accuracy	Micro-F1	Macro-F1	Precision	Recall	F-Score
Taxonexpan	341.81	0.07	0.11	0.29	-	-	-	-	-	-
TMN	203.28	0.28	0.41	0.71	-	-	-	-	-	-
TEAM-Attach	144.92	0.27	0.42	0.67	0.97	0.97	0.49	0.95	0.97	0.96
LG-TEAM-Attach	**110.72**	**0.31**	**0.45**	**0.75**	**0.99**	**0.99**	**0.50**	**0.97**	**0.99**	**0.98**
TEAM-Merge	1.27	0.95	0.98	1.00	0.81	0.81	0.45	0.66	0.82	0.73
LG-TEAM-Merge	1.51	0.91	0.96	0.99	0.81	0.81	0.43	0.62	0.82	0.71
TEAM-Merge/A	73.34	0.61	0.70	0.83	0.88	0.88	0.47	0.77	0.88	0.82
LG-TEAM-Merge/Attach	**56.26**	0.60	**0.71**	**0.84**	**0.90**	**0.90**	**0.48**	**0.79**	**0.90**	**0.83**
Method	**Hindi WordNet-Noun(multi-root)**									
	Ranking				Classification					
	Macro_mr	Hit_at_1	Hit_at_3	MRR_scaled_10	Accuracy	Micro-F1	Macro-F1	Precision	Recall	F-Score
Taxonexpan	648.72	0.04	0.08	0.14	-	-	-	-	-	-
TMN	246.45	0.31	0.25	0.61	-	-	-	-	-	-
TEAM- Attach	177.85	0.28	0.43	0.67	0.90	0.90	0.47	0.81	0.90	0.85
LG-TEAM-Attach	**142.45**	**0.37**	**0.51**	**0.72**	**0.94**	**0.94**	**0.48**	**0.88**	**0.93**	**0.91**
TEAM-Merge	5.38	0.83	0.88	0.95	0.55	0.55	0.30	0.23	0.15	0.39
LG-TEAM-Merge	**1.64**	**0.90**	**0.96**	**0.99**	**0.82**	**0.82**	**0.45**	**0.68**	**0.82**	**0.74**
TEAM-Merge/Attach	91.62	0.63	0.71	0.81	0.53	0.53	0.43	0.82	0.53	0.62
LG-TEAM-Merge/Attach	**72.05**	**0.52**	**0.64**	**0.84**	**0.88**	**0.88**	**0.88**	**0.89**	**0.88**	**0.88**
Method	**Bengali WordNet-Noun**(multi-root)									
	Ranking				Classification					
	Macro_mr	Hit_at_1	Hit_at_3	MRR_scaled_10	Accuracy	Micro-F1	Macro-F1	Precision	Recall	F-Score
Taxonexpan	679.26	0.03	0.04	0.10	-	-	-	-	-	-
TMN	319.36	0.10	0.15	0.69	-	-	-	-	-	-
TEAM- Attach	191.51	0.17	0.36	0.86	0.98	0.98	0.50	0.97	0.98	0.97
LG-TEAM-Attach	**174.11**	**0.17**	**0.41**	**0.71**	**0.99**	**0.99**	**0.52**	**0.97**	**0.99**	**0.97**
TEAM-Merge	2.04	0.92	0.98	1.00	0.29	0.29	0.29	0.57	0.29	0.48
LG-TEAM-Merge	**1.88**	0.92	0.97	0.99	**0.74**	**0.74**	**0.43**	**0.56**	**0.64**	**0.74**
TEAM-Merge/Attach	59.81	0.69	0.80	0.85	0.51	0.51	0.36	0.96	0.71	0.85
LG-TEAM-Merge/Attach	60.57	0.53	0.617	0.80	**0.88**	**0.88**	**0.47**	0.77	**0.88**	**0.82**
Method	**English WordNet-Noun**(single-root)									
	Ranking				Classification					
	Macro_mr	Hit_at_1	Hit_at_3	MRR_scaled_10	Accuracy	Micro-F1	Macro-F1	Precision	Recall	F-Score
Taxonexpan	1454.39	0.01	0.07	0.12	-	-	-	-	-	-
TMN	982.28	0.03	0.07	0.16	-	-	-	-	-	-
TEAM- Attach	184.19	0.05	0.22	0.56	0.82	0.82	0.45	0.68	0.82	0.74
LG-TEAM-Attach	**155.64**	0.05	**0.25**	**0.60**	**0.95**	**0.95**	**0.49**	**0.91**	**0.95**	**0.94**
TEAM-Merge	2.66	0.90	0.95	0.99	0.76	0.76	0.43	0.58	0.76	0.66
LG-TEAM-Merge	3.60	0.85	0.94	0.99	0.23	0.23	0.18	0.05	0.22	0.08
TEAM-Merge/Attach	93.34	0.48	0.58	0.77	0.59	0.59	0.44	0.81	0.73	0.84
LG-TEAM-Merge/Attach	**83.95**	0.44	0.58	0.80	**0.83**	**0.83**	**0.52**	**0.86**	**0.84**	**0.89**

5.6 Experiments

5.6.1 Experimental setup

Datasets. For our experiments, we use four WordNet taxonomies. Table 5.1 shows the basic statistics of these WordNet taxonomies. In these WordNet taxonomies 3 taxonomies are multi-root in nature i.e Assamese, Bengali, and Hindi WordNet and one taxonomy is single-root i.e English WordNet.

Evaluation Metrics. The performance of the model and its baselines is evaluated using Mean Rank (MR), Hit@k, and Mean Reciprocal Rank (MRR) with the MRR score scaled by a factor of 10 to emphasize performance variations. Additionally, the ability of the model to predict the appropriate operation (*merge*, *attach*) is evaluated using

Table 5.3: Overall experimental results in dummy-root settings

	Assamese WordNet-Noun				Bengali WordNet-Noun				Hindi WordNet-Noun			
Methods	Micro_MR	Hit@1	Hit@3	MRR	Micro_MR	Hit@1	Hit@3	MRR	Micro_MR	Hit@1	Hit@3	MRR
LG-TEAM-Attach	105.5	0.19	0.39	0.68	155.49	0.28	0.48	0.73	125.40	0.19	0.39	0.73
LG-TEAM-Merge	1.60	0.90	0.95	0.99	2.43	0.8	0.96	0.98	1.85	0.89	0.96	0.99
LG-TEAM-Merge/Attach	53.56	0.54	0.67	0.83	78.96	0.58	0.72	85.73	63.62	0.54	0.67	0.85

Accuracy, Micro/Macro F1, Precision, Recall, and F-Scores.

Baseline Methods. In this study, a new model is proposed to address the limitations of the TEAM model. To evaluate the effectiveness of the new model, it is compared to the **TEAM** model and its variants. Additionally, as the new model can provide both Attach and Merge ranking results independently, we compare its results for the Attach operation with two state-of-the-art models that only perform the Attach operation. However, as there is no existing model that directly uses the Merge operation to expand a taxonomy, the results for the Merge operation are compared with only the TEAM merge variants. We choose two most recent benchmark SOTA taxonomy-expansion frameworks **TaxoExpan**[78] and **Triplet Matching Network(TMN)**[105] as the competing methods. In terms of learning objective, Taxo-Expan is similar to ours. It uses ego-tree-based anchor features for matching query features in a regression-based setting. TMN captures the fine-grained relationship dynamics of query and anchor concepts using channel-wise gating mechanisms based on attention learning.

5.6.2 Experimental Results

The two sets of experimental results are shown in Table 5.2 and Table 5.3. Table 5.2 reports ranking and classification results of the test queries that were experimented with in the ground-truth taxonomies. Table 5.3 reports the ranking result of test queries in a dummy root settings. We are showing three variants of the the model LG-TEAM-Attach(attach independently), LG-TEAM-Merge(merge independently) and LG-TEAM-Attach/Merge(as a multi-task)

Performance in multi-root settings: Table 5.2, presents a comprehensive evaluation of the performance of our proposed model and the baseline models on a common set of test queries for all four taxonomies. The performance metrics reported in the table include both classification and ranking scores. When considering the performance of the independent attach operation, we see a consistent pattern in performance for all four

taxonomies. Our proposed model, LG-TEAM-Attach, demonstrates superior ranking performance compared to all baseline models, with an average margin of improvement of (28.5) in Micro_MR, (0.03) in Hit@1, and (0.05) in MRR across all four taxonomies. Similarly, we see improvement in classification performance by a margin of (0.05) in accuracy, (0.09) in precision, (0.07) in F-score. A key observation among these models is that models trained in a multitasking environment, i.e., TEAM and LG-TEAM, outperform models that are trained on a single-task, i.e., Taxo-expan and TMN. TMN gives better performance than Taxo-Expan owing to its useful attention mechanism. Between TEAM and the proposed LG-TEAM, LG-TEAM outperforms TEAM in all metrics of ranking and classification in individual attach operations. We see the same improved result for the single-root English Wordnet taxonomy. This improvement can be attributed to the use of a multitasking environment and the incorporation of local-global context-aware features. When evaluating the performance of our proposed model and the baseline models in the independent merge operation, i.e., LG-TEAM-Merge, we see that both TEAM and the proposed model performed exceptionally well, with near-perfect (near to 1) mean rank scores for all taxonomies. However, the results are competitive between TEAM-Merge and LG-TEAM-Merge. It shows improved merge performance only in Hindi and Bengali WordNet, with an improvement of (1.95) in Micro_MR, (0.03) in Hit@1, and (0.02) in MRR on average. However, the performance variance of LG-TEAM is smaller than TEAM. For example, in Hindi WordNet LG-TEAM-Merge has improved by (3.74) in mean rank. When comparing the performance of Merge/Attach, the proposed model demonstrated competitive results across all taxonomies, with the exception of the Bengali WordNet taxonomy. The overall observations on the Table 5.2 are 1) The local and global context aware features are effective for finding a position for a concept in a taxonomy. 2) Merge operation shows a near perfect result in both TEAM and LG-TEAM. However performance variation in LG-TEAM is lower than TEAM.

Performance in dummy root settings: Table 5.3 shows the performance of the model in dummy root settings (discussed in section 5.5.6). As we see in Table 5.2 when ranking results are improving, classification is also improving; therefore, for evaluation, we have considered only ranking metrics for this dummy-root settings. Moreover, as English Wordnet is itself a single-root taxonomy, we have not consider this data-set for

Table 5.4: Contribution of local and global context in anchor ranking on Assamese WordNet taxonomy

Local weights	0.0	1.0	0.5	0.2	0.8	0.1	0.9	0.3	0.7
Global weights	1.0	0.0	0.5	0.8	0.2	0.9	0.1	0.7	0.3
ranking(Mean rank)	144	721	100	123	139	133	122	116	132

evaluation in dummy-root settings.

While comparing Table 5.2 and Table 5.3, we see that introducing dummy-root to unify all the disconnected taxonomy trees has improved the attach task ranking performance by a significant margin in all multi-root taxonomies. We also see significant ranking performance improvement in the multi-task settings that consider both merge and attach operations.

5.7 Ablation study of local and global contexts

In this interesting study, we segregate and analyze the contributions of local and global contexts to improve the end-task performance. It is interesting to see that, solely local or global context does no good to improve the anchor ranking performance given a test query. Whereas, a combination of both the contexts has significant influence on the ranking performance.

5.8 Conclusion

In this study, we propose a model, LG-TEAM, for automatic taxonomy expansion that incorporates global context with local context to capture information in multi-root taxonomies. Additionally, to ensure efficient message passing between different sub-trees in multi-root taxonomies, we also experiment LG-TEAM with a dummy root augmented taxonomies. Through extensive experiments on various taxonomies, we evaluate the effectiveness of LG-TEAM for expanding multi-root taxonomies. As a future direction, the research can be extended by considering other relations present in a taxonomy to capture additional semantic information that may aid in taxonomy expansion, thus further improving the accuracy and effectiveness of the proposed approach.

6

Conclusion and Future work

6.1 Conclusion

This thesis explores the automatic taxonomy expansion problem by addressing the inherent challenges in WordNet, with a specific focus on IndoWordNet, a multilingual WordNet for Indian languages. Through this study, three significant contributions are presented, which address the identified gaps in existing research. Outlined below is a summary of these contributions.

1. This thesis first performs an empirical study to evaluate the effectiveness of link prediction methods in identifying missing synonymy relations in WordNet with a special emphasis on the Assamese WordNet from IndoWordNet. The Assamese WordNet, created through the *expansion* method using the Hindi WordNet, also experiences missing synonymy relations. As WordNets can be visualized as a

network of unique words connected by synonymy relations, link prediction in complex network analysis is a well-explored research field for predicting missing relations in a network. Hence this study evaluates the effectiveness of using state-of-the-art link prediction methods for automatically predicting missing synonymy relations in Assamese WordNet. From various experiments, it is observed that for discovering missing relations in the Assamese WordNet, simple local proximity-based methods are more effective than global and complex supervised models using network embedding. See Chapter 3

2. Further, this thesis proposes a multitask learning-based automatic taxonomy expansion approach that can perform both *merge* and *attach* operations in a single model. Most taxonomy expansion approaches are of two types, *attach* and *merge*. In a taxonomy like WordNet, both *merge* and *attach* operations are integral parts of the expansion operations, but the majority of studies consider them separately. This study proposes a novel multi-task learning-based deep learning method known as *Taxonomy Expansion with Attach and Merge (TEAM)* that performs both the *merge* and *attach* operations. To the best of our knowledge, this is the first study that integrates both operations in a single model. The proposed models have been evaluated on three WordNet taxonomies: Assamese, Bangla, and Hindi. The various experimental setups show that TEAM outperforms its state-of-the-art counterparts for *attach* operation and provides highly encouraging performance for the *merge* operation. See Chapter 4

3. Finally, this thesis proposes a local-global context-aware taxonomy expansion approach that integrates *merge* and *attach* operations to provide a more comprehensive solution to the taxonomy expansion problem. Though the second contribution, TEAM works efficiently for integrated taxonomy expansion; it focuses on the local taxonomic context. A taxonomy can be of two types: single-root taxonomies (such as English WordNet) and multi-root taxonomies (such as Assamese WordNet). As TEAM considers local context, it faces challenges when applied to multi-root taxonomies. To address the limitations in TEAM, this thesis proposes another approach, LG-TEAM, which combines both the *local* and *global* context of

taxonomy in an integrated *attach-merge* expansion environment, providing a more robust solution to the problem of taxonomy expansion. Extensive experiments on English, Assamese, Bengali, and Hindi WordNets demonstrate the effectiveness and efficiency of LG-TEAM for automatic taxonomy expansion. See Chapter 5

Through the presented contributions, this thesis aims to address the challenges of automatic WordNet taxonomy expansion, thereby improving the ability of WordNet to provide current and precise information about the relationships between words and concepts. These methods are not limited to the WordNet application domain and can be utilized in any kind of taxonomy expansion. For example, they can be applied to taxonomies used by online retailers such as eBay and Amazon, as well as taxonomies used in search engines. By utilizing these methods in other applications, we can improve the accuracy and effectiveness of these taxonomies, ultimately enhancing the user experience.

6.2 Towards more comprehensive and accurate Indian WordNets: A concluding discussion

The contributions presented in this thesis highlight the importance of improving WordNet taxonomies and, specifically, the need to enhance Indian WordNets such as IndoWordNet. While the proposed approaches have demonstrated significant improvements in the accuracy of WordNet taxonomies, there are still several areas where these Indian WordNets can be further improved.

One approach could be to expand the coverage of these WordNets by including more words and senses, particularly for under-resourced languages. This can be achieved through manual curation or automatic methods such as those presented in the thesis mentioned earlier. Additionally, there could be efforts to improve the accuracy and consistency of the WordNet content by implementing a standard set of guidelines for creating and maintaining synsets. There could also be efforts to incorporate more cross-lingual relations to facilitate better interlinking between Indian WordNets and WordNets in other languages. Improving the lexical representation of the Indian WordNets is also an area of concern. This could involve using more sophisticated natural language

processing techniques to disambiguate homographs and polysemous words, as well as incorporating more diverse sources of lexical knowledge such as corpus-based methods and distributional semantics.

An automatic ontology expansion approach could be developed to improve the semantic ontology of WordNets, which could help capture the nuances of meaning across different languages and cultures. This approach could involve using machine learning techniques to automatically extract and map new ontological categories to real-world entities and events, based on the analysis of large amounts of text data in multiple languages.

In summary, improving the Indian WordNets is a multi-faceted problem that requires addressing various challenges related to content data, synsets, lexical representation, lexical network, sense relations, and semantic ontology. While there are existing approaches that have made significant contributions to this field, there is still much work to be done to achieve more comprehensive and accurate WordNets that better reflect the linguistic and cultural diversity of India.

6.3 Limitations and challanges

The thesis has several limitations that need to be acknowledged. Firstly, the contributions of this thesis are experimented with a limited number of datasets. While the proposed techniques show promising results, they need to be evaluated on a larger and more diverse set of datasets to ensure their generalizability. Secondly, in the first contribution, sense disambiguation is not considered, which may lead to some predicted new synonyms not representing the same sense accurately. This limitation may affect the accuracy of the proposed techniques in some cases. Thirdly, the proposed model for taxonomy construction only considers one relation, while taxonomies may have more than one relation. This limitation may affect the quality of the constructed taxonomies and their suitability for certain tasks.

The research work faced several major challenges throughout its course. One significant challenge was the creation of a suitable ground truth dataset for the study. Another notable challenge was finding appropriate embedding techniques for the under-researched languages, which required extensive experimentation and testing.

6.4 Future work

This study has several potential future directions. One direction may be evaluating the performance of the proposed model in a broader range of taxonomies and WordNets for various languages. This extension of the study will provide crucial information regarding the scalability and robustness of the proposed approach, as well as its potential for application to different language resources. Moreover, it is also possible to explore the use of more advanced contextual encoders in combination with the proposed model. This could improve the representation of context-based relationships between words and concepts, thus enhancing the accuracy and comprehensiveness of the expanded taxonomies. This line of research holds great promise in advancing the state-of-the-art in taxonomy expansion and has the potential to provide valuable resources for NLP tasks. Furthermore, The focus of this thesis is limited to the identification of missing synonymy relations. However, a potential future study could encompass the detection of other missing relations, such as meronymy, antonymy, and hypernymy. This area of research holds significant potential in advancing the field of automatic taxonomy expansion and providing valuable resources for NLP tasks.

References

[1] Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. Social networks, 25(3):211–230, Elsevier.

[2] Adamic, L. A. and Huberman, B. A. (2000). Power-law distribution of the world wide web. science, 287(5461):2115–2115, American Association for the Advancement of Science.

[3] Allan, J., Papka, R., and Lavrenko, V. (1998). On-line new event detection and tracking. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 37–45, August(24-28), Melbourne, Australia. ACM.

[4] Barabási, A.-L. (2009). Scale-free networks: a decade and beyond. science, 325(5939):412–413, American Association for the Advancement of Science.

[5] Baxter, J. (1997). A bayesian/information theoretic model of learning to learn via multiple task sampling. Machine learning, 28:7–39, Springer.

[6] Bharali, H., Mahanta, M., Sarma, S. K., Saikia, U., and Sarmah, D. (2014). An analytical study of synonymy in assamese language using worldnet: Classification and structure. In Proceedings of the Seventh Global Wordnet Conference, pages 250–255, January (25-29), Tartu, Estonia.

[7] Bhattacharyya, P. (2010). Indowordnet. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), May(19-21), Valletta, Malta.

[8] Bhattacharyya, P. (2017). Indowordnet. In The WordNet in Indian Languages, pages 1–18. Springer.

[9] Blondel, V. D., Gajardo, A., Heymans, M., Senellart, P., and Van Dooren, P. (2004). A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM review, 46(4):647–666, SIAM.

[10] Blondel, V. D. and Senellart, P. P. (2002). Automatic extraction of synonyms in a dictionary. vertex, 1:x1, Pierre Senellart.

[11] Boteanu, A., Kiezun, A., and Artzi, S. (2019). Synonym expansion for large shopping taxonomies. In 1st Conference on Automated Knowledge Base Construction, AKBC, May (20-22), Amherst, MA, USA.

[12] Chandramouli, C. and General, R. (2011). Census of india. Rural Urban Distribution of Population, Provisional Population Total. New Delhi: Office of the Registrar General and Census Commissioner, India.

[13] Dash, N. S., Bhattacharyya, P., and Pawar, J. D. (2017). The WordNet in Indian Languages. Springer.

[14] DiMarco, C., Hirst, G., and Stede, M. (1993). The semantic and stylistic differentiation of synonyms and near-synonyms. In AAAI Spring Symposium on Building Lexicons for Machine Translation, pages 114–121, March (23-25), Washington, DC, USA.

[15] Duong, L., Cohn, T., Bird, S., and Cook, P. (2015). Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers), pages 845–850, July (26-31), Beijing, China.

[16] Eagle, N., Pentland, A. S., and Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. Proceedings of the national academy of sciences, 106(36):15274–15278, National Acad Sciences.

[17] Edmonds, P. and Hirst, G. (2002). Near-synonymy and lexical choice. Computational linguistics, 28(2):105–144, MIT Press.

[18] Edmonds, P. G. (2000). Semantic representations of near-synonyms for automatic lexical choice. University of Toronto.

[19] Fei, H., Tan, S., and Li, P. (2019). Hierarchical multi-task word embedding learning for synonym prediction. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 834–842, August (4–8), Anchorage, AK, USA. ACM.

[20] Glänzel, W. and Schubert, A. (2004). Analysing scientific networks through co-authorship. In Handbook of quantitative science and technology research, pages 257–276. Springer.

[21] Gonçalves, R. S., Horridge, M., Li, R., Liu, Y., Musen, M. A., Nyulas, C. I., Obamos, E., Shrouty, D., and Temple, D. (2019). Use of owl and semantic web technologies at pinterest. In International Semantic Web Conference, pages 418–435, October 26–30, Auckland, New Zealand. Springer.

[22] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pages 855–864, August(13-17), San Francisco, CA, USA.

[23] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 297–304. May (13-15), Chia Laguna Resort, Sardinia, Italy.

[24] Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology, 143(1):29–36, Radiological Society of North America.

[25] Hauer, B. and Kondrak, G. (2020). Synonymy= translational equivalence. arXiv preprint arXiv:2004.13886.

[26] He, Y., Chakrabarti, K., Cheng, T., and Tylenda, T. (2016). Automatic discovery of attribute synonyms using query logs and table corpora. In Proceedings of the 25th International Conference on World Wide Web, pages 1429–1439. April (11-15), Montreal, Canada. ACM.

[27] Jannink, J. and Wiederhold, G. (1999). Thesaurus entry extraction from an online dictionary. In Proceedings of Fusion, volume 99. Citeseer.

[28] Jeh, G. and Widom, J. (2002). Simrank: a measure of structural-context similarity. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 538–543, July(23–26), Edmonton, Alberta, Canada. ACM.

[29] Jha, S., Narayan, D., Pande, P., Bhattacharyya, P., et al. (2001). A wordnet for hindi. In International Workshop on Lexical Resources in Natural Language Processing, Hyderabad, India.

[30] Jiang, M., Song, X., Zhang, J., and Han, J. (2022). Taxoenrich: Self-supervised taxonomy completion via structure-semantic representations. In Proceedings of the ACM Web Conference 2022, pages 925–934, April (25-29), Virtual Event, Lyon, France. ACM.

[31] Jurgens, D. and Pilehvar, M. T. (2015). Reserating the awesometastic: An automatic extension of the wordnet taxonomy for novel terms. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1459–1465, May 31 – June 5, Denver, Colorado, USA. The Association for Computational Linguistics.

[32] Jurgens, D. and Pilehvar, M. T. (2016). Semeval-2016 task 14: Semantic taxonomy enrichment. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pages 1092–1102, June (16-17), San Diego, CA, USA. The Association for Computational Linguistics.

[33] Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., and Tsuda, K. (2009). Link propagation: A fast semi-supervised learning algorithm for link prediction. In Proceedings of the 2009 SIAM international conference on data mining, pages 1100–1111. April 30 - May 2, Sparks, Nevada, USA, SIAM.

[34] Kipf, T. N. and Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[35] Kipf, T. N. and Welling, M. (2016b). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.

[36] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 46(5):604–632, ACM New York, NY, USA.

[37] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, IEEE.

[38] Kovács, I. A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.-K., Kishore, N., Hao, T., et al. (2019). Network-based prediction of protein interactions. Nature communications, 10(1):1240, Nature Publishing Group.

[39] Lee, D., Shen, J., Kang, S., Yoon, S., Han, J., and Yu, H. (2022). Taxocom: Topic taxonomy completion with hierarchical discovery of novel topic clusters. In Proceedings of the ACM Web Conference 2022, pages 2819–2829, April (25 - 29), Virtual Event, Lyon, France. ACM.

[40] Leeuwenberg, A., Vela, M., Dehdari, J., and van Genabith, J. (2016). A minimally supervised approach for synonym extraction with word embeddings. The Prague Bulletin of Mathematical Linguistics, 105(1):111–142, Sciendo.

[41] Lei, C. and Ruan, J. (2012). A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. Bioinformatics, 29(3):355–364, Oxford University Press.

[42] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7):1019–1031,Wiley Online Library.

[43] Lipscomb, C. E. (2000). Medical subject headings (mesh). Bulletin of the Medical Library Association, 88(3):265, Medical Library Association.

[44] Liu, Z., Xu, H., Wen, Y., Jiang, N., Wu, H., and Yuan, X. (2021). Temp: Taxonomy expansion with dynamic margin loss through taxonomy-paths. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3854–3863, Virtual Event / Punta Cana, Dominican Republic, November(7-11), Virtual Event / Punta Cana, Dominican Republic. Association for Computational Linguistics.

[45] Luu, A. T., Tay, Y., Hui, S. C., and Ng, S. K. (2016). Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 403–413, November (1–4), Austin, Texas, USA. The Association for Computational Linguistics.

[46] Ma, M. D., Chen, M., Wu, T.-L., and Peng, N. (2021). Hyperexpan: Taxonomy expansion with hyperbolic representation learning. Findings of the Association for Computational Linguistics: EMNLP , pages 4182–4194, November (16–20), Virtual Event / Punta Cana. Association for Computational Linguistics.

[47] Manzoor, E., Li, R., Shrouty, D., and Leskovec, J. (2020). Expanding taxonomies with implicit edge semantics. In Proceedings of The Web Conference 2020, pages 2044–2054, April (20-24), Taipei, Taiwan. ACM / IW3C2.

[48] Mao, Y., Zhao, T., Kan, A., Zhang, C., Dong, X. L., Faloutsos, C., and Han, J. (2020). Octet: Online catalog taxonomy enrichment with self-supervision. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2247–2257, August (23-27), Virtual Event, CA, USA. ACM.

[49] Menon, A. K. and Elkan, C. (2011). Link prediction via matrix factorization. In Joint european conference on machine learning and knowledge discovery in databases, pages 437–452, September (5–9), Athens, Greece. Springer.

[50] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. 1st International Conference on Learning Representations, ICLR, Workshop Track Proceedings, Scottsdale, May (2-4), Scottsdale, Arizona, USA.

[51] Miller, G. (1998). WordNet: An electronic lexical database. MIT press.

[52] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. International journal of lexicography, 3(4):235–244, Oxford University Press.

[53] Mish, F. C. et al. (2003). Merriam-webster's collegiate dictionary. 11th edspringfield (ma) merriam-webster.

[54] Murphy, M. L. and Koskela, A. (2010). Key terms in semantics. A&C Black.

[55] Nakashole, N., Weikum, G., and Suchanek, F. (2012). Patty: A taxonomy of relational patterns with semantic types. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1135–1145, July (12–14), Jeju Island, Korea. ACL.

[56] Navarro, E., Sajous, F., Gaume, B., Prévot, L., ShuKai, H., Tzu-Yi, K., Magistry, P., and Chu-Ren, H. (2009). Wiktionary and nlp: Improving synonymy networks. In Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, pages 19–27, 7 August, Singapore. Association for Computational Linguistics.

[57] Nguyen, K. A., Walde, S. S. i., and Vu, N. T. (2017). Distinguishing antonyms and synonyms in a pattern-based neural network. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Volume 1: Long Papers, pages 76–85, April (3–7), Valencia, Spain,.

[58] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. CoRR, abs/1807.03748.

[59] Ott, H. R., Rudolf, P., and Schweitzer, F. (1998). The European Physical Journal: Condensed Matter and Complex Systems. B. Springer.

[60] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[61] Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2):1–135, Now Publishers, Inc.

[62] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, October (25–29), Doha, Qatar. ACL.

[63] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pages 701–710, August (24 – 27), New York, NY, USA. ACM.

[64] Phukon, B., Mitra, A., Sanasam, R., and Sarmah, P. (2022). Team: A multitask learning based taxonomy expansion approach for attach and merge. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 366–378, July (10–15), Seattle, WA, United States. Association for Computational Linguistics.

[65] Ponzetto, S. P. and Navigli, R. (2009). Large-scale taxonomy mapping for restructuring and integrating wikipedia. In IJCAI Proceedings of the 21st International Joint Conference on Artificial Intelligence,, volume 9, pages 2083–2088, Pasadena, California, USA.

[66] Poprat, M., Beisswanger, E., and Hahn, U. (2008). Building a biowordnet using wordnet data structures and wordnet's software infrastructure–a failure story. In Software engineering, testing, and quality assurance for natural language processing, pages 31–39, June 20, Columbus, Ohio, USA. Association for Computational Linguistics.

[67] Pujari, M. and Kanawati, R. (2012). Link prediction in complex networks by supervised rank aggregation. In 2012 IEEE 24th International Conference on Tools with Artificial Intelligence, volume 1, pages 782–789, November (7–9), Athens, Greece. IEEE Computer Society.

[68] Qian, L., Zhou, G., Kong, F., and Zhu, Q. (2009). Semi-supervised learning for semantic relation classification using stratified sampling strategy. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, pages 1437–1445, August (6–7), Singapore. Association for Computational Linguistics.

[69] Qu, M., Ren, X., and Han, J. (2017). Automatic synonym discovery with knowledge bases. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 997–1005, August (13 – 17), Halifax, NS, Canada. ACM.

[70] Reichert, R., Olney, J., and Paris, J. (1969). Two dictionary transcripts and programs for processing them. volume i. the encoding scheme, parsent and conix. Technical report, SYSTEM DEVELOPMENT CORP SANTA MONICA CALIF.

[71] Roller, S., Erk, K., and Boleda, G. (2014). Inclusive yet selective: Supervised distributional hypernymy detection. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers, pages 1025–1036, August (23–29), Dublin, Ireland. ACL.

[72] Ruiz-Casado, M., Alfonseca, E., and Castells, P. (2005). Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In International Atlantic Web Intelligence Conference, pages 380–386, June (6–9), Lodz, Poland. Springer.

[73] Sarma, S. K., Gogoi, M., Saikia, U., and Medhi, R. (2010,). Foundation and structure of developing assamese wordnet. In 5th International Conference of the Global WordNet Association (GWC-2010), 31st Jan – 4th Feb, IIT Bombay, India. Global Wordnet Association.

[74] Schlichtkrull, M. and Alonso, H. M. (2016). Msejrku at semeval-2016 task 14: Taxonomy enrichment by evidence ranking. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pages 1337–1341, June 16–17, San Diego, CA, USA. The Association for Computer Linguistics.

[75] Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In The Semantic Web - 15th International Conference, ESWC, pages 593–607, June (3–7), Heraklion, Crete, Greece, Springer.

[76] Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein–protein interactions in yeast. Nature biotechnology, 18(12):1257–1261, Nature Publishing Group.

[77] Shen, J., Lyu, R., Ren, X., Vanni, M., Sadler, B., and Han, J. (2019). Mining entity synonyms with efficient neural set generation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 249–256, January 27 – February, Honolulu, Hawaii, USA, AAAI Press.

[78] Shen, J., Shen, Z., Xiong, C., Wang, C., Wang, K., and Han, J. (2020). Taxoexpan: self-supervised taxonomy expansion with position-enhanced graph neural network. In Proceedings of The Web Conference 2020, pages 486–497, April (20–24), Taipei, Taiwan, ACM. ACM / IW3C2.

[79] Shen, J., Wu, Z., Lei, D., Zhang, C., Ren, X., Vanni, M. T., Sadler, B. M., and Han, J. (2018). Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2180–2189, August (19–23), London, UK, ACM. ACM.

[80] Singhal, A. et al. (2001). Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4):35–43.

[81] Snow, R., Jurafsky, D., and Ng, A. (2004). Learning syntactic patterns for automatic hypernym discovery. Advances in neural information processing systems, 17.

[82] Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 801–808, July (17–21), Sydney, Australia. Association for Computational Linguistics.

[83] Steyvers, M. and Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. Cognitive science, 29(1):41–78, Wiley Online Library.

[84] Symeonidis, P. and Tiakas, E. (2014). Transitive node similarity: predicting and recommending links in signed social networks. World Wide Web, 17(4):743–776, Springer.

[85] Takeoka, K., Akimoto, K., and Oyamada, M. (2021). Low-resource taxonomy enrichment with pretrained language models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2747–2758, November(7–11), Virtual Event / Punta Cana, Dominican Republic. Association for Computational Linguistics.

[86] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web, pages 1067–1077, May (18–22), Florence, Italy. ACM.

[87] Toral, A., Muñoz, R., and Monachini, M. (2008). Named entity WordNet. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), 26 May - 1 June, Marrakech, Morocco. European Language Resources Association (ELRA).

[88] Tsitsulin, A., Mottin, D., Karras, P., and Müller, E. (2018). Verse: Versatile graph embeddings from similarity measures. In Proceedings of the 2018 world wide web conference, pages 539–548, April (23–27), Lyon, France. ACM.

[89] Ustalov, D., Chernoskutov, M., Biemann, C., and Panchenko, A. (2017). Fighting with the sparsity of synonymy dictionaries for automatic synset induction. In International Conference on Analysis of Images, Social Networks and Texts, pages 94–105, July (27–29), Moscow, Russia. Springer.

[90] Vedula, N., Nicholson, P. K., Ajwani, D., Dutta, S., Sala, A., and Parthasarathy, S. (2018). Enriching taxonomies with functional domain knowledge. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pages 745–754, July (08–12), Ann Arbor, MI, USA. ACM.

[91] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. 6th International Conference on Learning Representations, ICLR, April 30 - May 3, Vancouver, BC, Canada, OpenReview.net.

[92] Vossen, P. (1998). Introduction to eurowordnet. In EuroWordNet: A multilingual database with lexical semantic networks, pages 1–17. Springer.

[93] Wang, D., Cui, P., and Zhu, W. (2016). Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1225–1234. August (13–17), San Francisco, CA, USA. ACM. ACM.

[94] Wang, J., Lin, C., Li, M., and Zaniolo, C. (2019). An efficient sliding window approach for approximate entity extraction with synonyms. In Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT, pages 109–120, March (26–29), Lisbon, Portugal. OpenProceedings.org.

[95] Wang, P., Xu, B., Wu, Y., and Zhou, X. (2015). Link prediction in social networks: the state-of-the-art. Science China Information Sciences, 58(1):1–38, Springer.

[96] Wang, S., Zhao, R., Chen, X., Zheng, Y., and Liu, B. (2021). Enquire one's parent and child before decision: Fully exploit hierarchical structure for self-supervised taxonomy expansion. In Proceedings of the Web Conference 2021, pages 3291–3304, April (19–23), Virtual Event / Ljubljana, Slovenia. ACM / IW3C2.

[97] Wang, S., Zhao, R., Zheng, Y., and Liu, B. (2022). Qen: Applicable taxonomy completion via evaluating full taxonomic relations. In Proceedings of the ACM Web Conference 2022, pages 1008–1017, April (25 – 29), Virtual Event, Lyon, France. ACM.

[98] Wang, T. and Hirst, G. (2009). Extracting synonyms from dictionary definitions. In Proceedings of the International Conference RANLP-2009, pages 471–477, September (14–16), Borovets, Bulgaria. Association for Computational Linguistics.

[99] Wang, X. F. and Chen, G. (2003). Complex networks: small-world, scale-free and beyond. IEEE circuits and systems magazine, 3(1):6–20, IEEE.

[100] Weeds, J., Clarke, D., Reffin, J., Weir, D., and Keller, B. (2014). Learning to distinguish hypernyms and co-hyponyms. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2249–2259, August (23–29), Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

[101] Yamada, I., Oh, J.-H., Hashimoto, C., Torisawa, K., Kazama, J., De Saeger, S., and Kawada, T. (2011). Extending WordNet with hypernyms and siblings acquired from Wikipedia. In Proceedings of 5th International Joint Conference on Natural Language Processing, November (8-13), Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

[102] Yang, Y. and Hospedales, T. M. (2017). Trace norm regularised deep multi-task learning. 5th International Conference on Learning Representations, ICLR, April (24-26), Toulon. France, OpenReview.net.

[103] Yu, J., Shen, Y., Ma, X., Jia, C., Chen, C., and Lu, W. (2020a). Synet: Synonym expansion using transitivity. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 1961–1970, November (16–20), Online Event. Association for Computational Linguistics.

[104] Yu, Y., Li, Y., Shen, J., Feng, H., Sun, J., and Zhang, C. (2020b). Steam: Self-supervised taxonomy expansion with mini-paths. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1026–1035, August (23–27), Virtual Event, CA, USA. ACM.

[105] Zhang, J., Song, X., Zeng, Y., Chen, J., Shen, J., Mao, Y., and Li, L. (2021). Taxonomy completion via triplet matching network. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 4662–4670, February (2–9), Virtual. AAAI Press.

Publications

6.5 Publications

- Journal publications from thesis

  1. **Bornali Phukon**, Akash Anil,Sanasam Ranbir Singh, and Priyankoo Sarmah. **Synonymy Expansion Using Link Prediction Methods: A Case Study of Assamese WordNet**, ACM Trans. Asian Low-Resour. Lang. Inf. Process., Vol. 21, No. 1, Article 15. Publication date: November 2021.

- Conference publications from thesis

  1. **Bornali Phukon**, Anasua Mitra, Sanasam Ranbir Singh, and Priyankoo Sarmah. **TEAM: A multitask learning based Taxonomy Expansion approach for Attach and Merge**. Findings of the Association for Computational Linguistics: NAACL 2022, pages 366 - 378 July 10-15, Seattle, Washington, 2022 ©2022 Association for Computational Linguistic
  2. **Bornali Phukon**, Anasua Mitra, Sanasam Ranbir Singh, and Priyankoo Sarmah. **LG-TEAM: Local and Global context aware multitask learning based Taxonomy Expansion approach for Attach and Merge**. *(Submitted at The 61st Annual Meeting of the Association for Computational Linguistics)*
  3. **Bornali Phukon**, Sanasam Ranbir Singh, and Priyankoo Sarmah. **A Survey on Taxonomy Expansion: Issues, Resources and Recent Advances** . *(Preparing for submission)*