Hypernym Discovery over WordNet and English Corpora - using Hearst Patterns and Word Embeddings

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Manikya Swathi Vallabhajosyula

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Professor Ted Pedersen

July 2018

Acknowledgements

I would like to thank everyone who extended their support and trust throughout my research work.

It was a great pleasure working with Dr. Ted Pedersen as my advisor. I would like to thank him for the immense amount of time he dedicated towards my research work. Without his help It would be really difficult to finish my thesis within one year.

I would like to thank Dr. Pete Willemsen, Director of Graduate Studies, for his guidance throughout the process of research documentation. The seminar sessions conducted by him helped me to keep track of my research progress.

I would like to thank my committee members Dr. Peter Peterson and Dr. Chongwon Park for their time and valuable feedback.

I would like to thank Dr. James Allert for his support and motivation.

I would like to thank Arshia Z Hassan for her immense support and effort towards the SemEval 2018 Task. This research would be incomplete without her contribution.

I would like to thank my friends who supported me thoughtout this process. My friend Dr. Tejaswi Kodem was always there to motivate me irrespective of the time zone differences.

I would like to thank my family for their continuous support and trust. Thank you all.

Dedication

I would like to dedicate my work to my mother Anita Vemparala, sister Tejaswini and husband Krishna Perivilli.

Abstract

Languages evolve over time. With new technical innovations, new terms get created and new senses are added to existing words. Dictionaries like WordNet which act as a database for English vocabulary should be updated with these new concepts. WordNet organizes these concepts in sets of synonyms and interlinks them by using semantic relations. Many Natural Language Processing applications like Machine Translation and Word Sense Disambiguation rely on WordNet for their functionality. WordNet was last updated in 2006. If WordNet is not updated with new vocabulary, the performance of applications which rely on WordNet would drop. The objective of our research is to automatically update WordNet with the new senses by using resources like online dictionaries and text corpora available over the internet. We use the *ISA hierarchy* structure of WordNet to insert new senses. In an ISA hierarchy, the concepts higher in a hierarchy (called hypernyms) are more abstract representations of the concepts lower in hierarchy (called hyponyms). To improve the coverage of our solution, we rely on two complementary techniques - traditional pattern matching and modern vector space models - to extract candidate hypernym from WordNet for a new sense. Our system was ranked **4** among the systems that participated in for this SemEval task *SemEval 2016 Task 14 Semantic Taxonomy Enrichment*. We also evaluate our system by participating in the task *SemEval 2018 Task 09 Hypernym Discovery*. In this task, we apply our system to the huge UMBC WebBase text corpus to extract candidate hypernyms for a given input term. Our system was ranked 3^{rd} among the systems which find hypernyms for Concepts.

Contents

List of Tables

List of Figures

1 Introduction

Languages evolve. New words are created; new meanings are added to existing words or vocabulary from one language is borrowed by another language. New words can be created from existing words[1]. A new word can be created by truncating/clipping the existing word (for example, *lab* obtained from *laboratory*). A new word can be created by combining two or more existing words (*breakfast* + *lunch* is *brunch*). The word *phone* was added to the English vocabulary in the late 19^{th} century as an abbreviation for the existing word "*telephone*" [2]. The dictionaries which act like databases for this vocabulary should also be updated with these new concepts. *Oxford English Dictionary*[3] *(OED)* is one such dictionary which is updated with new vocabulary every quarter[4]. The most recent OED update happened in January 2018 and 700 new senses were added to its dictionary. Some new words like ***balisong***, ***e-publisher*** and ***jugaad*** were added to the English vocabulary. Where *Balisong*[5] *is a pocket knife with blades hidden in grooves* is the new sense added to the existing sense(s), *E-publisher*[6] *one who publishes content in electronic media* is a new sense created by adding a shortened prefix *E* (for electronic) to an existing word *publisher* and *Jugaad*[7] *is finding a cheaper alternative for a given problem or simply a life-hack* is a *Hindi* word added to English vocabulary.

[1] http://www.thehistoryofenglish.com/issues$_{n}ew.html$

[2] http://www.dictionary.com/browse/phone

[3] https://www.oxforddictionaries.com/oed

[4] https://public.oed.com/the-oed-today/recent-updates-to-the-oed/

[5] https://en.wiktionary.org/wiki/balisong

[6] https://en.wiktionary.org/wiki/e-publisher

[7] https://en.wiktionary.org/wiki/jugaad

WordNet[8] is a lexical database for English vocabulary. As the name suggests, it is the network of concepts where a network represents various relationships among these concepts. Two concepts could be related to one another as *synonyms* or *antonyms*. For example, the concepts *large* and *big* are related by synonymy and *large* and *small* are related by antonymy. A concept could be an abstract representation of another concept. For example, *electronic-device* is an abstract representation for the term *mobile phone*. In this case, these concepts are said to in a *hypernym-hyponym* relationship or an *ISA* relationship. In order to add a new concept to WordNet, all the dependent relationship links should also be updated. The semantic structure of WordNet enables it to be used in a wide range of Natural Language Processing applications such as Word Sense Disambiguation, Semantic Similarity Measurement, Query Expansion and Information Retrieval, Machine Translation and Sentiment Analysis. The performance of these applications depends on the vocabulary coverage of WordNet. Hence if WordNet is not updated with modern vocabulary, the results predicted by any dependent application would be obsolete. The most recent version of WordNet (3.0) was released in December 2006 with a total of 155,287 unique words aggregated into 117,659 synonym sets and a total word-sense pairs of 206,941. Unlike OED, the public release of WordNet is not updated since 2006. The task of manually updating WordNet is both intensive in terms of human effort and expensive. As such we need techniques to automatically update WordNet with new vocabulary and domain specific terms.

Through this research, we try to address the problem of automatically updating WordNet by predicting the hypernyms for some given concepts. There are two Semantic Evaluation (*SemEval*) tasks -*SemEval 2016 Task 14 Semantic Taxonomy Enrichment* [Camacho-Collados, Delli Bovi, Espinosa-Anke, Oramas, Pasini, Santus,

[8] https://wordnet.princeton.edu/

Shwartz, Navigli, and Saggion 2018] and *SemEval 2018 Task 09 Hypernym Discovery* [Jurgens and Pilehvar 2016] - proposed to address this problem. There is an existing system [Rusert and Pedersen 2016] which addresses the semantic taxonomy enrichment problem by using word similarity algorithms over the definitions of the new Out-of-Vocabulary terms and existing sense glosses from WordNet. We propose some new techniques[9] to address the problem of identifying hypernyms for a given set of concepts. We hypothesize that some traditional pattern matching algorithms [Hearst 1992] could be used along with modern vector space models [Mikolov, Chen, Corrado, and Dean 2013] to predict a set of hypernyms (one or more) from a pre-defined vocabulary. Hearst Patterns 2.5.3 are used for our pattern matching algorithm. For example, Hearst Pattern like "⟨*hypernym*⟩ ***such as*** ⟨*hyponym_1*⟩, ⟨*hyponym_2*⟩, *....,(and | or)* ⟨*hyponym_n*⟩" is used over a target text to extract a hypernym and a list of hyponyms. If this pattern is applied over a target text "*Electronic devices* **such as** *tablet, personal computer or mobile phone* are needed for the course work.", then the hypernym-hyponym pair extracted is *"(electronic devices, {tablet, personal computer, mobile phone})"*. We used *Word2Vec* algorithm proposed by Mikolov [Mikolov, Sutskever, Chen, Corrado, and Dean 2013] to build word embedding matrix over a huge plain text corpus. Vector similarity is applied over these embeddings to extract candidate hypernyms for the given concepts.

Our system queries different resources to identify potential hypernyms for the target terms. The first resource which we used in this research is a structured lexical database for English vocabulary - *WordNet* [Fellbaum 1998]. Word embedding matrices are built over all the noun glosses and verb glosses from WordNet by using Word2Vec algorithm. The second resource which we used is the UMBC WebBase Corpus [Han, Kashyap, Finin, Mayfield, and Weese 2013]. This is a huge plain text corpus

[9] Techniques different from existing system

with three billion part-of-speech tagged words. All possible hypernym-hyponym candidates are extracted from this corpus by using Hearst Patterns. A word embedding matrix is built over this UMBC corpus by using Word2Vec algorithm. Finally, our system also uses *Google News Vectors*[10] to extract hypernyms.

Though the underlying problem of both SemEval tasks is hypernym discovery, the detailed descriptions of these tasks differ. Apart from the input and output format, the systems proposed for these tasks also differ in the combination of resources they use to predict the candidate hypernyms.

For *SemEval 2016 Task 14 Semantic Taxonomy Enrichment* task, the provided input term is a new Out-Of-Vocabulary (OOV) term which does not exist in WordNet. A definition and a part-of-speech (POS) tag are provided along with each OOV term. We use all the provided resources to identify a candidate synset from WordNet which satisfies a hypernym-hyponym relationship with an input OOV term. For example, if the provided input is "(*emergicenter*[11], *noun*, *'a clinic, often in a shopping mall, offering immediate outpatient treatment for minor ailments and injuries'*)". Then the result predicted by our system is *clinic#n#1*. Finally, we apply *Word Sense Disambiguation* algorithms like Lesk [Lesk 1986] and Extended Gloss Overlaps [Banerjee and Pedersen 2003] to refine the sense assigned to *clinic*. The final result *clinic#n#3* is reported as a hypernym for the given OOV term *emergicenter*. Multiple resources are used here in order to improve the coverage of our system. When one resource fails to fetch a result hypernym, another resource might fetch a candidate hypernym for a given input term.

Our system was able to perform better than the lower baseline system which predicts a random hypernym 4.2.1. The *Wu & Palmer similarity* score of our system

[10] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[11] https://en.wiktionary.org/wiki/emergicenter#English

was 25% higher than this baseline. It performed on par with a more sophisticated baseline system which chooses the first word from the OOV term's definition as the hypernym 4.2.2. The *Wu & Palmer similarity* score of our system was only 4% lower than this baseline. Our system was ranked 4^{th} when considering both nouns and verbs. It was ranked 3^{rd} when we ignored all the verbs and considered only noun inputs.

For *SemEval 2018 Task 09 Hypernym Discovery* task, the provided input is a noun which could either represent a Concept (common noun) or an Entity (proper noun). The UMBC WebBase corpus is used to identify a list of candidate hypernym terms from a pre-defined vocabulary corpus. For example, if the provided input is "(*buckler*, *concept*)". Then the result predicted by our system is {*buckler, shield, armor, interest, men, fight, scabbard, breastplate, dagger, sword, steed, hand*}. Our system cannot predict more than **15** candidate hypernyms for one input term.

Our system was able to perform better than the baseline system proposed for this task - *TaxoEmbed:Supervised Distributional Hypernym Discovery via Domain Adaptation*[12] [Espinosa-Anke, Camacho-Collados, Delli Bovi, and Saggion 2016]. We submitted our system for the *Concept-only* English sub-task with a resource limitation[13]. Our system was ranked 3^{rd} among all the submitted systems under this category[14]. If we ignore the resource limitation, our system is ranked 5^{th} with respect to *Mean Average Precision* (MAP) score and 3^{rd} with respect to *Mean Reciprocal Rank* (MRR) score. We submitted another run for the *Entity-only* English sub-task with a resource limitation. Our system was ranked 9^{th} with respect to MAP score and ranked **11** without the resource limitation.

[12] http://wwwusers.di.uniroma1.it/ dellibovi/taxoembed/

[13] Should not use any other resource apart from UMBC WebBase corpus

[14] Ranked according to mean average precision (MAP) score

2 Background

In this research we focus on identifying the **ISA semantic relationship** between two concepts by using some given corpora. An ISA relationship is called hypernymy or inclusion relationship where the meaning of one concept includes the meaning of another concept. The word which includes the meaning of another concept is called the *hypernym* and the concept whose meaning is included is called *hyponym*. In this section we discuss the resources and background concepts used in this thesis. We start by describing the keywords used throughout this documentation.

2.1 Keywords

Semantic relationship is a relationship which exists between the meanings of different concepts. Some examples of semantic relationships include - synonymy, antonymy, hypernymy, meronymy and homonymy.

- ***Synonymy*** is a relationship in which concepts share the same meaning when used in similar contexts. These words could be used interchangeably. For example, the words *ring* and *band* share the same meaning "jewelry worn on a finger". So *ring* and *band* are synonyms to one another.

- ***Antonymy*** is a relationship in which the meaning of one concept contradicts the meaning of the other concept. For example, the words *fear* and *fearlessness* are opposite words.

- ***Hypernymy*** is a relationship between a concept with more general meaning and a concept with a more specific meaning. We can say that the meaning of one concept includes the meaning of another concept. For example, the word *vehicle* and *car* hold a hypernymy relationship. The concept with a general meaning is called a *hypernym* and the concept with a specific meaning is called a *hyponym*. In the *vehicle-car* example, *vehicle is a hypernym* and *car is a hyponym*.

- ***Meronymy*** is a part-whole relationship where one word describes a part of another word. For example, the word *wheel* describes the part of the word *vehicle* which enables movement.

- ***Homonymy*** is a relationship between two words which share the same spelling but a different meaning. For example, the word ***lead*** could mean "an advantage held by a competitor in a race" or "a soft heavy toxic malleable metallic element; bluish white when freshly cut but tarnishes readily to dull grey" based on the context of usage[1]. So we say that the word *lead* has two different ***senses***.

2.2 Word Sense Disambiguation

As described in the *keywords* section, words which have the same spellings but different meanings are called *homonyms* or *homographs*. Homonyms are words which share the same spelling and pronunciation but have different meaning or sense. For example, the word *ruby* could mean a *color* (a shade of red) or a *stone* (a precious gem). Homographs are words which have same spelling but different pronunciations or meaning. For example, the word *wind* could either mean *speeding air* or *winding the*

[1] http://wordnetweb.princeton.edu/perl/webwn?s=lead

clock. *Word Sense Disambiguation* is the *Natural Language Processing* application which addresses the problem of identifying an appropriate sense for a given word based on the context in which it is used. For example, if a given sentence is "She **stood on** the bank to fetch some **water**" with the target word "*bank*", then the meaning of this word is *"sloping land (especially the slope beside a body of water)"*[2]. This sense for the word *bank* could be identified with the help of other context words such as *water* or *stood on*. This is called word sense disambiguation.

2.3 Word Similarity

Two words are considered similar to one another if relationships like hypernymy or synonymy hold between them. For example, the word *apple* in the sentence "An apple a day keeps the doctor away" is more similar to the words *banana* or *fruit* than the words *device* or *vehicle*. Word similarity technique could be used to identify a hypernym, hyponym or a synonym for a given word. We could use structured knowledge base like WordNet 2.4.1 or unstructured text files corpus like UMBC WebBase corpus 2.4.2 to identify these relationships.

2.4 Available Resources

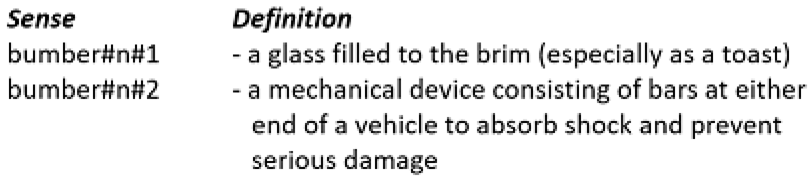
2.4.1 WordNet

WordNet is a lexical database for English vocabulary where words are organized into groups of nouns, verbs, adjectives and adverbs. Each word is assigned to a synonym set[3]. These sets are called ***synsets***. Unlike traditional dictionaries which

[2] http://wordnetweb.princeton.edu/perl/webwn?s=bank

[3] concepts which have same meaning

store English vocabulary in alphabetical order, WordNet organizes these synsets into an interlinked network of semantic relations. Some examples of these relations include hypernymy, meronymy, synonymy and antonymy. Homonyms are words with the same spelling but different meanings. WordNet differentiates these homonyms by assigning sense numbers to them. For example, the word *bumper*[4] has two meanings and each is assigned a sense number in WordNet (Figure 2.1). A word in WordNet is represented as ⟨*word#part-of-speech_tag#sense_num*⟩. The most frequent sense of the word is given the sense number *#1*. ***Lemma*** is the *word* part of the synset without the part-of-speech tag and the sense number. These sense numbers help with Word Sense Disambiguation problem. Noun and verb synsets are connected in hypernymy and antonymy relations. Nouns synsets hold meronymy relationship between them. Antonymy relation holds between adjective synsets. Figures 2.2, 2.3 and 2.4 show a few examples for these WordNet relationships.



Sense	***Definition***
bumber#n#1	- a glass filled to the brim (especially as a toast)
bumber#n#2	- a mechanical device consisting of bars at either end of a vehicle to absorb shock and prevent serious damage

Figure 2.1: Example synsets with definitions from WordNet

These WordNet structures makes it possible to work with various natural language processing applications like Machine Translation, Word Sense Disambiguation and Automatic Text Classification. Each hypernymy relationship connects a hypernym with one of its hyponyms. A meronymy relationship connects a part holonym to a part meronym. A meronym represents a part of the whole concept holonym. For example, *wing* or *blade* is a part meronym to the whole concept *fan*, where *fan* is considered as a holonym. When synsets are connected to one another using the hypernymy

[4] http://wordnetweb.princeton.edu/perl/webwn?s=bumper

relationship, they form an *ISA* hierarchy tree. This ISA structure of WordNet is useful in determining the similarity between two given synsets. All the nouns in WordNet are arranged into one single ISA hierarchy with the root hypernym as *entity#n#1*. However, verbs have multiple disconnected ISA hierarchies in WordNet. Please refer Wu&Palmer Similarity section 4.1.1 in the Results Chapter (Chapter 4) for more details about word similarity.

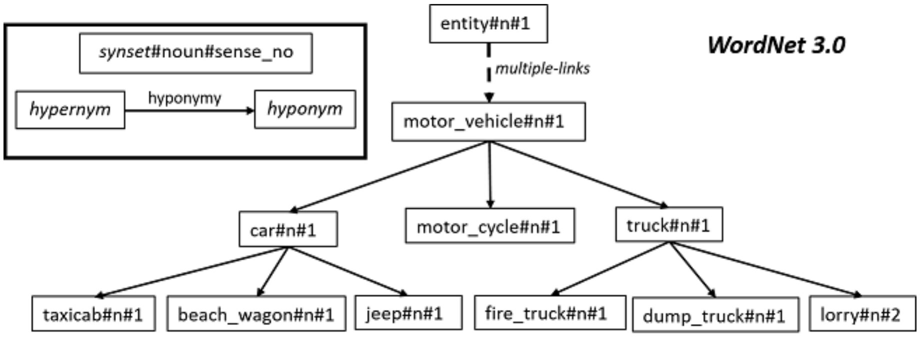


Figure 2.2: Example synsets with hypernymy from WordNet

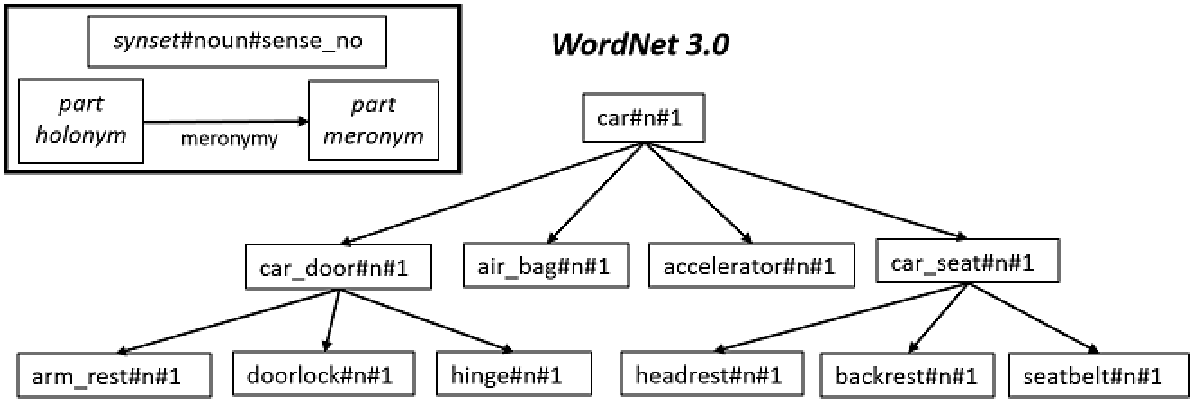


Figure 2.3: Example synsets with meronymy from WordNet

2.4.2 UMBC WebBase Corpus

UMBC WebBase Corpus is constructed as a resource which is built as part of the publication UMBC EBIQUITY CORE Semantic Textual Similarity Systems [Han,

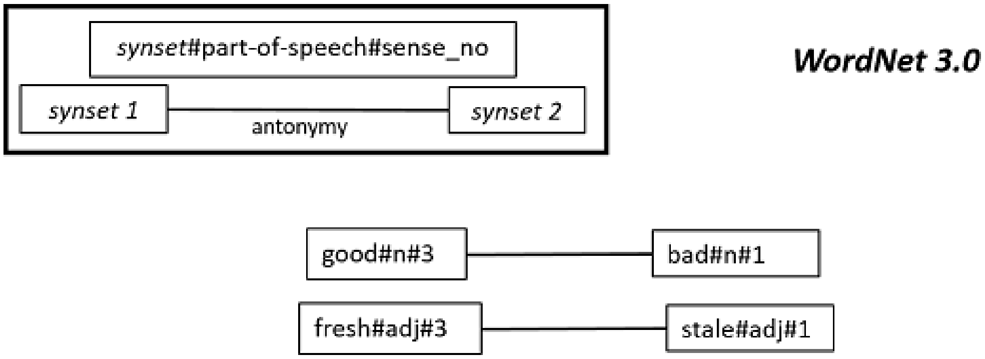


Figure 2.4: Example synsets with antonymy from WordNet

Kashyap, Finin, Mayfield, and Weese 2013]. This is an English text corpus of approximately **28.5**GB size. This corpus is derived from the Stanford WebBase Project's February 2007 Web crawl, which is one of the largest collections of English data. It consists of data from hundred million web pages gathered from over fifty thousand websites. The creators of the Stanford WebBase Crawl corpus could successfully extract text from html tags but could not eliminate special characters, non-English text and duplicated content. Through the UMBC WebBase corpus, the creators Lushan Han and Tim Finin handled these problems. The following are the characteristics of the UMBC WebBase Corpus:

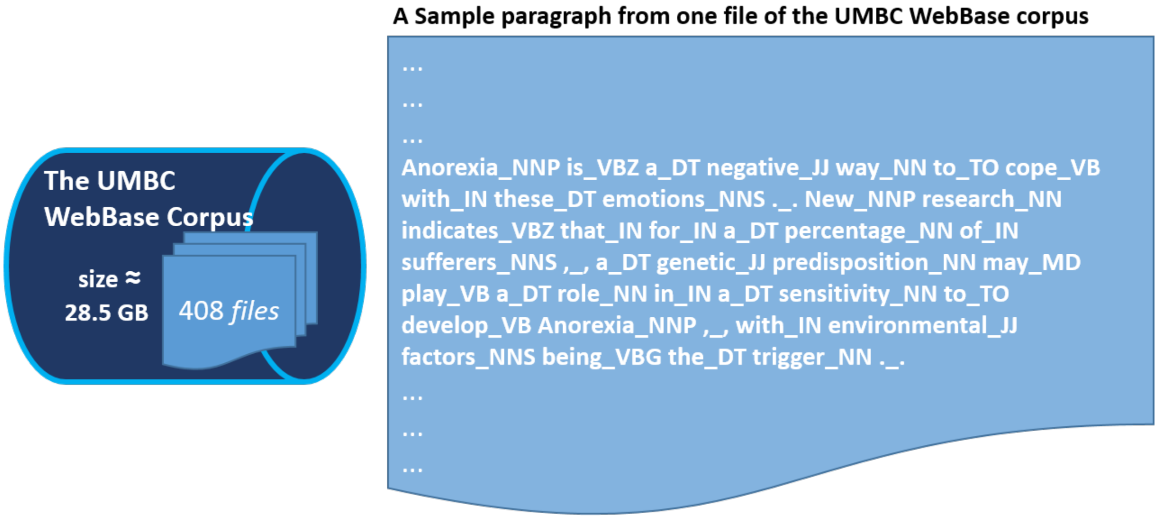
- A Paragraph from Stanford WebBase crawl corpus (or Stanford corpus) would be added to the UMBC WebBase corpus (or UMBC corpus) if and only if the number of characters in this paragraph is more than 200.

- Only English paragraphs from Stanford corpus are included in the UMBC Corpus.

- There are no duplicate paragraphs in the entire UMBC corpus.

- Stanford Part-Of-Speech Tagger is used to assign the part of speech tags to all the words in the UMBC corpus [Toutanova and Manning 2000].

Figure 2.5 represents the organization of the UMBC WebBase corpus.



…
…
…
Anorexia_NNP is_VBZ a_DT negative_JJ way_NN to_TO cope_VB with_IN these_DT emotions_NNS ._. New_NNP research_NN indicates_VBZ that_IN for_IN a_DT percentage_NN of_IN sufferers_NNS ,_, a_DT genetic_JJ predisposition_NN may_MD play_VB a_DT role_NN in_IN a_DT sensitivity_NN to_TO develop_VB Anorexia_NNP ,_, with_IN environmental_JJ factors_NNS being_VBG the_DT trigger_NN ._.
…
…
…

Figure 2.5: The UMBC WebBase Corpus and a sample paragraph from the corpus

2.4.3 Google News Vectors

Google News Vectors is one of the many pre-trained vectors created as part of the word2vec project[5]. Google News Data set with more than **100** billion words were used as an input corpus to build an embedding matrix with **3** million words and **300** vector dimensions.

2.4.4 SemEval Tasks

As part of this research, we addressed two SemEval tasks with the same underlying problem, i.e., identifying hypernyms for a given input term. A hypernym presents a more abstract representation of a more specific concept or term. SemEval is an international workshop on Semantic Evaluation [Jurgens and Pilehvar 2016]. It is derived

[5] https://code.google.com/archive/p/word2vec/

from another international workshop SensEval (for Word Sense Disambiguation). We created solutions which address hypernym discovery for these two tasks:

1. SemEval 2016 Task 14 - Semantic Taxonomy Enrichment

2. SemEval 2018 Task 09 - Hypernym Discovery

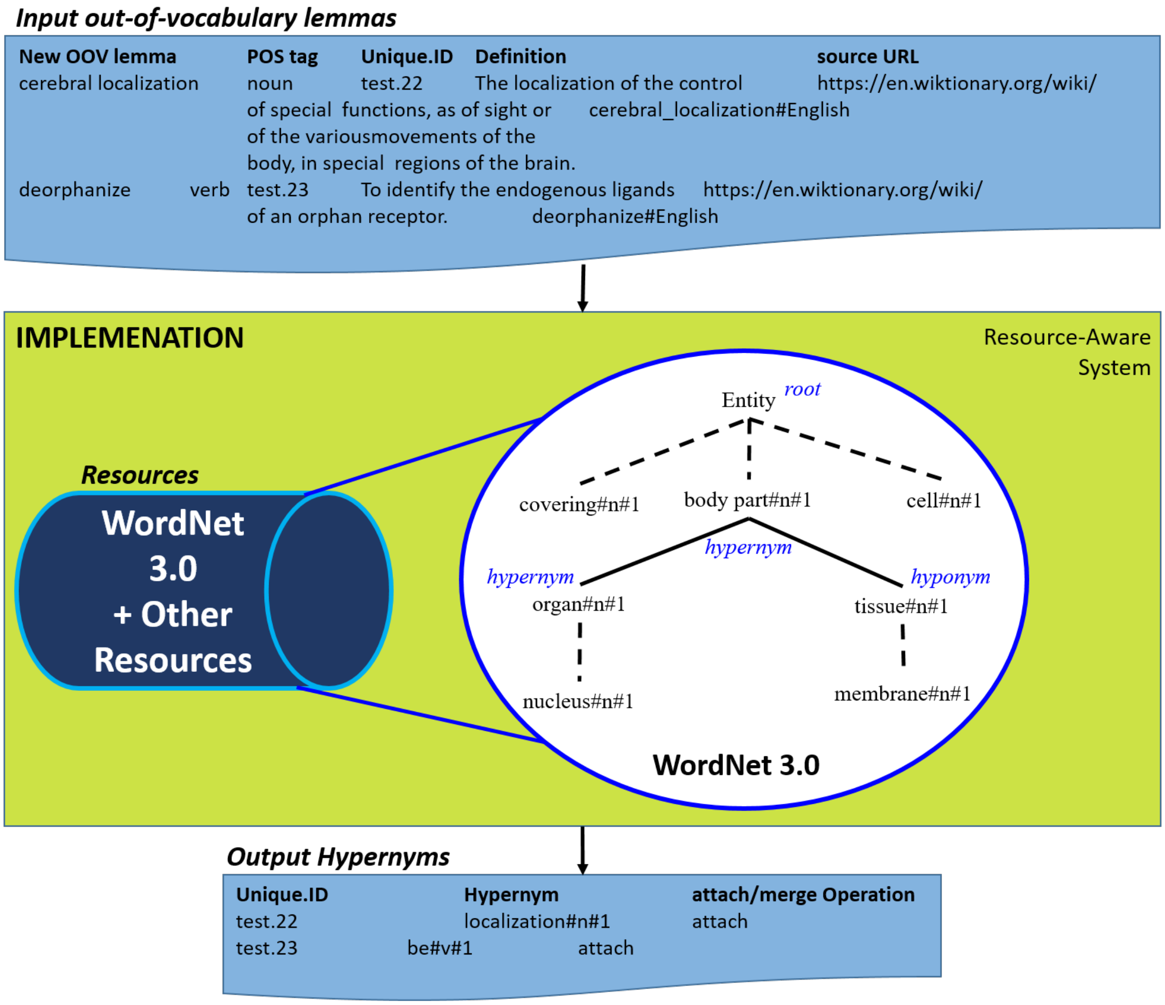
SemEval 2016 Task 14 - Semantic Taxonomy Enrichment

SemEval 2016 is the 10^{th} workshop on semantic evaluation [Jurgens and Pilehvar 2016]. WordNet provides semantic information about concepts and how these concepts are related to one another. Task 14 - Semantic Taxonomy Enrichment is a task designed to identify an optimal location to insert a new out-of-vocabulary lemma into WordNet's ISA hierarchy. For this task, a new sense (an OOV lemma) is provided along with its definition. The proposed system should identify a synset which could either be a *hypernym* or a *synonym* to the given synset.

- The predicted synset is a hypernym synset if it generalizes the meaning of the given OOV lemma.

- The predicted synset is a synonym if all its senses shares meaning with the given OOV lemma.

In this research we chose to only identify a synset which holds hypernymy relationship with the new OOV lemma. A given input term could either be a noun OOV lemma or a verb OOV lemma. The new OOV lemmas and their definitions are gathered from several websites like `http://www.genome.gov/`, `https://en.wiktionary.org/`, `https://www.lpi.org/` and `https://en.wikipedia.org/`. From Figure 2.6, we could see the format of the input provided with this task. The first part is the new OOV lemma followed by its part-of-speech tag. A unique identification number with a text prefix like $test\dot{2}2$ is provided to differentiate the results. Finally, a definition along with the resource url is provided with the new lemma. The implementations are considered as resource aware systems and are evaluated accordingly. A resource aware system is a one which relies on a dictionary like Wiktionary or WordNet to fetch the results. The input data is divided into training and test data with **400** and **600** new OOV lemmas respectively. Table 2.1 shows the noun and verb counts from these data sets.



Input out-of-vocabulary lemmas

New OOV lemma	POS tag	Unique.ID	Definition	source URL
cerebral localization	noun	test.22	The localization of the control of special functions, as of sight or of the variousmovements of the body, in special regions of the brain.	https://en.wiktionary.org/wiki/cerebral_localization#English
deorphanize	verb	test.23	To identify the endogenous ligands of an orphan receptor.	https://en.wiktionary.org/wiki/deorphanize#English

Output Hypernyms

Unique.ID	Hypernym	attach/merge Operation
test.22	localization#n#1	attach
test.23	be#v#1	attach

Figure 2.6: SemEval 2016 Task 14 - Semantic Taxonomy Enrichment task data

	Nouns	**Verbs**
Training Data	349	51
Test Data	517	83

Table 2.1: Noun and Verb counts for training and test data - SemEval 2016 Task 14 : Semantic Taxonomy Enrichment

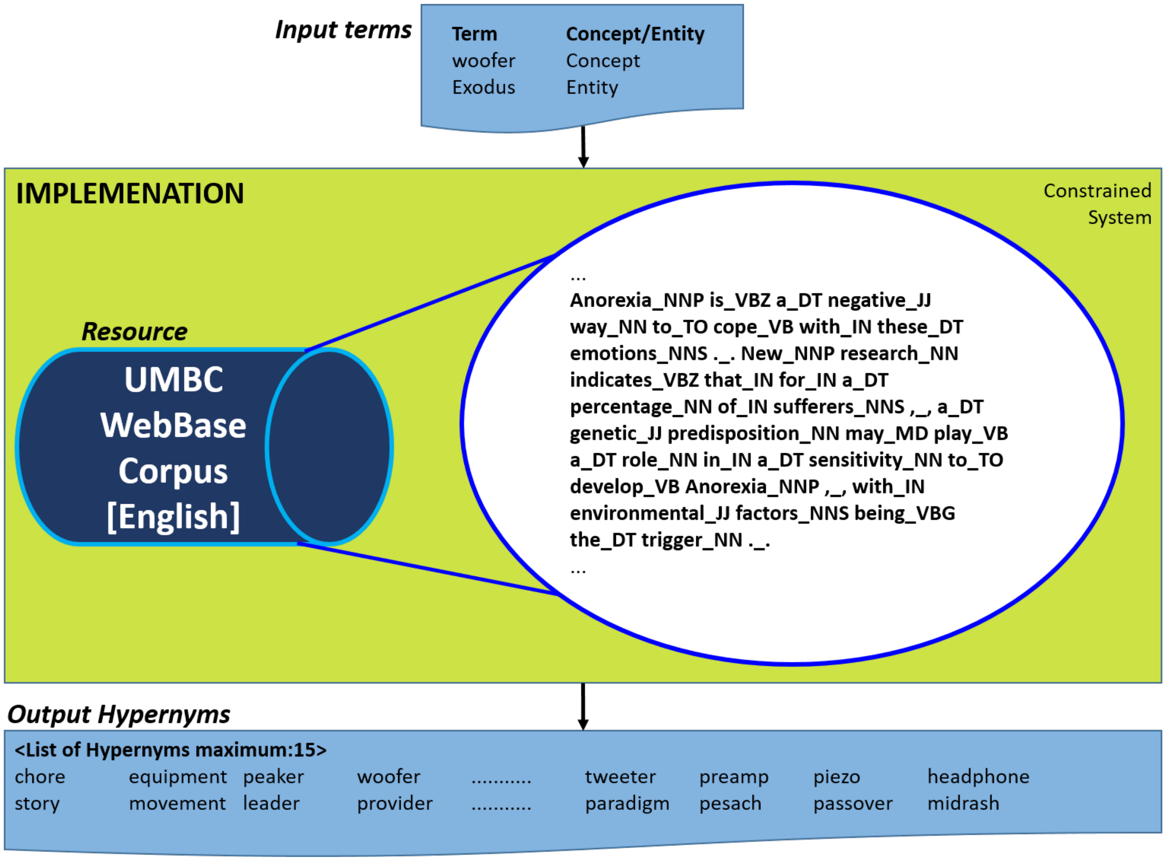
SemEval 2018 Task 09 - Hypernym Discovery

SemEval 2018 is the 12^{th} workshop on semantic evaluation [Camacho-Collados, Delli Bovi, Espinosa-Anke, Oramas, Pasini, Santus, Shwartz, Navigli, and Saggion 2018]. In this task, for a given term a system should identify the most appropriate hypernyms from a given pre-defined corpus. This task has five independent sub-tasks classified into two groups: *General-Purpose Hypernym Discovery* and *Domain-Specific Hypernym Discovery*. There are three General-Purpose Hypernym Discovery tasks addressing three different languages: English (Subtask 1A), Italian (Subtask 1B) and Spanish (Subtask 1C). The other two tasks for Domain-Specific Hypernym Discovery include medical (Subtask 2A) and music (Subtask 2B) domains. For any subtask, the participants were supposed to either work for *Concepts* or *Entities*. In this research we consider the General-Purpose Hypernym Discovery English subtask. We analyze out implementations against both *Concepts-only* and *Entities-only* subtasks. The pre-defined corpus provided for this task is the UMBC WebBase Corpus 2.4.2. For this task, we were required to rely only on the UMBC WebBase Corpus and were not supposed to use any other resource. There are no unique identification numbers provided with the input terms so the order of the inputs is important and should be retained in the output file. If a system fails to fetch a hypernym list for a given input, then a blank line should be added to the output file. This system is called a *Constrained* system as we are not using any other system apart from the UMBC WebBase Corpus. The input data for the English subtask contains **3000** input terms

	Concepts	Entities
Training Data	979	521
Test Data	1057	443

Table 2.2: Concept and Entity counts for training and test data - SemEval 2018 Task 09 - Hypernym Discovery task data

which are equally divided between training and test data. Table 2.2 shows the counts for concepts and entities in these data sets.


Input terms
IMPLEMENATION
Constrained System
Resource
UMBC WebBase Corpus [English]
Output Hypernyms
<List of Hypernyms maximum:15>
chore equipment peaker woofer .... tweeter preamp piezo headphone
story movement leader provider .... paradigm pesach passover midrash

Term	Concept/Entity
woofer	Concept
Exodus	Entity

...
Anorexia_NNP is_VBZ a_DT negative_JJ way_NN to_TO cope_VB with_IN these_DT emotions_NNS ._. New_NNP research_NN indicates_VBZ that_IN for_IN a_DT percentage_NN of_IN sufferers_NNS ,_, a_DT genetic_JJ predisposition_NN may_MD play_VB a_DT role_NN in_IN a_DT sensitivity_NN to_TO develop_VB Anorexia_NNP ,_, with_IN environmental_JJ factors_NNS being_VBG the_DT trigger_NN ._.
...

Figure 2.7: SemEval 2018 Task 09 - Hypernym Discovery task data

2.5 Related Works

2.5.1 Lesk Algorithm and its variations

The Lesk algorithm was proposed to determine the appropriate sense of a word based on the context in which it appears. There are three variations of the Lesk algorithm: Simplified Lesk, Original Lesk and Adapted Lesk algorithms. In Simplified Lesk, the gloss overlap score is used to determine a more appropriate sense for the target word. For example, suppose the given instance is "She is cooking the food in the pan over the stove." and the target word is *pan*. The following are the steps followed by Simplified Lesk algorithm to assign a sense to the chosen target word. [Kilgarriff and Rosenzweig 2000]

1. Fetch the glosses for all *pan* senses from WordNet. To simplify this example we use the definitions of only two noun senses of the word *pan*[6].

   - *pan#n#1* with definition "cooking utensil consisting of a wide metal vessel"

   - *pan#n#2* with definition "(Greek mythology) god of fields and woods and shepherds and flocks"

2. Compare the context from the instance with these glosses and assign an overlap score for each sense. Stop words are not considered for gloss overlap score.

   - Gloss Overlap score with *pan#n#1*: the gloss "**cooking** utensil consisting of a wide metal vessel" is compared with the instance "she is **cooking** the food in the pan over the stove". The overlap score is **1** as they have only one word in common.

[6] http://wordnetweb.princeton.edu/perl/webwn?s=pan

- Gloss Overlap score with *pan#n#2*: the gloss "(Greek mythology) god of fields and woods and shepherds and flocks" is compared to the instance "she is cooking the food in the pan over the stove". The overlap score is **0** as they do not have any word in common.

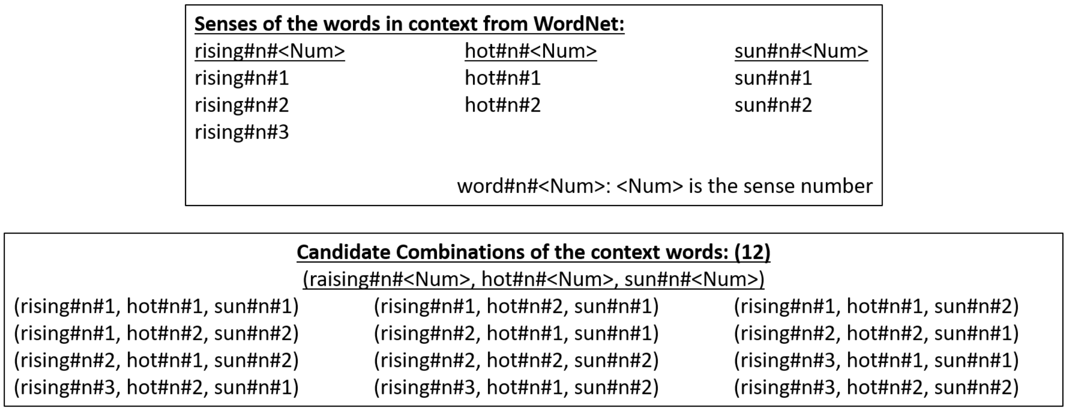
3. The sense with high overlap score is considered as the resulting sense for the target word *pan* with respect to the given instance. So *pan#n#1* is the sense of the word *pan* in the given instance "She is cooking the food in the pan over the stove."

In the paper *Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone* [Lesk 1986], the author Michael E. Lesk proposed a more sophisticated approach for word sense disambiguation which still relies on the gloss overlap score. This algorithm is called *Original Lesk* algorithm. The Original Lesk algorithm considers the glosses of all the words in context of the given target word from the given instance to compute the overall overlap score.

Adapted Lesk Algorithm [Banerjee and Pedersen 2002] is another variation of Lesk algorithm which uses WordNet as a source for the definitions/glosses to calculate the overlap score. *Target word, instance, context, candidate combination* and *combination score* are the few keywords defined in this paper. A *target word* is the word for which a sense is to be assigned. An *instance* is the given sentence containing the target word. *Context* is the words which appear along with the target word in a given sentence. If the context window is of size $n = 1$, then the number or tokens/words in context is $2 * n + 1 = 3$, including the target word. For example, consider the sentence "*A beautiful morning with rising hot sun from the east*" as an instance with target word *hot* and context window of size 1. Number of words in context is 3 and they are *rising, hot, sun.* Each word in context could have multiple meanings and hence could have multiple senses in WordNet. For example, the words *r, hot, sun* have *3, 2* and *2* number of senses respectively. *Candidate Combination* is a combination obtained by combining each sense of one word with the senses of other words. Figure 2.7 shows the senses of the words *raining, hot, sun* and their possible combinations. *Combination score* is computed for all the candidate combinations one at a time. The overlap score is computed for each pair in a chosen combination and all the overlap scores are combined to get the overall combination score. For example, lets consider the combination tuple *(rising#n#3, hot#n#1, sun#n#2).* The overlap score is computed between the pairs (rising#n#3, hot#n#1), (rising#n#3, sun#n#2) and (hot#n#1, sun#n#2). These overlap scores are combined to give a combination score for the tuple (rising#n#3, hot#n#1, sun#n#2). Similarly, combination scores for all the 12 tuples shown in Figure 2.8 are calculated. The *Candidate Combination* with the highest *Combination Score* is the chosen result combination of sense for a given instance. The sense of the word *hot* in this tuple is the result sense derived from this algorithm. For example, lets assume that the combination (rising#n#1, hot#n#1, sun#n#1) has the highest combination score and hence the sense for the word *hot* is *hot#n#1.*

2.5.2 Extended Gloss Overlap

Extended Gloss Overlap or ***EGO*** [Banerjee and Pedersen 2003] is an algorithm which is used to calculate the similarity score between two given concepts (senses from WordNet). In the previous Lesk based algorithms, the gloss overlap score between two concepts "A" and "B" was determined by counting the number of common words or phrases between the glosses of *A* and *B*. But with EGO, the glosses



Senses of the words in context from WordNet:

rising#n#<Num>	hot#n#<Num>	sun#n#<Num>
rising#n#1	hot#n#1	sun#n#1
rising#n#2	hot#n#2	sun#n#2
rising#n#3		

word#n#<Num>: <Num> is the sense number

Candidate Combinations of the context words: (12)

(raising#n#<Num>, hot#n#<Num>, sun#n#<Num>)

(rising#n#1, hot#n#1, sun#n#1)	(rising#n#1, hot#n#2, sun#n#1)	(rising#n#1, hot#n#1, sun#n#2)
(rising#n#1, hot#n#2, sun#n#2)	(rising#n#2, hot#n#1, sun#n#1)	(rising#n#2, hot#n#2, sun#n#1)
(rising#n#2, hot#n#1, sun#n#2)	(rising#n#2, hot#n#2, sun#n#2)	(rising#n#3, hot#n#1, sun#n#1)
(rising#n#3, hot#n#2, sun#n#1)	(rising#n#3, hot#n#1, sun#n#2)	(rising#n#3, hot#n#2, sun#n#2)

Figure 2.8: Set of possible candidate combinations

of the concepts which are related to concepts A and B are also used to compute the similarity score between them. These concepts are the synsets which satisfy some semantic relationship with the given concepts A and B in WordNet. A set $RELS$ is created with the relations of our interest from WordNet. Formula 2.1 shows some pair-wise permutations of these relationships. These relationship pairs are stored in set $RELPAIRS$.

$$\begin{gathered} RELPAIRS = \{(R_1, R_2) \| R_1, R_2 \in RELS; \\ if(R_1, R_2) \in RELPAIRS, then(R_2, R_1) \in RELPAIRS\} \end{gathered} \tag{2.1}$$

Each pair from $RELPAIRS$ is applied in to the concepts A and B and a pair-wise gloss overlap score is computed. This value is represented by the function *score*. This function takes in two glosses and return the gloss overlap score. The *score* is computed for all the pairs with respect to concepts A and B. Then these calculated values are combined to give the *Relatedness* or *EGO* score between the two given concepts. Formula 2.2 shows how this score is calculated.

$$\begin{aligned} relatedness(A, B) = \sum score(R_1(A), R_2(B)) \\ \forall (R_1, R_2) \in RELPAIRS \end{aligned} \tag{2.2}$$

For example, if the $RELS$ $\bar{\{}$gloss, hyper, hypo$\}$ where *gloss* is the definition of the concept of interest, *hyper* is the definition of the hypernym of the concept of interest and *hypo* is the definition of the hyponym of the concept of interest. The permutations selected are given by $RELPAIRSS$ $\bar{\{}$(gloss, gloss), (hyper, hyper), (hypo, hypo), (hyper, gloss), (gloss, hyper)$\}$. The *Relatedness* or similarity score between the two concepts A and B is be given by Formula 2.3.

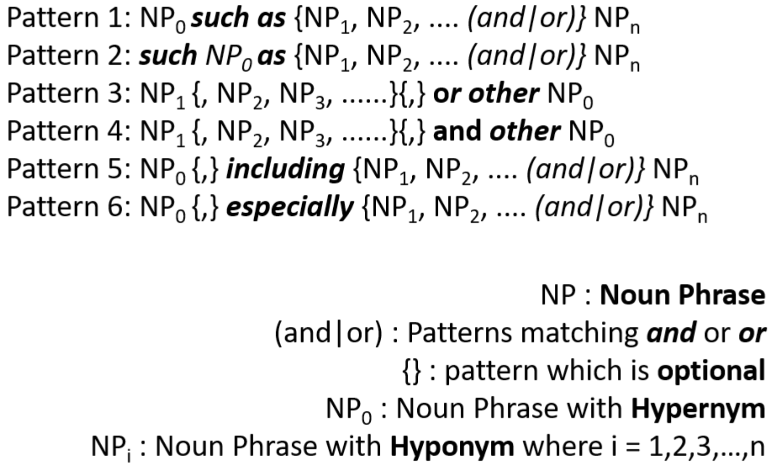
$$\begin{aligned} relatedness(A, B) = score(gloss(A), gloss(B)) \\ +score(hyper(A), hyper(B)) + score(hypo(A), hypo(B)) \\ +score(hyper(A), gloss(B)) + score(gloss(A), hyper(B)) \end{aligned} \tag{2.3}$$

Extended gloss overlap score can replace the traditional gloss overlap score in Adapted Lesk algorithm. Hence EGO score could be used for word sense disambiguation. Consider the example where the target word is *apple* and the given instance is *Apple tastes better than seed.* The *Candidate Combination* pairs are created for all the senses of the words *apple*, *taste* and *seed.* For ease of explanation, assume that *apple* has two senses, *taste* and *seed* have one sense each. The candidate pairs are *(apple#n#1, taste#n#1, seed#n#1)* and *(apple#n#2, taste#n#1, seed#n#1).* The relatedness score could be calculated between the pairs *(apple#n#1, taste#n#1)*, *(apple#n#1, seed#n#1)* and *(taste#n#1, seed#n#1)* for candidate *(apple#n#1,* *taste#n#1, seed#n#1)* and pairs *(apple#n#2, taste#n#1)*, *(apple#n#2, seed#n#1)* and

(taste#n#1, seed#n#1) for candidate *(apple#n#2, taste#n#1, seed#n#1)*. These scores are used to compute the respective *Combination Scores* for the two candidate combinations. The sense for the word *apple* in the candidate with the higher combination score is the result sense with respect to the given instance. For example, the combination score for candidate *(apple#n#1, taste#n#1, seed#n#1)* is **57** and candidate *(apple#n#2, taste#n#1, seed#n#1)* is **2**. So the sense for the word apple in instance "Apples tastes better than seeds" is *apple#n#1*.

2.5.3 Hearst Patterns

Hearst patterns were introduced in the paper *Automatic Acquisition of Hyponyms from Large Text Corpora* by Marti Hearst [Hearst 1992]. The key objective of this research was to extract hypernym relationships between concepts by using a large unstructured text corpus. The author proposed a set of patterns which do not need any pre-encoded information to extract hypernym relationship from a plain text. These patterns were proposed to automatically extract this relationship only between nouns or noun phrases. Initially Hearst identified a few patterns (patterns 1, 2 and 3 from Figure 2.9) manually. An algorithm (also called a *boot-strapping algorithm*) is proposed to extract some new patterns from a large corpus using known hyponym-hypernym pairs. Then the corpus is searched for the context in which these pairs appear close to one another. The most frequent patterns in these contexts are hypothesized as the new patterns for hypernymy. These new patterns are used to fetch new hypernym-hyponym pairs from the text corpus. The patterns found by the boot-strapping algorithm also contain the patterns which the author identified manually. Figure 2.9 represent these lexico-syntactic patterns presented in this paper. These patterns are called **Hearst Patterns**.



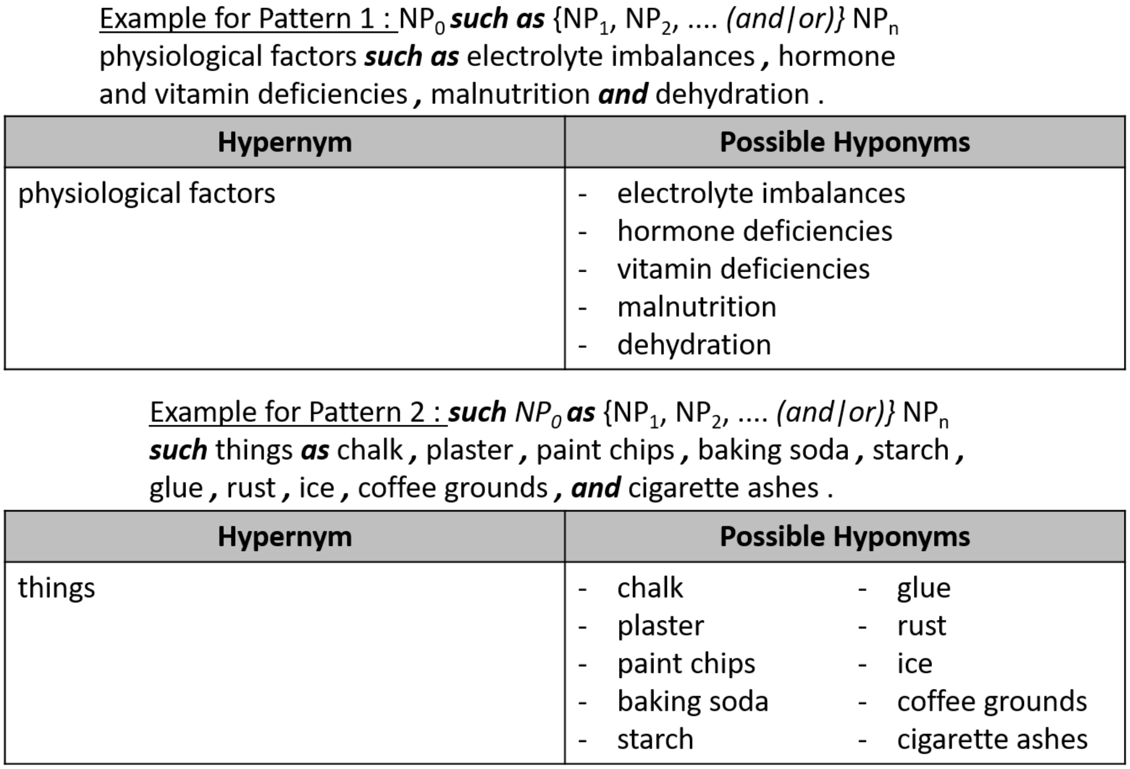
Pattern 1: NP_0 ***such as*** {NP_1, NP_2, ... *(and|or)*} NP_n
Pattern 2: ***such*** *NP_0* ***as*** {NP_1, NP_2, ... *(and|or)*} NP_n
Pattern 3: NP_1 {, NP_2, NP_3, ...}{,} ***or other*** NP_0
Pattern 4: NP_1 {, NP_2, NP_3, ...}{,} **and *other*** NP_0
Pattern 5: NP_0 {,} ***including*** {NP_1, NP_2, ... *(and|or)*} NP_n
Pattern 6: NP_0 {,} ***especially*** {NP_1, NP_2, ... *(and|or)*} NP_n

NP : **Noun Phrase**
(and|or) : Patterns matching ***and*** or ***or***
{} : pattern which is **optional**
NP_0 : Noun Phrase with **Hypernym**
NP_i : Noun Phrase with **Hyponym** where i = 1,2,3,...,n

Figure 2.9: Proposed Hearst Patterns

Since a hypernym-hyponym relationship is also called an ***ISA relationship***, we add another pattern "*hyponym* ***is (a | an | the)*** *hypernym*" to these 6 patterns. In Figure 2.9, a **Noun Phrase** represents a group of words which act as a subject or object in a given sentence. A noun phrase can have one or more headwords with a noun part-of-speech tag. For example, in the given sentence "The sun rises in the east and sets in the west", the noun phrases are *the sun*, *the east*, *the west*. In example, "The moon and the stars shine in the night.", the noun phrases are *the moon*, *the stars*, *the night* and *the moon and the stars*. In the noun phrase *the moon and the stars*, the nouns head words are *moon, stars*.

Figures 2.10, 2.11, 2.12 and 2.13 show a few example phrases from the UMBC corpus 2.4.2 where Hearst Patterns could be applied to identify potential hypernym-hyponym(s) pairs.



Example for Pattern 1 : NP_0 ***such as*** {NP_1, NP_2, ... *(and|or)}* NP_n
physiological factors ***such as*** electrolyte imbalances **,** hormone and vitamin deficiencies **,** malnutrition ***and*** dehydration .

Hypernym	Possible Hyponyms
physiological factors	- electrolyte imbalances - hormone deficiencies - vitamin deficiencies - malnutrition - dehydration

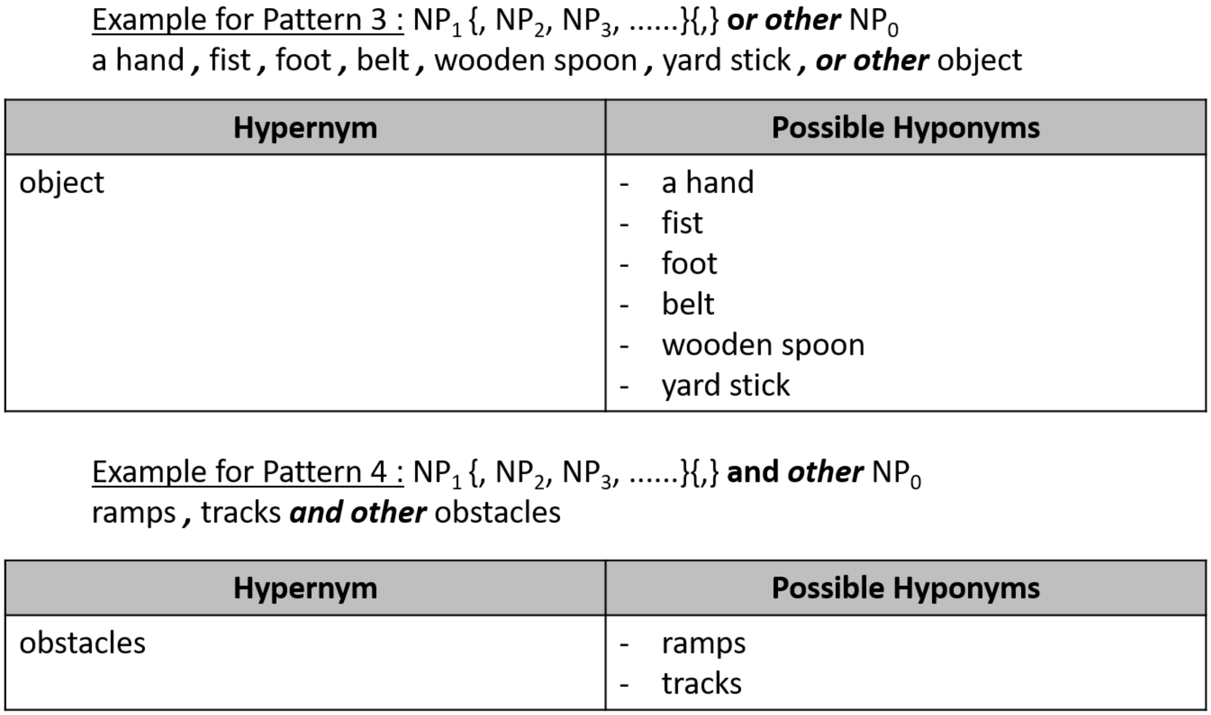
Example for Pattern 2 : ***such*** *NP_0* ***as*** {NP_1, NP_2, ... *(and|or)}* NP_n
such things ***as*** chalk **,** plaster **,** paint chips **,** baking soda **,** starch **,** glue **,** rust **,** ice **,** coffee grounds **,** ***and*** cigarette ashes .

Hypernym	Possible Hyponyms	
things	- chalk - plaster - paint chips - baking soda - starch	- glue - rust - ice - coffee grounds - cigarette ashes

Figure 2.10: Examples for Proposed Hearst Patterns 1 and 2

2.5.4 word2vec

All the words from a document can be represented in vector space in such a way that the semantic information from the document is also retained in the vector space. Vector models help to group words based on their co-occurrence frequencies from the given corpus. Simple linear transformations could be applied over these word vectors [Mikolov, Chen, Corrado, and Dean 2013]. For example, a word analogy problem could be applied over the word vectors built over a text corpus. *Word Analogy* is a problem in which the relationship between two given concepts could be used to fetch a concept which satisfies this relationship with another word. For example, the relationship between the words *Delhi* and *Capital* could be used to fetch a concept which resembles the same relationship (is-a) with *Rupee*. This problem is



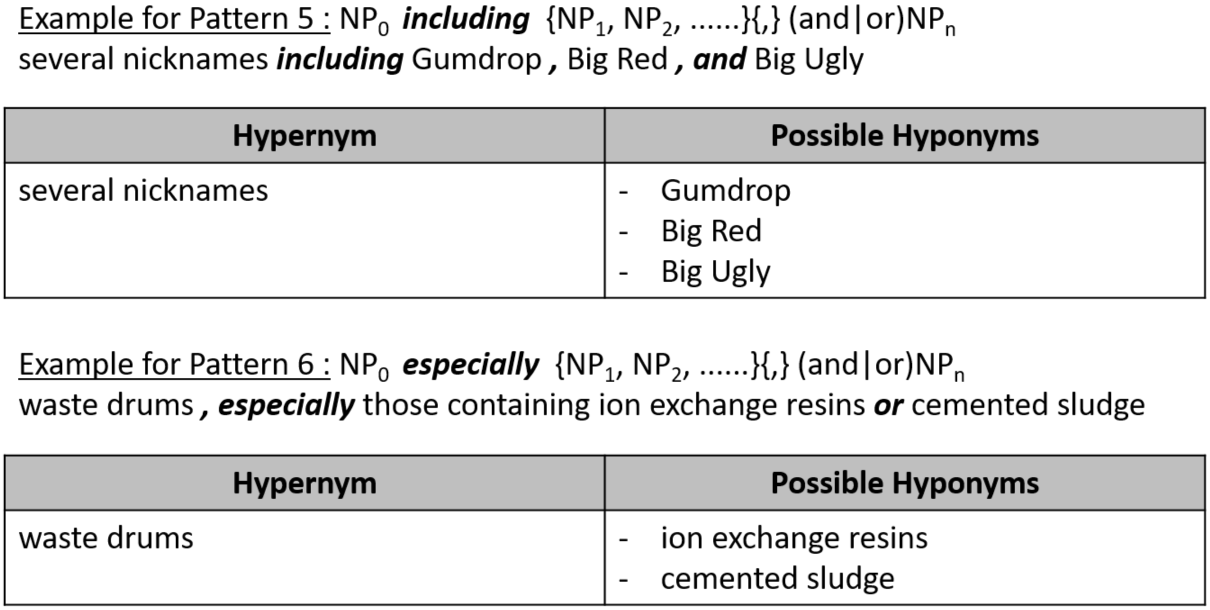
Example for Pattern 3 : NP$_1$ {, NP$_2$, NP$_3$, ...}{,} ***or other*** NP$_0$
a hand **,** fist **,** foot **,** belt **,** wooden spoon **,** yard stick ***, or other*** object

Hypernym	Possible Hyponyms
object	- a hand - fist - foot - belt - wooden spoon - yard stick

Example for Pattern 4 : NP$_1$ {, NP$_2$, NP$_3$, ...}{,} **and *other*** NP$_0$
ramps **,** tracks ***and other*** obstacles

Hypernym	Possible Hyponyms
obstacles	- ramps - tracks

Figure 2.11: Examples for Proposed Hearst Patterns 3 and 4



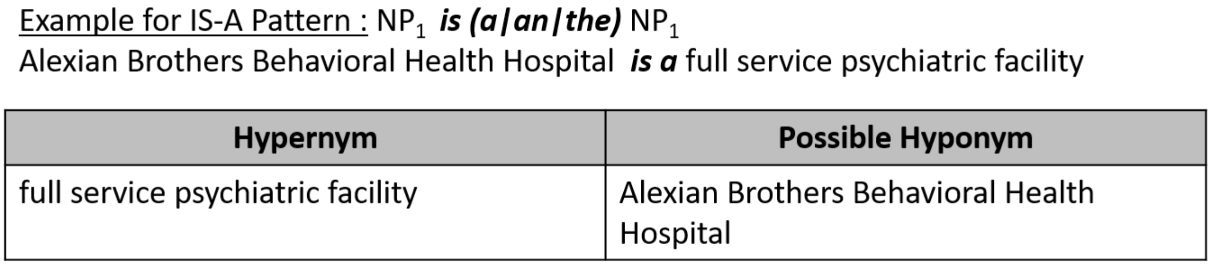
Example for Pattern 5 : NP$_0$ ***including*** {NP$_1$, NP$_2$, ...}{,} (and|or)NP$_n$
several nicknames ***including*** Gumdrop **,** Big Red ***, and*** Big Ugly

Hypernym	Possible Hyponyms
several nicknames	- Gumdrop - Big Red - Big Ugly

Example for Pattern 6 : NP$_0$ ***especially*** {NP$_1$, NP$_2$, ...}{,} (and|or)NP$_n$
waste drums ***, especially*** those containing ion exchange resins ***or*** cemented sludge

Hypernym	Possible Hyponyms
waste drums	- ion exchange resins - cemented sludge

Figure 2.12: Examples for Proposed Hearst Patterns 5 and 6



Example for IS-A Pattern : NP$_1$ ***is (a|an|the)*** NP$_1$
Alexian Brothers Behavioral Health Hospital ***is a*** full service psychiatric facility

Hypernym	Possible Hyponym
full service psychiatric facility	Alexian Brothers Behavioral Health Hospital

Figure 2.13: Example for Proposed Hearst Pattern IS-A

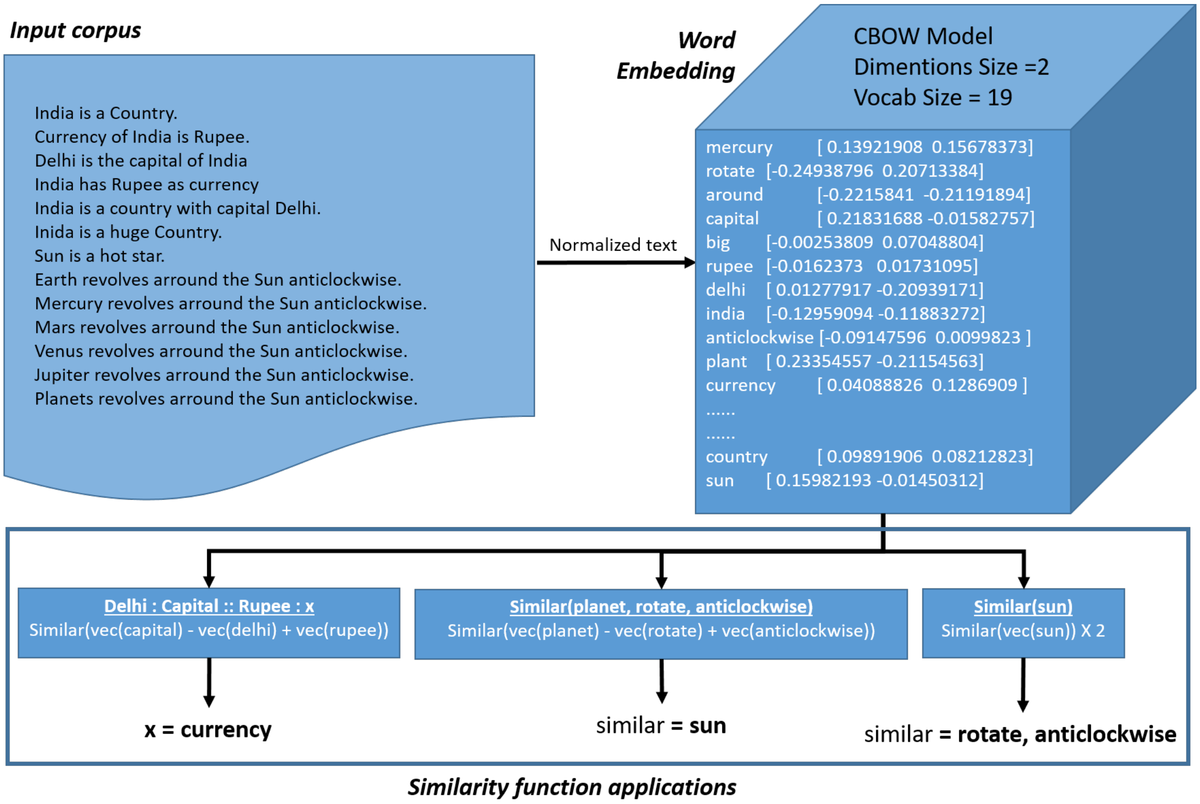
be represented as ***A:B::C:x***, where *A*, *B*, *C* are *Delhi*, *Capital*, *Rupee* and *x* is the *unknown concept*. After applying the word analogy *Delhi:Capital::Rupee:x*, the concept which fits in place of x is *Currency* which is the hypernym for *rupee*. If *M* is a vector matrix built over an English text corpus, then *vec(concept)* represents the vector value for the *concept* from this corpus. Figure 2.4 show the vector space representation of the word analogy *A:B::C:x*.

$$vec(x) = vec(B) - vec(A) + vec(C) \tag{2.4}$$

The initial transformation *vec(B) - vec(A)* fetches a vector that we hypothesize may represent the relationship between the words *A* and *B*. This relationship could be hypernymy, synonymy, meronymy or any other semantic or lexical relationship. Then this relation vector is added to the concept *C*ś vector to fetch the vector *vec(x)*. The word whose vector value is closest to the *vec(x)* is our result word for the word analogy *A:B::C:x*. Let us consider a word analogy problem *apple:fruit::onion:x*. The relationship between the words *fruit* and *apple* is **hypernymy**. The concept which satisfies a hypernym relationship with the word *onion* is *vegetable*. Hence the desired solution is *apple:fruit::onion:vegetable*. Likewise, the *Word Similarity* problem could also be applied to the word vectors. Here, the *Word Similarity* problem is different that the word similarity problem explained in the section 2.3. When given a set of words, a word embedding could be used to fetch a concept which has the best co-occurrence frequency and is therefore hypothesized to be similar to the given words. Similarly, when given a concept, a vector model could be used to fetch a list of high frequency context words.

In the paper *Distributed Representations of Words and Phrases and their Compositionality* [Mikolov, Sutskever, Chen, Corrado, and Dean 2013], the authors propose various algorithms which can be used to build word-embedding matrices over huge text corpora with billions of words. The model which implements this algorithm is called *Word2Vec*[7]. This model is used to build two different word embedding models - *Continuous Bag of Words (CBOW)* model and *Continuous Skip Gram (skip-gram)* model. The CBOW model learnt from the corpus (shown in Figure 2.14) could be used to identify a word which has the highest probability to co-occur with the given set of input words. The order of the words in a context does not matter for the CBOW model. For example, if the set of words given are "planet, rotate, anticlockwise", then the predicted context word in the best case should be *sun*. Unlike CBOW model, the skip-gram model learnt from the same corpus could be used identify the context words given an input word. For example, if the given word is *anticlockwise*, then the context words predicted could be "sun, rotate". Figure 2.14 shows a sample corpus, a CBOW model learnt on s sample corpus and examples for word-analogy and word similarity problems.

[7] https://code.google.com/archive/p/word2vec/



India is a Country.
Currency of India is Rupee.
Delhi is the capital of India
India has Rupee as currency
India is a country with capital Delhi.
Inida is a huge Country.
Sun is a hot star.
Earth revolves arround the Sun anticlockwise.
Mercury revolves arround the Sun anticlockwise.
Mars revolves arround the Sun anticlockwise.
Venus revolves arround the Sun anticlockwise.
Jupiter revolves arround the Sun anticlockwise.
Planets revolves arround the Sun anticlockwise.

mercury	[0.13921908 0.15678373]
rotate	[-0.24938796 0.20713384]
around	[-0.2215841 -0.21191894]
capital	[0.21831688 -0.01582757]
big	[-0.00253809 0.07048804]
rupee	[-0.0162373 0.01731095]
delhi	[0.01277917 -0.20939171]
india	[-0.12959094 -0.11883272]
anticlockwise	[-0.09147596 0.0099823]
plant	[0.23354557 -0.21154563]
currency	[0.04088826 0.1286909]
......	
......	
country	[0.09891906 0.08212823]
sun	[0.15982193 -0.01450312]

Delhi : Capital :: Rupee : x
Similar(vec(capital) - vec(delhi) + vec(rupee))

x = currency

Similar(planet, rotate, anticlockwise)
Similar(vec(planet) - vec(rotate) + vec(anticlockwise))

similar **= sun**

Similar(sun)
Similar(vec(sun)) X 2

similar **= rotate, anticlockwise**

Figure 2.14: Proposed Hearst Patterns

3 Implementation

In this chapter we would present a detailed description of the modules developed for the two SemEval tasks (Chapter 2 Section 2.4.4). The following is a brief overview of various phases of our modules. These phases are explained in detail in later parts of this chapter.

1. **Pre-processing**: We pre-process the various data sources (Chapter 2 Section 2.4) used by our systems. This helps in reducing the execution time of the Discovering Hypernym phase. We create a normalized corpus from the UMBC Corpus, identify hypernym - hyponym pairs from UMBC Corpus using Hearst Patterns, create word-embeddings over the UMBC corpus, and create word-embeddings over the definitions of the WordNet lemmas. Once these are created, they can be reused with any input data (except embeddings over WordNet Definitions). For more information about the UMBC Corpus and the WordNet, please refer the background Chapter (Chapter 2) Section 2.4

2. **Discovering Hypernyms**: We extract hypernym(s) for any given input term by using the pre-processed corpora. There are two types of inputs given to our systems:

   (a) an input term without a definition

   (b) an input term with a definition

   Depending on the combination of the pre-processed corpus and the type of input, the techniques used to extract hypernym(s) change. We use *co-occurrence frequencies* over the UMBC Normalized Corpus and UMBC Hearst Pattern Corpora. *Similarity Distance* is used from the word-embeddings learnt from the UMBC Corpus and WordNet Definition Corpora. *Pure Hearst Pattern matching* is used over the input term's definition to identify one potential hypernym. Each technique is considered as one sub-system.

3. **Re-assign Sense**: This module is applicable only for the Semantic Taxonomy Enrichment task (SemEval 2016 Task 14, Chapter 2, Section 2.4.4). In the discovery phase, we assign a default sense "*#1*" to all the hypernyms identified by our sub-systems. In this phase we ignore this default sense and try to re-assign a more precise sense to these identified hypernyms by using the algorithms - Lesk[1] and Extended Gloss Overlaps[2].

4. **Merging the Results**: We merge the results obtained from various sub-systems of the Discovering Hypernyms phase. Based on the type of task (Chapter 2 Section 2.4.4), we choose one of the following merge techniques- *Select One* or *Merge All* - to obtain the final results.

3.1 Pre-processing the Available Resources

The very first challenge we faced while working with the SemEval 2018 Task 9 Hypernym Discovery is the *size* of the UMBC data corpus (Chapter 2 Section 2.4.2). Since most of our work involves nouns, we had the potential to reduce the size of this **28.3GB** corpus by eliminating all other part-of-speech tags and modify it as per the

[1] https://www.nltk.org/_modules/nltk/wsd.html

[2] https://github.com/m1ha1f/disambiguation

requirements of this task. As such we apply various pre-processing techniques on the original UMBC Corpus to obtain the following task specific reduced corpora:

1. **UMBC Normalized Corpus**

2. UMBC One-to-One Hearst Pattern Corpus (we call this corpus **UMBC IS-A Hearst Corpus**)

3. UMBC Many-to-One Hearst Patterns Corpus (we call this corpus **UMBC Other Hearst Corpus**)

4. UMBC Word-Embeddings built on Normalized Corpus (we call this corpus **UMBC Word-Embedding**)

While learning the word-embedding matrix for the UMBC corpus, we have realized that a similar word-embedding matrix could be learnt from the WordNet data as well. So we create a word-embedding matrices over the definitions of all the noun and verb synsets obtained from WordNet. We call these matrices **WordNet (Noun—Verb) Definition Embeddings**.

3.1.1 UMBC Normalized Corpus

From the Hypernym Discovery task description (Chapter 2 Section 2.4.4), it was clear that both the input terms and the output hypernyms are noun phrases[3]. As a result, we decided to reduce the size of the UMBC corpus even before processing it against the input data to fetch candidate hypernyms. This would reduce the execution time of the Discovering Hypernym phase. Figure 3.2 represents the implementation of this normalization and Figure 3.3 shows how normalization reduces the size of a simple corpus.

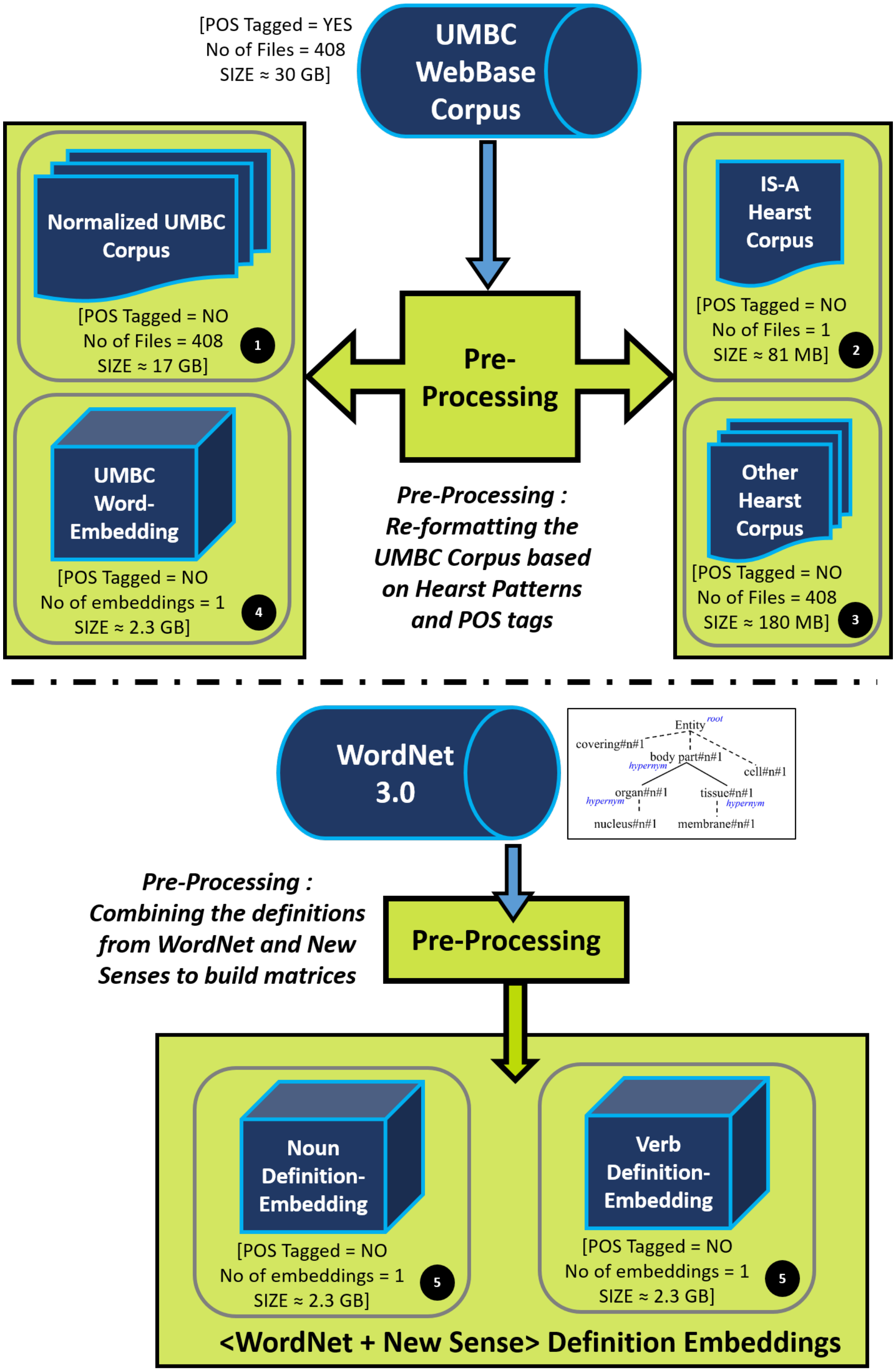


Figure 3.1: Corpora extracted from UMBC WebBase Corpus and WordNet 3.0

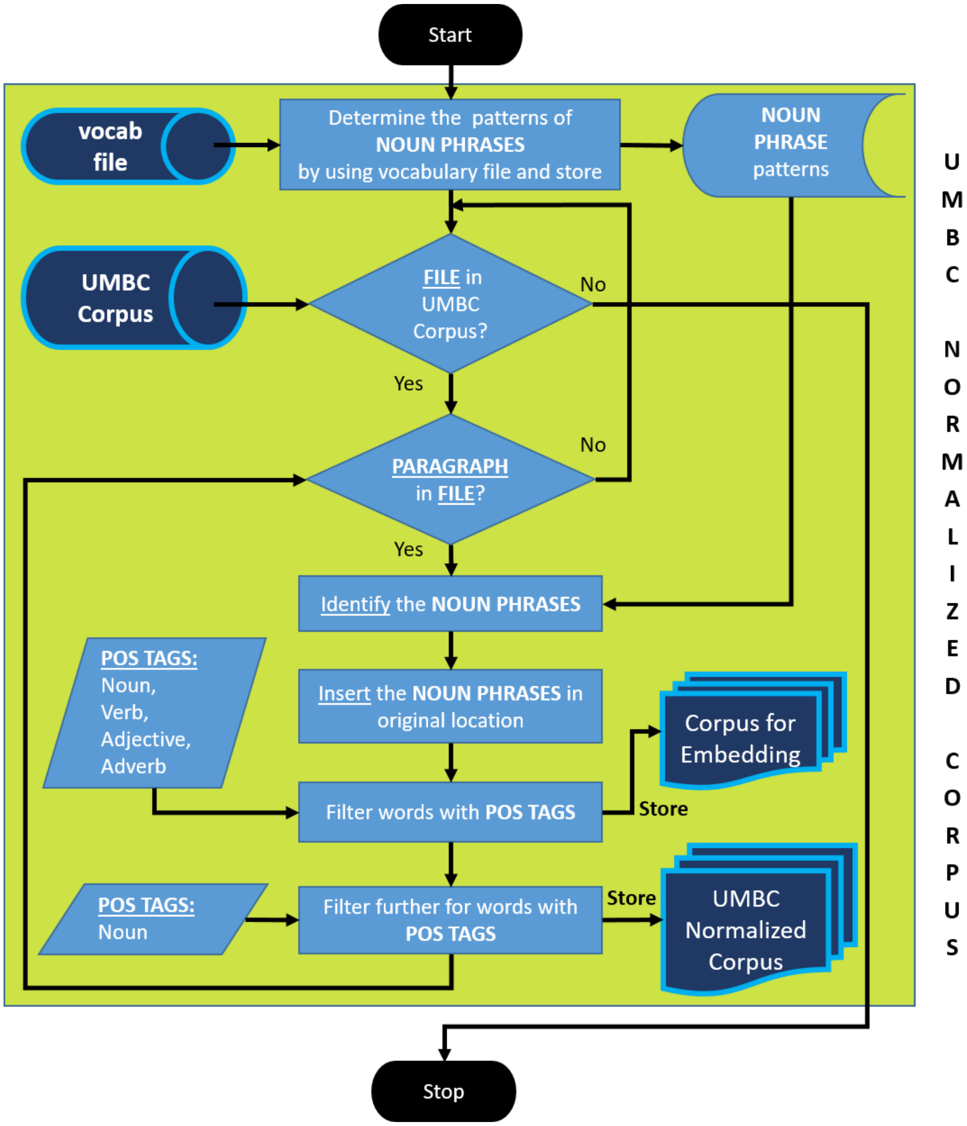
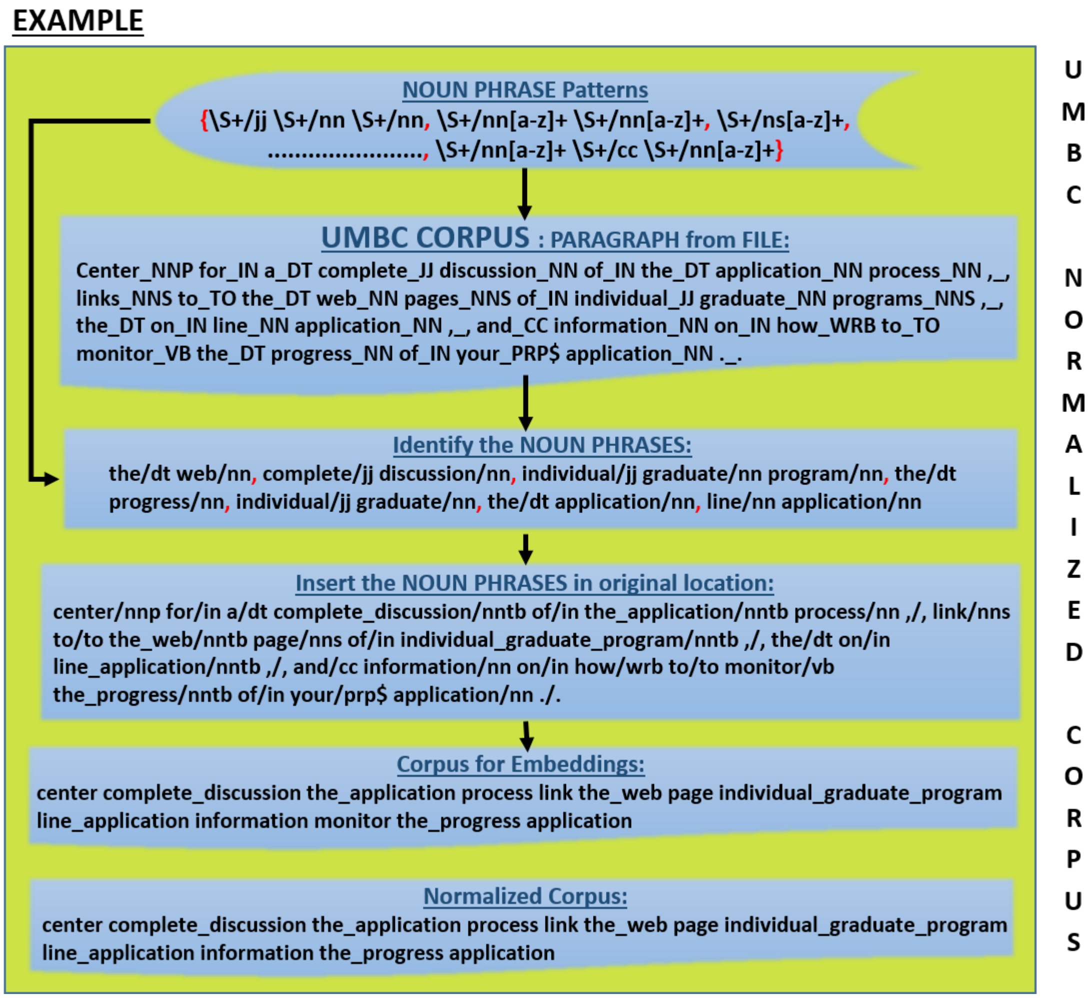

Start
vocab file
Determine the patterns of NOUN PHRASES by using vocabulary file and store
NOUN PHRASE patterns
UMBC Corpus
FILE in UMBC Corpus?
No
Yes
PARAGRAPH in FILE?
No
Yes
Identify the NOUN PHRASES
POS TAGS: Noun, Verb, Adjective, Adverb
Insert the NOUN PHRASES in original location
Corpus for Embedding
Filter words with POS TAGS
Store
POS TAGS: Noun
Filter further for words with POS TAGS
Store
UMBC Normalized Corpus
UMBC NORMALIZED CORPUS
Stop

Figure 3.2: The flow chart for creating UMBC Normalized Corpus

Before processing the UMBC WebBase Corpus, we process the *vocabulary* file provided with the task. This vocabulary file lists all the candidate result hypernyms for this task, such that the noun phrase patterns which exist in this file would be the bi-gram and tri-gram patterns of interest. We used Natural Language Toolkit (NLTK) part-of-speech tagger to tag all the vocabulary terms and then we fetch those

[3] The part-of-speech tag of the phrase's headword is noun



EXAMPLE

NOUN PHRASE Patterns

{\S+/jj \S+/nn \S+/nn, \S+/nn[a-z]+ \S+/nn[a-z]+, \S+/ns[a-z]+, ..., \S+/nn[a-z]+ \S+/cc \S+/nn[a-z]+}

UMBC CORPUS : PARAGRAPH from FILE:

Center_NNP for_IN a_DT complete_JJ discussion_NN of_IN the_DT application_NN process_NN ,_, links_NNS to_TO the_DT web_NN pages_NNS of_IN individual_JJ graduate_NN programs_NNS ,_, the_DT on_IN line_NN application_NN ,_, and_CC information_NN on_IN how_WRB to_TO monitor_VB the_DT progress_NN of_IN your_PRP$ application_NN ._.

Identify the NOUN PHRASES:

the/dt web/nn, complete/jj discussion/nn, individual/jj graduate/nn program/nn, the/dt progress/nn, individual/jj graduate/nn, the/dt application/nn, line/nn application/nn

Insert the NOUN PHRASES in original location:

center/nnp for/in a/dt complete_discussion/nntb of/in the_application/nntb process/nn ,/, link/nns to/to the_web/nntb page/nns of/in individual_graduate_program/nntb ,/, the/dt on/in line_application/nntb ,/, and/cc information/nn on/in how/wrb to/to monitor/vb the_progress/nntb of/in your/prp$ application/nn ./.

Corpus for Embeddings:

center complete_discussion the_application process link the_web page individual_graduate_program line_application information monitor the_progress application

Normalized Corpus:

center complete_discussion the_application process link the_web page individual_graduate_program line_application information the_progress application

Figure 3.3: An example for creating a sample UMBC Normalized Corpus

POS tag patterns which appear more than 100 times in this file. These patterns are then generalized to all nouns and are used to refine the UMBC Corpus as shown in the *figure* 3.2 (Noun Phrase patterns). All these bi-gram and tri-gram patterns are identified from the original UMBC paragraph. These identified bi-gram or tri-gram terms are converted to a uni-gram term by replacing the inter word spaces with underscores. These uni-grams are inserted after their original bi-gram or tri-gram locations and a new noun POS tag "*nntb*" is assigned to them. This POS tag helps to filter this modified paragraph in the next steps. All the words with POS tags other than noun, verb, adjective and adverb are removed from this new text and the resulting paragraph is written into the *Word-Embedding Corpus* files. This text is further refined to retain only the noun POS tag words. This final text is written into the *UMBC Normalized Corpus* files.

Figure 3.3 shows how a paragraph from the UMBC WebBase Corpus is refined to form a Word-Embedding Corpus paragraph and the final Normalized Corpus paragraph.

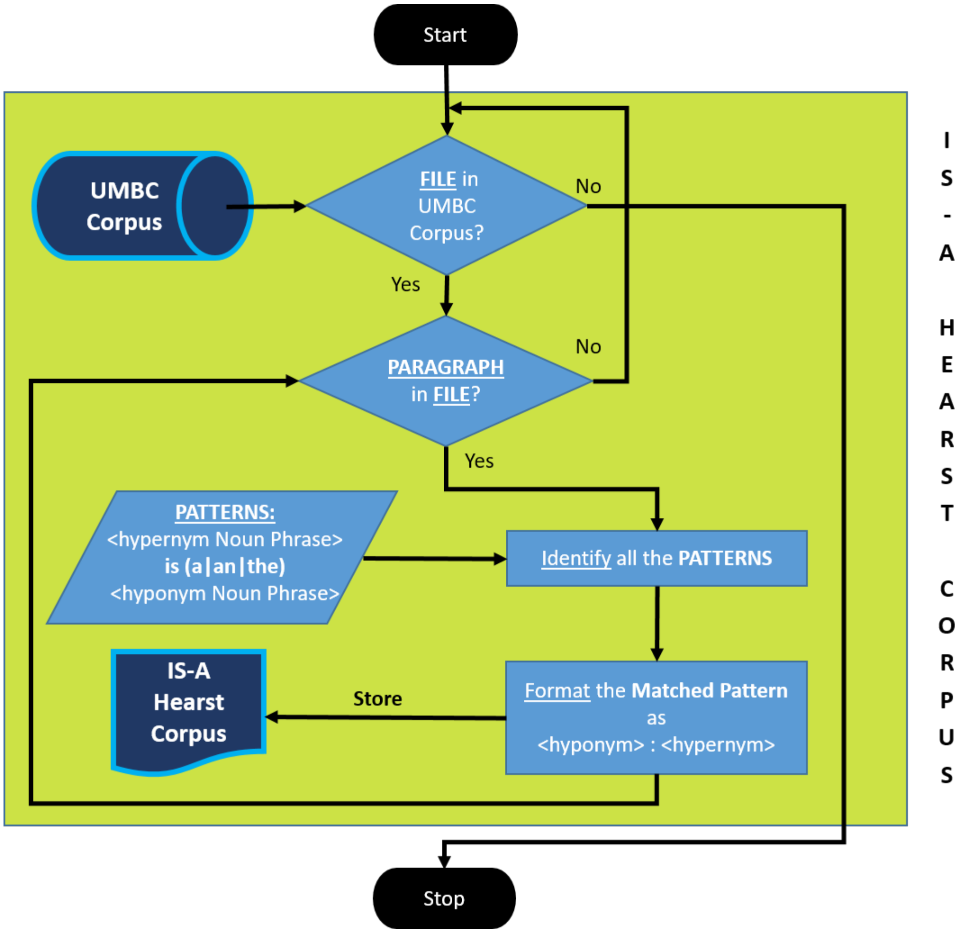
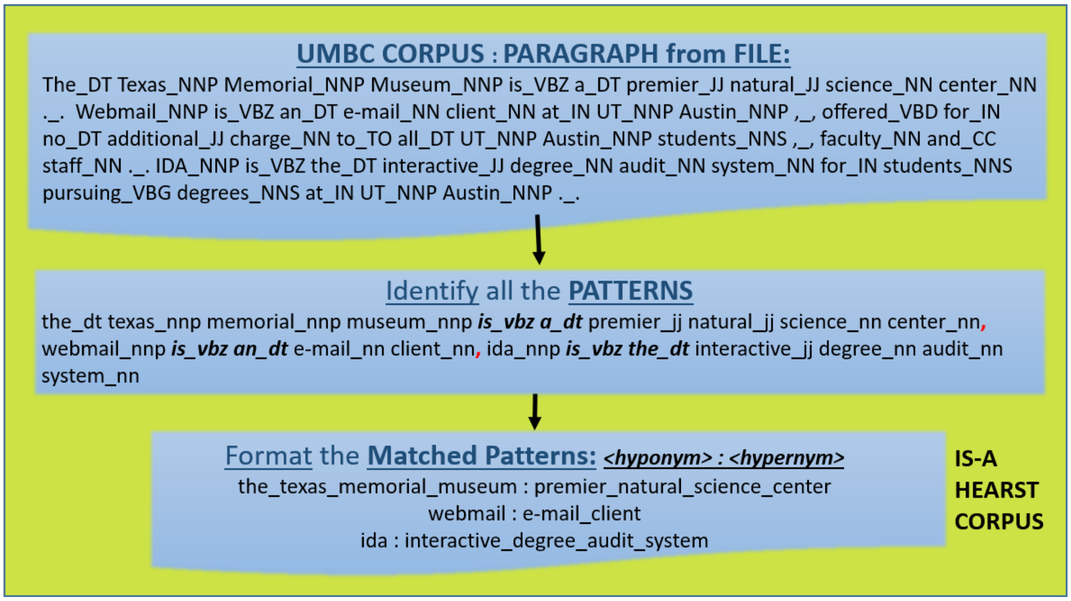


Figure 3.4: The flow chart for creating IS-A (One-to-One) Hearst Pattern Corpus on UMBC WebBase Corpus

EXAMPLE



UMBC CORPUS : PARAGRAPH from FILE:

The_DT Texas_NNP Memorial_NNP Museum_NNP is_VBZ a_DT premier_JJ natural_JJ science_NN center_NN ._. Webmail_NNP is_VBZ an_DT e-mail_NN client_NN at_IN UT_NNP Austin_NNP ,_, offered_VBD for_IN no_DT additional_JJ charge_NN to_TO all_DT UT_NNP Austin_NNP students_NNS ,_, faculty_NN and_CC staff_NN ._. IDA_NNP is_VBZ the_DT interactive_JJ degree_NN audit_NN system_NN for_IN students_NNS pursuing_VBG degrees_NNS at_IN UT_NNP Austin_NNP ._.

Identify all the PATTERNS

the_dt texas_nnp memorial_nnp museum_nnp ***is_vbz a_dt*** premier_jj natural_jj science_nn center_nn, webmail_nnp ***is_vbz an_dt*** e-mail_nn client_nn, ida_nnp ***is_vbz the_dt*** interactive_jj degree_nn audit_nn system_nn

Format the Matched Patterns: <hyponym> : <hypernym>

the_texas_memorial_museum : premier_natural_science_center
webmail : e-mail_client
ida : interactive_degree_audit_system

IS-A HEARST CORPUS

Figure 3.5: An example of identifying IS-A Hearst Patterns from UMBC WebBase Corpus

3.1.2 UMBC IS-A Hearst Corpus

Hearst Patterns are used to extract hypernymy from a large text corpus (Chapter 2 Section 2.4.2). For the UMBC IS-A Hearst Corpus we choose only one specific Hearst Pattern - "*hyponym* ***is (a*** | ***an*** | ***the)*** *hypernym*". This pattern is applied to the UMBC Corpus to extract all (*hyponym*, *hypernym*) pairs. We call this corpus *UMBC One-to-One Hearst Pattern Corpus* because this pattern identifies only one term as a potential *hyponym* for the *hypernym* term in context. For example, if we apply this pattern to the text "*Earth is a blue planet with 70% water and only 30% land*", we obtain *blue planet* as a hypernym with *earth* as its only hyponym. The other Hearst Patterns used in our research identify one or more hyponyms for the hypernym term in context. This is one of the reasons why we create a separate corpus for this pattern. The other reason to create a separate corpus is the accuracy of the results. The hypernyms obtained from this corpus are far more accurate than the hypernyms obtained from the corpus built over all the other Hearst Patterns (please refer Table 4.6 for the score values). Figure 3.4 represents the flow chart which is used to extract the hyponym-hypernym pairs from the UMBC corpus.

Figure 3.5 shows how our algorithm identifies a few *hyponym : hypernym* pairs from UMBC WebBase Corpus sample.

3.1.3 UMBC Other Hearst Corpus

We used one Hearst Pattern to extract hypernym from the UMBC Corpus in the previous sub-section. In this section, we use all other Hearst Patterns (shown in the flow chart 3.6) to identify the hypernyms and their respective hyponyms from the UMBC corpus. All the Hearst Patterns used in our research could be obtained from the Background Chapter (Chapter 2). We call this corpus *UMBC Many-to-One Hearst Pattern Corpus* because this pattern identifies one or more noun phrases as potential *hyponyms* for the *hypernym* noun phrase in context. For example, if we apply one of the patterns to the text "*electronic devices such as phones, laptops and tablets*", we obtain *(electronic devices : phones, laptops, tablets* as the hypernym-hyponym(s) pair. Here for one hypernym - *electronic devices*, our system identified three hyponyms - *phones, laptops, tablets*. Figure 3.6 represents this process which extracts the hypernym-hyponym_list pairs from the UMBC corpus.

Figure 3.7 shows how our algorithm identified a few *hypernym : hyponym_list* pairs from sample UMBC WebBase Corpus.

3.1.4 UMBC Word-Embedding

By creating the *UMBC Normalized Corpus*, we reduced the size of the original corpus by 50% approximately. This is achieved by using the part-of-speech(POS)

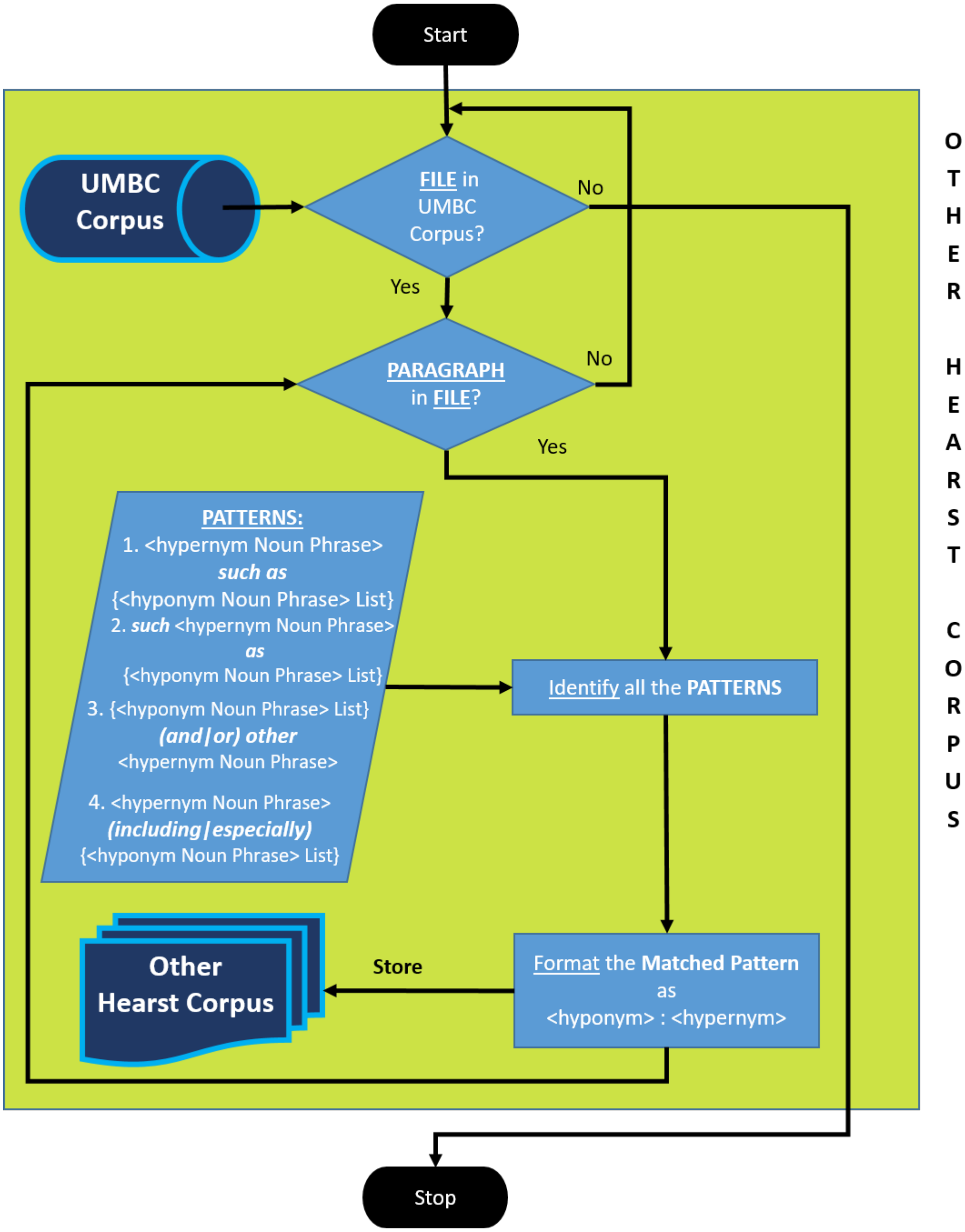
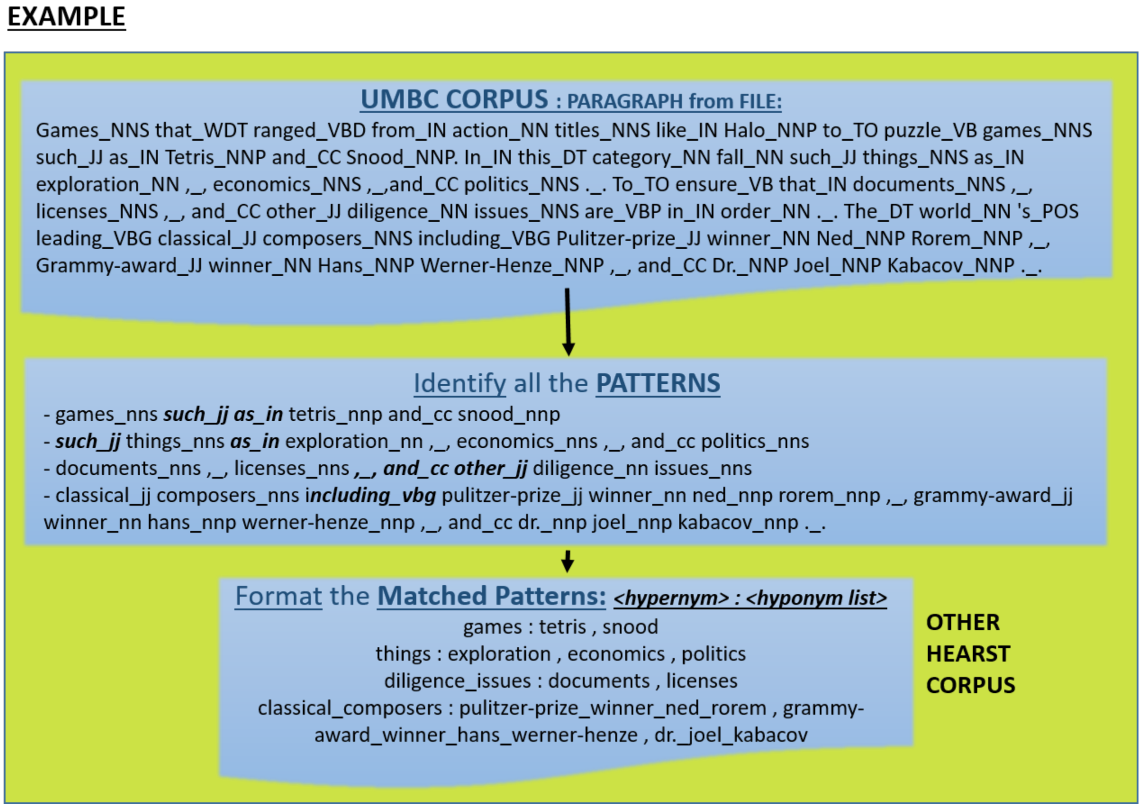

Start
UMBC Corpus
FILE in UMBC Corpus?
No
Yes
PARAGRAPH in FILE?
No
Yes
PATTERNS:
1. <hypernym Noun Phrase> *such as* {<hyponym Noun Phrase> List}
2. *such* <hypernym Noun Phrase> *as* {<hyponym Noun Phrase> List}
3. {<hyponym Noun Phrase> List} *(and|or) other* <hypernym Noun Phrase>
4. <hypernym Noun Phrase> *(including|especially)* {<hyponym Noun Phrase> List}
Identify all the PATTERNS
Format the Matched Pattern as <hyponym> : <hypernym>
Store
Other Hearst Corpus
OTHER HEARST CORPUS
Stop

Figure 3.6: The flow chart for creating Other (Many-to-One) Hearst Pattern Corpus on UMBC WebBase Corpus

tags of the original corpus and input words to retain all the uni-gram, bi-gram and tri-gram noun phrases. But to retain more information from the original corpus, we



EXAMPLE

UMBC CORPUS : PARAGRAPH from FILE:

Games_NNS that_WDT ranged_VBD from_IN action_NN titles_NNS like_IN Halo_NNP to_TO puzzle_VB games_NNS such_JJ as_IN Tetris_NNP and_CC Snood_NNP. In_IN this_DT category_NN fall_NN such_JJ things_NNS as_IN exploration_NN ,_, economics_NNS ,_,and_CC politics_NNS ._. To_TO ensure_VB that_IN documents_NNS ,_, licenses_NNS ,_, and_CC other_JJ diligence_NN issues_NNS are_VBP in_IN order_NN ._. The_DT world_NN 's_POS leading_VBG classical_JJ composers_NNS including_VBG Pulitzer-prize_JJ winner_NN Ned_NNP Rorem_NNP ,_, Grammy-award_JJ winner_NN Hans_NNP Werner-Henze_NNP ,_, and_CC Dr._NNP Joel_NNP Kabacov_NNP ._.

Identify all the PATTERNS

- games_nns ***such_jj as_in*** tetris_nnp and_cc snood_nnp
- ***such_jj*** things_nns ***as_in*** exploration_nn ,_, economics_nns ,_, and_cc politics_nns
- documents_nns ,_, licenses_nns ***,_, and_cc other_jj*** diligence_nn issues_nns
- classical_jj composers_nns i***ncluding_vbg*** pulitzer-prize_jj winner_nn ned_nnp rorem_nnp ,_, grammy-award_jj winner_nn hans_nnp werner-henze_nnp ,_, and_cc dr._nnp joel_nnp kabacov_nnp ._.

Format the **Matched Patterns:** ***<hypernym> : <hyponym list>***

games : tetris , snood
things : exploration , economics , politics
diligence_issues : documents , licenses
classical_composers : pulitzer-prize_winner_ned_rorem , grammy-award_winner_hans_werner-henze , dr._joel_kabacov

OTHER HEARST CORPUS

Figure 3.7: An example of identifying Other (Many-to-One) Hearst Patterns from UMBC WebBase Corpus

wanted to include more words with other POS tags like verbs, adjectives and adverbs and create a new normalized corpus. The size of this normalized corpus is greater than the *UMBC Normalized Corpus* and hence the computation time to discover hypernyms from this corpus would also increase. In order to reduce the corpus size and computation time we learn a word embedding matrix from this new normalized corpus. This word embedding matrix is created using the following configurations:

1. **Model:** ***Continuous Bag of Words (CBOW)***. The vector values of all the words in context windows are computed based on the vector value of the word which is in the center of the context. For example, in the sentence "Planet Earth revolves around Sun", the vector values of the words *planet, earth, around, sun*

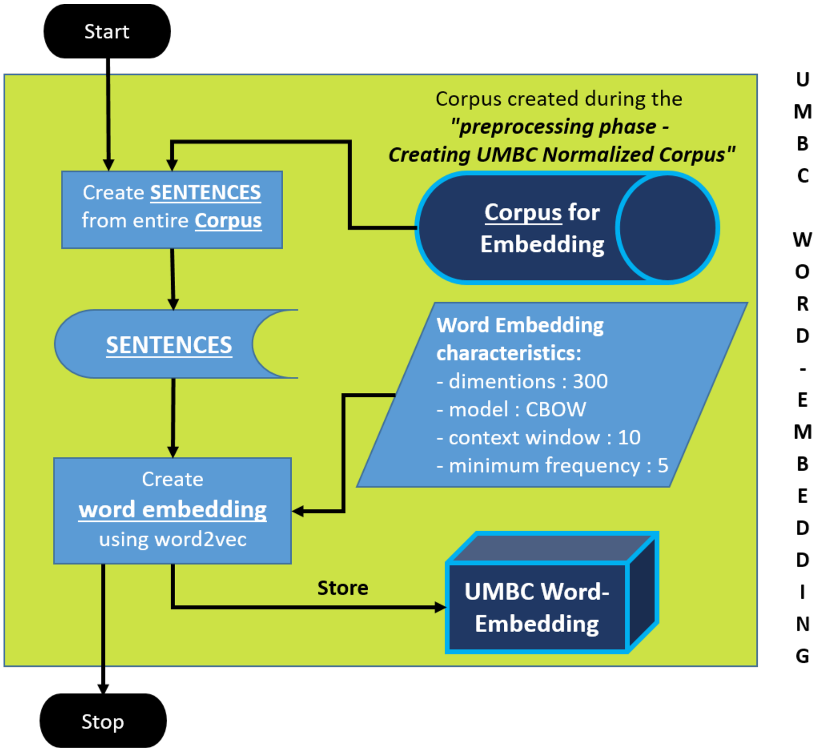


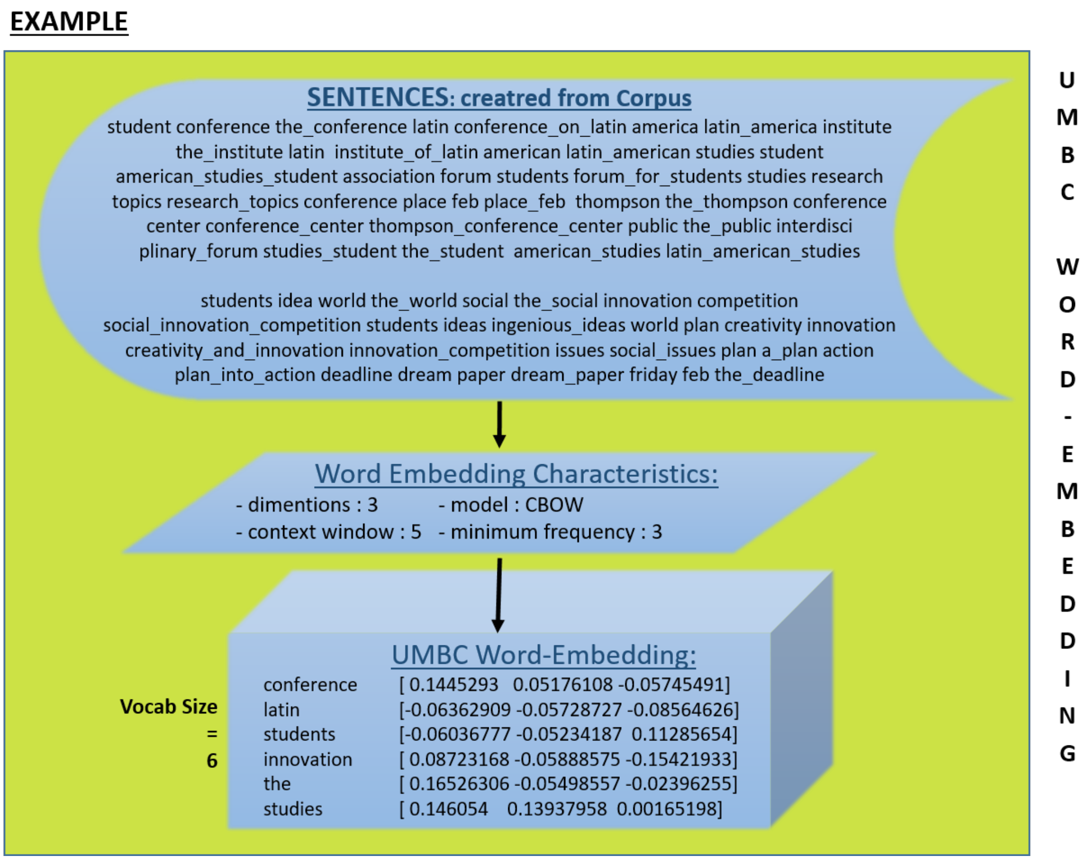
Figure 3.8: The flow chart for creating UMBC Word-Embedding matrix

are computed by using *revolves* vector value. So when a CBOW word embedding is queried for a word which appears in context of *earth, sun*, "revolves" would be retrieved.

2. **Window Size:** ***10***. The context window size for the embedding which calculates and re-assigns the vector values of all the words in context with respect to the center word. The number words surrounding the word of interest in a given context is 10 including the target word.

3. **Minimum Frequency Count:** ***5***. If the overall frequency of a term is less than this value, the vector for this term is deleted from the embedding.

4. **Dimension Size:** ***300***. This value represents the number of dimensions of the feature vector. If this value is low, then a model would lose more information


EXAMPLE
SENTENCES: creatred from Corpus
student conference the_conference latin conference_on_latin america latin_america institute
the_institute latin institute_of_latin american latin_american studies student
american_studies_student association forum students forum_for_students studies research
topics research_topics conference place feb place_feb thompson the_thompson conference
center conference_center thompson_conference_center public the_public interdisci
plinary_forum studies_student the_student american_studies latin_american_studies
students idea world the_world social the_social innovation competition
social_innovation_competition students ideas ingenious_ideas world plan creativity innovation
creativity_and_innovation innovation_competition issues social_issues plan a_plan action
plan_into_action deadline dream paper dream_paper friday feb the_deadline
Word Embedding Characteristics:
- dimentions : 3
- model : CBOW
- context window : 5
- minimum frequency : 3
UMBC Word-Embedding:
Vocab Size = 6
UMBC WORD-EMBEDDING

conference	[0.1445293 0.05176108 -0.05745491]
latin	[-0.06362909 -0.05728727 -0.08564626]
students	[-0.06036777 -0.05234187 0.11285654]
innovation	[0.08723168 -0.05888575 -0.15421933]
the	[0.16526306 -0.05498557 -0.02396255]
studies	[0.146054 0.13937958 0.00165198]

Figure 3.9: An example Word-Embedding matrix creation

about the original corpus. If this value is high, then this embedding would be a closer representation of the original corpus. We chose **300** because the Google News Vectors[4] is also trained with the same dimension. We chose the same value firstly because Google News Vectors are also trained on 3 billion words like the UMBC Corpus. And secondly, we wanted to compare the results obtained from various embeddings with the same input data. In order to do this, all these embeddings should have equal amount of information with respect to their training corpora.

Figure 3.8 shows the creation of *UMBC Word-Embedding* from the *Corpus for*

[4] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

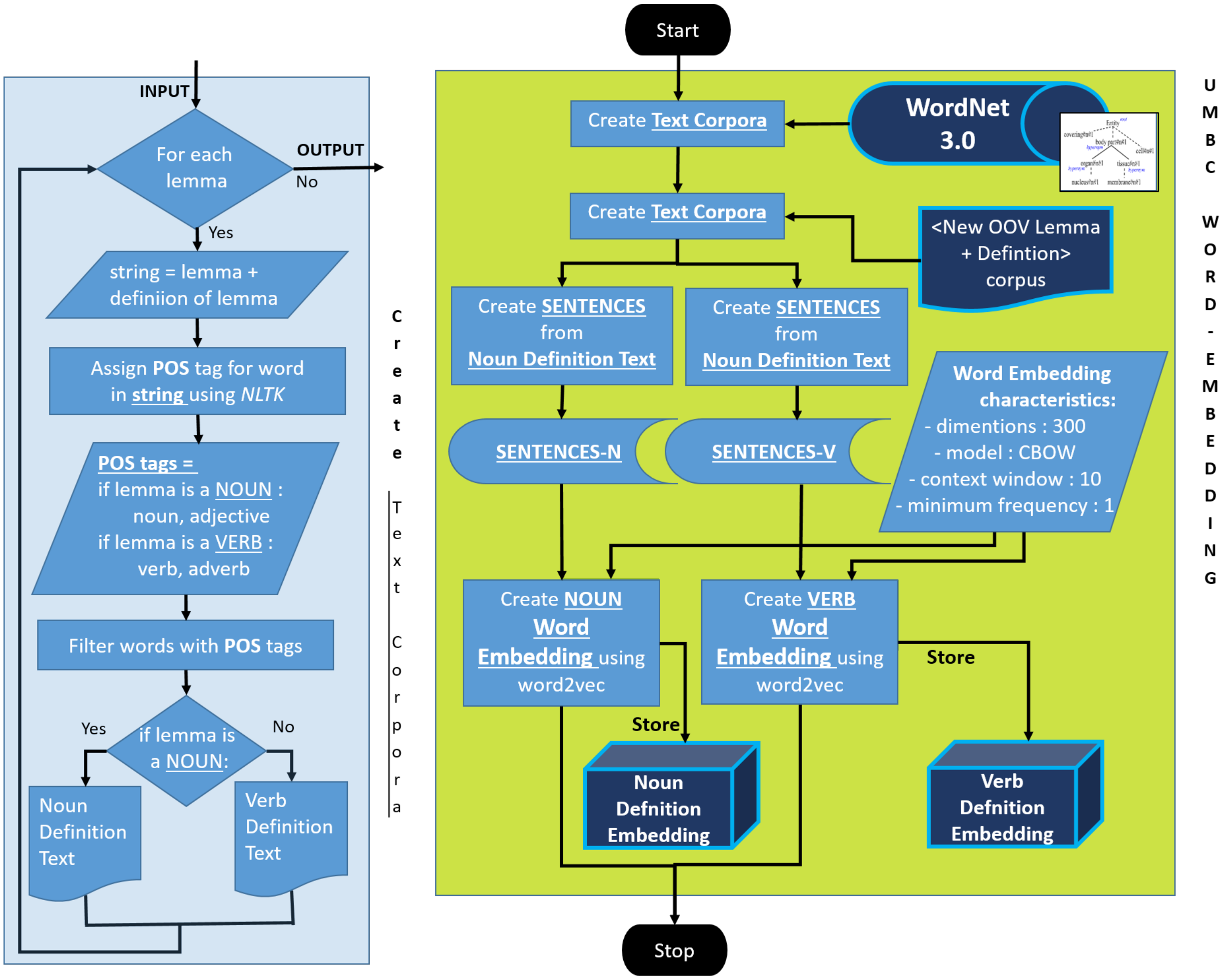


Figure 3.10: The flow chart for creating a word embedding matrix over WordNet Definitions Text Corpus. The module on the left shows how we created this Text Corpus from WordNet and the input definitions.

Embedding which is created while creating *UMBC Normalized Corpus Figure 3.2*. The size of this word embedding matrix is approximately *one-tenth* of the original UMBC Corpus size. Figure 3.9 shows a sample word embedding matrix created over a small text corpus. It is created by using *word2vec* [Kilgarriff and Rosenzweig 2000] with dimension, context window size and minimum frequency values as 3, 5, 3 respectively. We chose these values for the convenience of representation and understand-ability.

3.1.5 WordNet Definition Embeddings

Initially when we conducted some experiments using *Google News Vectors*, we realized that for some new out-of-vocabulary (OOV) lemmas from the training data, the hypernyms fetched from WordNet have a greater similarity score than the baseline results. For example, the Wu & Palmer Similarity Score for the new OOV lemma "*chain*" is *0.5882* when the Google Vectors predicted result is "*string#n#1*" and the human predicted result is "*measuring_instrument#n#1*". The average Wu & Palmer Similarity Score for Random baseline, First-word First-sense baseline and Default baseline are *0.2179*, *0.4763* and *0.2495* respectively. But the recall value for this experiment is only to 51% only (0.505 = fetched hypernyms for 202 input terms out of 400 input terms). Please refer to the *Results Chapter* for more details. In order to improve the recall value, we wanted to experiment with a new word-embedding. So we leanrt a word-embeddings from the noun and verb lemma definitions from WordNet. The characteristics of these word Embeddings are same as that of *UMBC Word-Embedding* matrix. Figure 3.9 shows a sample word embedding matrix created over a small text corpus. Figure 3.10 shows the steps involved in creating these word-embeddings.

3.2 Discovering Hypernyms

Once we have all the corpora pre-processed, we apply techniques specific to the type of corpus to extract a candidate hypernym or hypernyms for the given input term. The following are the techniques we used in our modules to extract hypernyms. We refer to these techniques as sub-systems in many sections of this thesis.

- **Co-occurrence Frequencies over the UMBC Normalized Corpus**

- **Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus**
- **Co-occurrence Frequencies over the UMBC Other Hearst Corpus**
- **Hypernymy Similarity Distance over the UMBC Word-Embedding**
- **Hearst Patterns over the Definition of new Out-Of-Vocabulary (OOV) Lemma**
- **Similarity with Definition over the Word-Embeddings**

3.2.1 Co-occurrence Frequencies over the UMBC Normalized Corpus

By using the Normalized Corpus built over the UMBC WebBase Corpus, this sub-system identifies a set of hypernyms for the given input hyponym term. For every input term, we iterate through all the paragraphs in this corpus to obtain their candidate hypernyms. If a paragraph consists of the input term, then all its words are added to a list. This list could contain repeating elements which would determine the frequency of each element at a later point. Each input term has its own independent list. Once the entire corpus is iterated and all the words in the context of the input terms are added to the list, a final set is created from this list. This set consists of the sorted list elements and they are sorted based on their frequencies in descending order. List elements which do not have a minimum frequency - **5** - are removed from the final set. We further refine this set by removing the hypernyms which do not belong to the *Vocabulary List* and store the remaining hypernyms in a result file. Figure 3.11 shows the flow of this sub-system - Co-occurrence Frequencies over this UMBC Normalized Corpus. Please refer to sub-section *e-assign Sense* for more

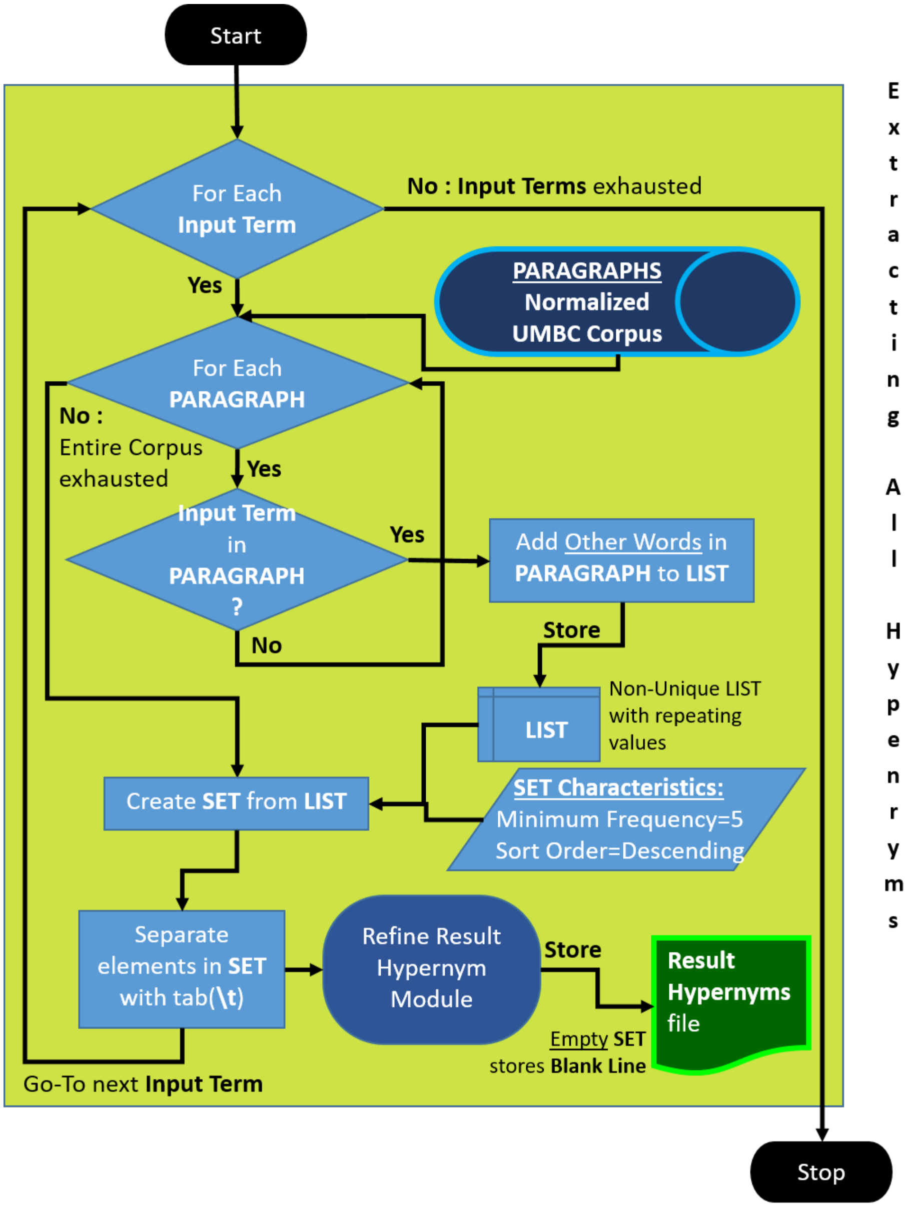
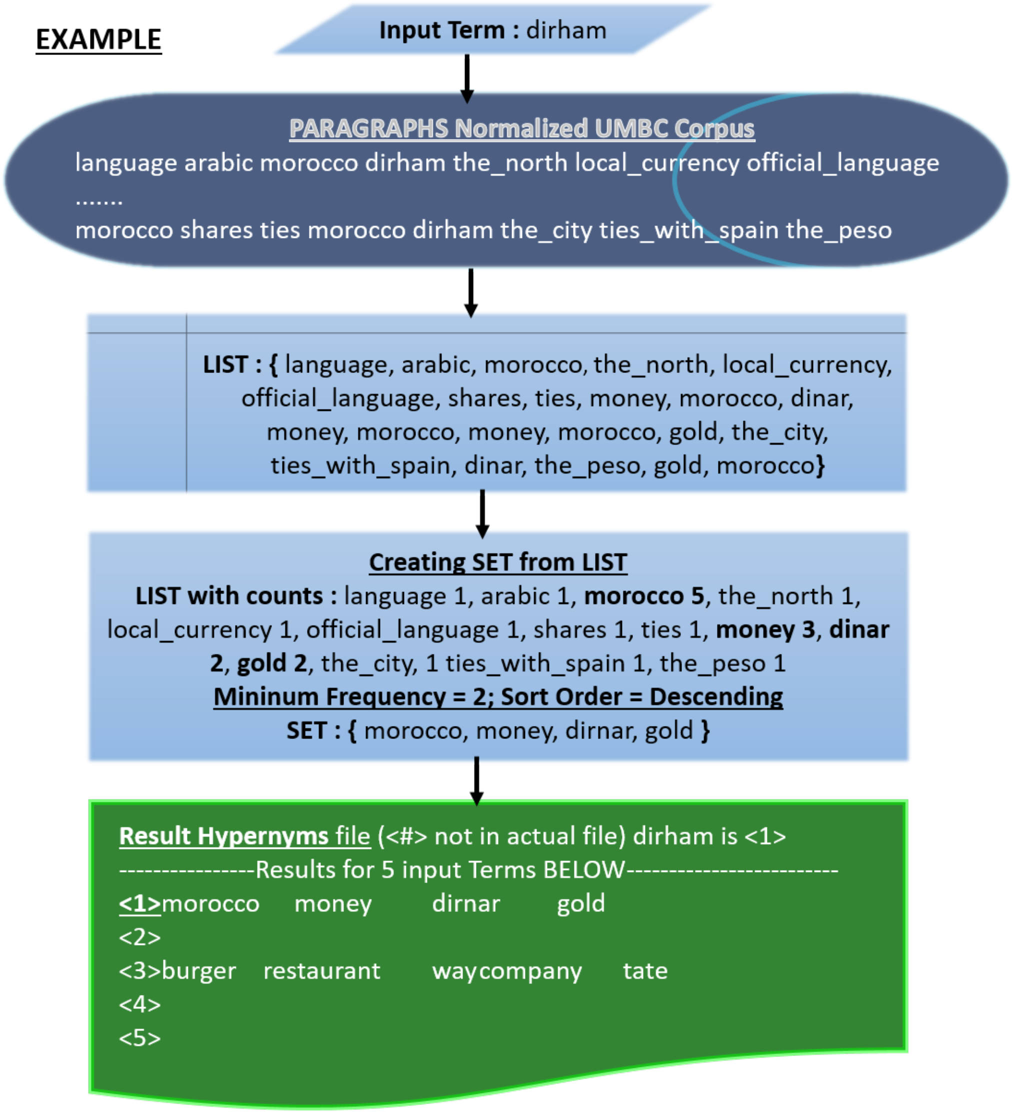


Figure 3.11: Extracting hypernyms using Co-occurrence Frequencies over UMBC Normalized Corpus

details on how we refine the predicted hypernyms. Figure 3.12 shows an example on how this sub-system retrieves candidate hypernyms for the input term "*dirham*".


EXAMPLE
Input Term : dirham
PARAGRAPHS Normalized UMBC Corpus
language arabic morocco dirham the_north local_currency official_language
....
morocco shares ties morocco dirham the_city ties_with_spain the_peso
LIST : { language, arabic, morocco, the_north, local_currency, official_language, shares, ties, money, morocco, dinar, money, morocco, money, morocco, gold, the_city, ties_with_spain, dinar, the_peso, gold, morocco}
Creating SET from LIST
LIST with counts : language 1, arabic 1, morocco 5, the_north 1, local_currency 1, official_language 1, shares 1, ties 1, money 3, dinar 2, gold 2, the_city, 1 ties_with_spain 1, the_peso 1
Mininum Frequency = 2; Sort Order = Descending
SET : { morocco, money, dirnar, gold }
Result Hypernyms file (<#> not in actual file) dirham is <1>

```
----------------Results for 5 input Terms BELOW------------------------
<1>morocco    money      dirnar      gold
<2>
<3>burger   restaurant      waycompany      tate
<4>
<5>
```

Figure 3.12: Using Co-occurrence Frequencies to extract hypernyms for *dirham* from UMBC Normalized Corpus

3.2.2 Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus

This sub-system identifies candidate hypernyms from the *Hypernym* part of this corpus by matching the input term to the *Hyponym* part of it. For every input term, we iterate through all the *Hyponym : Hypernym* pairs in this corpus. If a *Hyponym*

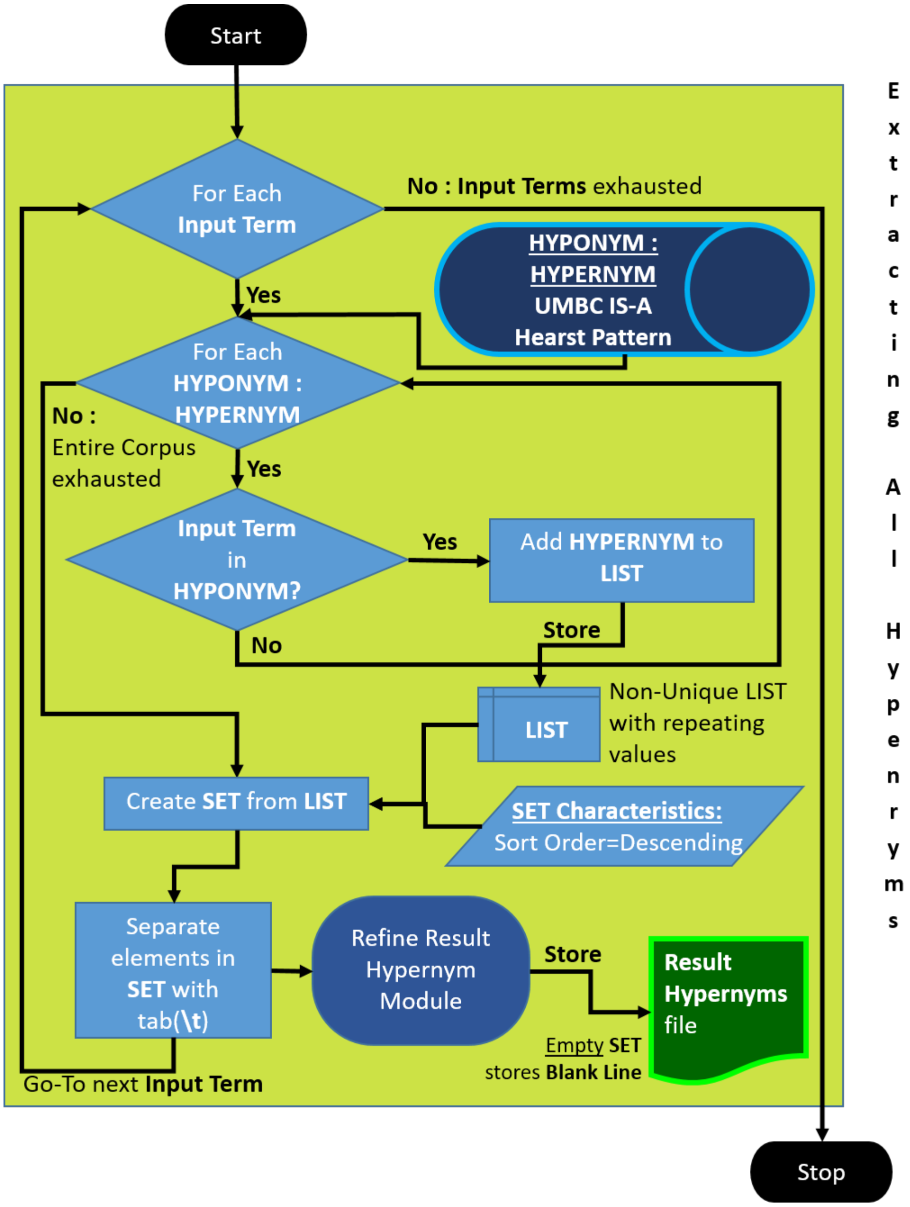


Figure 3.13: Extracting hypernyms using Co-occurrence Frequencies over UMBC IS-A Hearst Corpus

matches with the input term, then its *Hypernym* is added to a list. For each input term we maintain an independent list. Once the entire corpus is processed and all the valid *hypernyms* are added to the list, a final set is created in a similar fashion as the previous sub-system. Figure 3.13 shows this flow of the sub-system co-occurrence

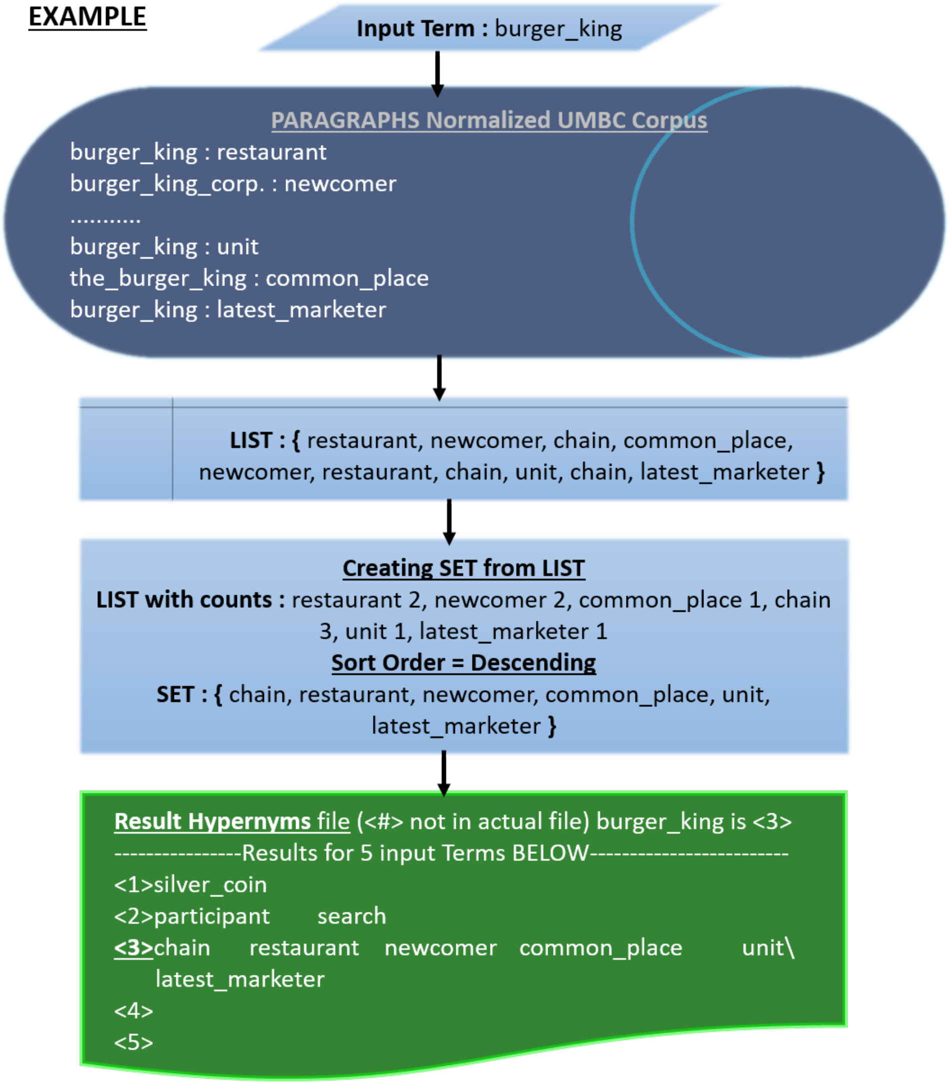

EXAMPLE
Input Term : burger_king
PARAGRAPHS Normalized UMBC Corpus
burger_king : restaurant
burger_king_corp. : newcomer
....
burger_king : unit
the_burger_king : common_place
burger_king : latest_marketer
LIST : { restaurant, newcomer, chain, common_place, newcomer, restaurant, chain, unit, chain, latest_marketer }
Creating SET from LIST
LIST with counts : restaurant 2, newcomer 2, common_place 1, chain 3, unit 1, latest_marketer 1
Sort Order = Descending
SET : { chain, restaurant, newcomer, common_place, unit, latest_marketer }
Result Hypernyms file (<#> not in actual file) burger_king is <3>
----------------Results for 5 input Terms BELOW------------------------
<1>silver_coin
<2>participant search
<3>chain restaurant newcomer common_place unit\ latest_marketer
<4>
<5>

Figure 3.14: Using Co-occurrence Frequencies to extract hypernyms for *burger_king* from UMBC IS-A Hearst Corpus

frequencies over the UMBC IS-A Hearst Corpus. Figure 3.14 shows an example of how this sub-system retrieves candidate hypernyms for the input term "*Burger King*". If this corpus is used to determine a hypernym for a new out-of-vocabulary lemma from WordNet, then the candidate hypernyms fetched are refined using the lemmas from WordNet. Refer the SemEval 2016 Task 14 - Semantic Taxonomy Enrichment sub-section of *Refine Result Hypernym(s)* (Section 3.2.7) section for more information.

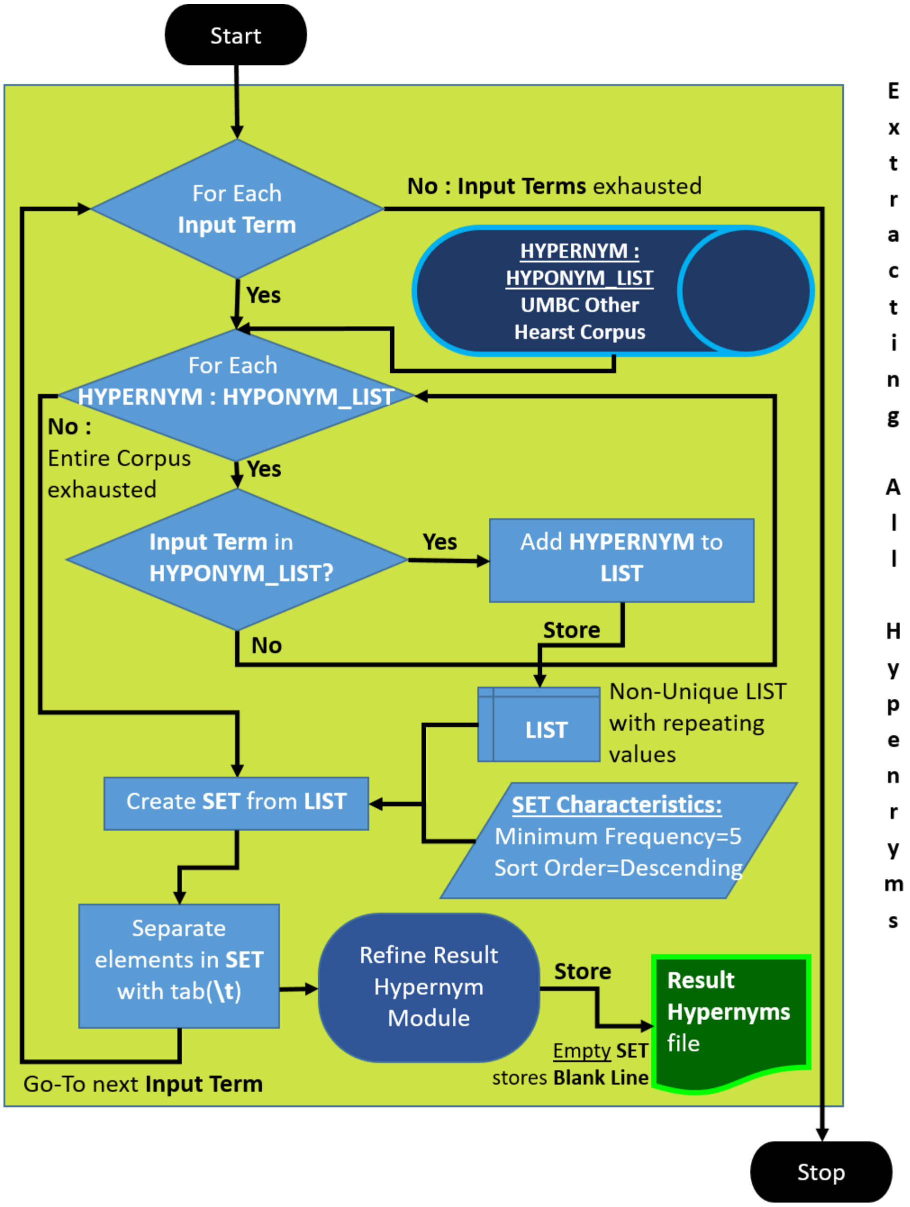

Start
For Each Input Term
No : Input Terms exhausted
Yes
HYPERNYM : HYPONYM_LIST UMBC Other Hearst Corpus
For Each HYPERNYM : HYPONYM_LIST
No : Entire Corpus exhausted
Yes
Input Term in HYPONYM_LIST?
Yes
Add HYPERNYM to LIST
Store
No
LIST
Non-Unique LIST with repeating values
Create SET from LIST
SET Characteristics: Minimum Frequency=5 Sort Order=Descending
Separate elements in SET with tab(\t)
Refine Result Hypernym Module
Store
Result Hypernyms file
Empty SET stores Blank Line
Go-To next Input Term
Extracting All Hypernyms
Stop

Figure 3.15: Extracting hypernyms using co-occurrence frequencies over UMBC Other Hearst Corpus

3.2.3 Co-occurrence Frequencies over the UMBC Other Hearst Corpus

The execution of this sub-system is similar to the previous sub-system "co-occurrence Frequencies over the UMBC IS-A Hearst Corpus". Since the structure of this corpus

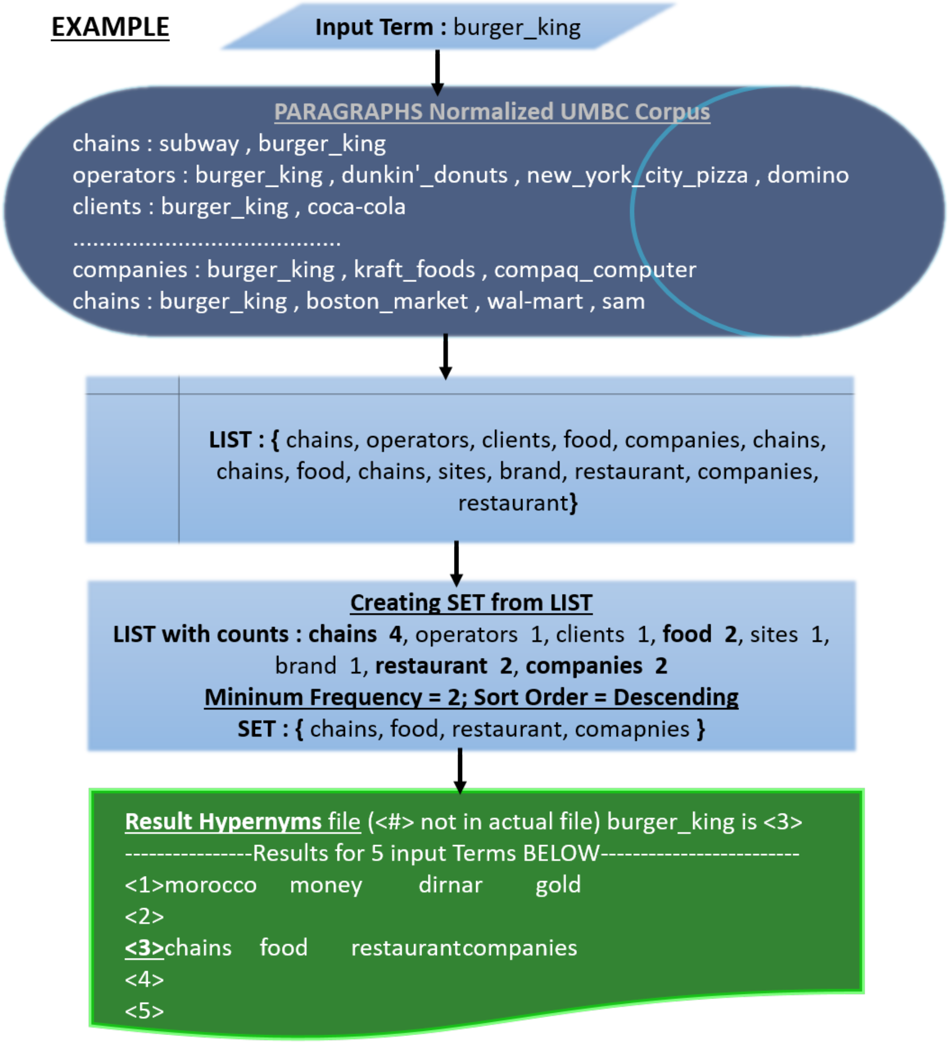

EXAMPLE
Input Term : burger_king
PARAGRAPHS Normalized UMBC Corpus
chains : subway , burger_king
operators : burger_king , dunkin'_donuts , new_york_city_pizza , domino
clients : burger_king , coca-cola
....
companies : burger_king , kraft_foods , compaq_computer
chains : burger_king , boston_market , wal-mart , sam
LIST : { chains, operators, clients, food, companies, chains, chains, food, chains, sites, brand, restaurant, companies, restaurant}
Creating SET from LIST
LIST with counts : chains 4, operators 1, clients 1, food 2, sites 1, brand 1, restaurant 2, companies 2
Mininum Frequency = 2; Sort Order = Descending
SET : { chains, food, restaurant, comapnies }
Result Hypernyms file (<#> not in actual file) burger_king is <3>
----------------Results for 5 input Terms BELOW------------------------
<1>morocco money dirnar gold
<2>
<3>chains food restaurantcompanies
<4>
<5>

Figure 3.16: Using co-occurrence frequencies to extract hypernyms for *burger_king* from UMBC Other Hearst Corpus

is slightly different than the structure of the UMBC IS-A Hearst Corpus, identifying candidate hypernyms from the previous patterns using this corpus would be different. Here, if the input term exists in the *Hyponym List* of this corpus structure - "*Hypernym : Hyponym List*", then this *Hypernym* is added to the candidate hypernym list. The rest of the steps are same as the previous sub-system. Figure 3.15 shows the execution of this sub-system. Figure 3.16 shows how candidate hypernyms are identified for the input term "*Burger King*".

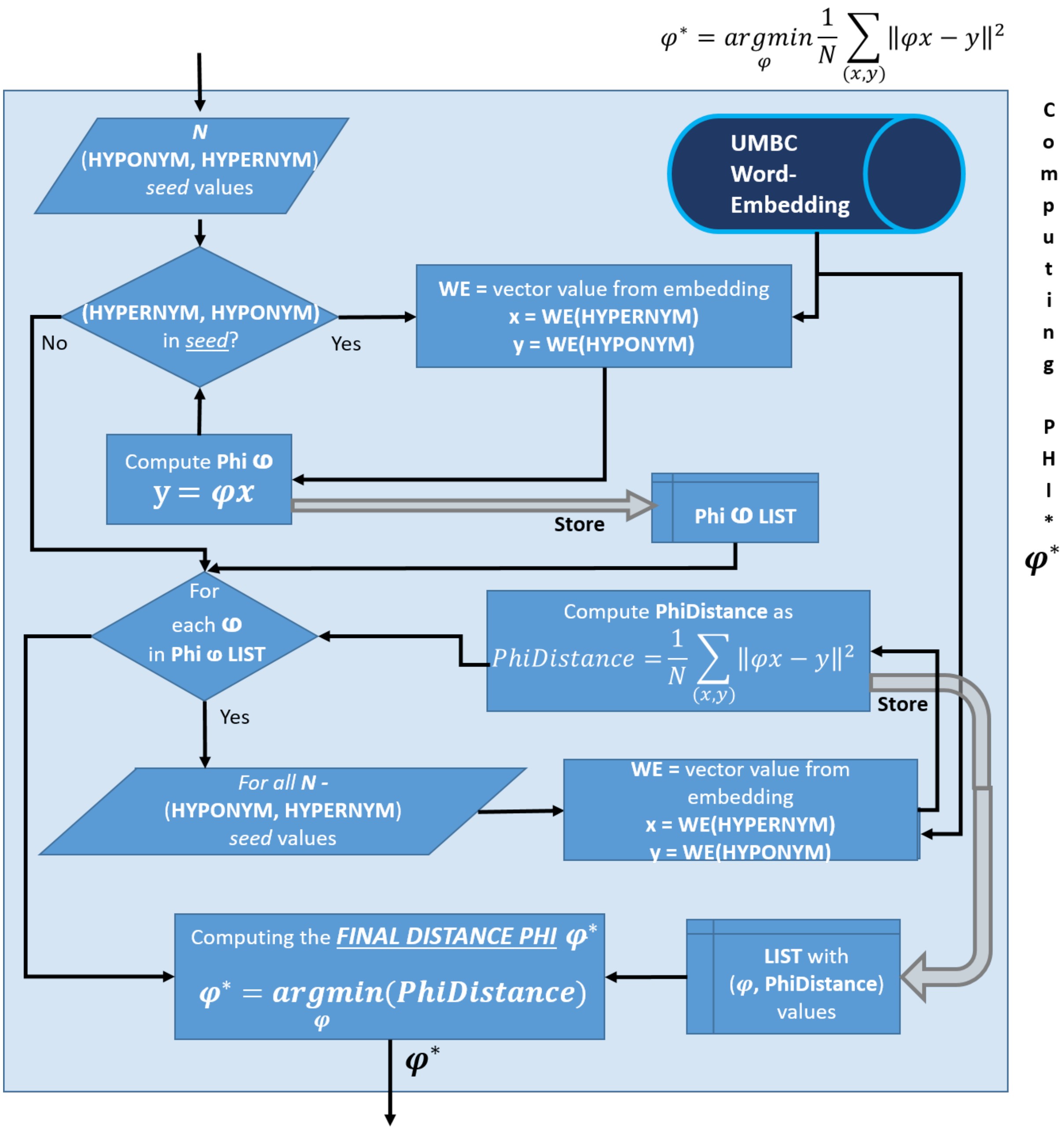

$\varphi^* = \underset{\varphi}{argmin} \frac{1}{N} \sum_{(x,y)} \|\varphi x - y\|^2$
Computing PHI *
φ^*
N
(HYPONYM, HYPERNYM)
seed values
UMBC Word-Embedding
(HYPERNYM, HYPONYM) in seed?
No
Yes
WE = vector value from embedding
x = WE(HYPERNYM)
y = WE(HYPONYM)
Compute Phi ω
$y = \varphi x$
Store
Phi ω LIST
For each ω in Phi ω LIST
Compute PhiDistance as
$PhiDistance = \frac{1}{N} \sum_{(x,y)} \|\varphi x - y\|^2$
Store
Yes
For all N -
(HYPONYM, HYPERNYM)
seed values
WE = vector value from embedding
x = WE(HYPERNYM)
y = WE(HYPONYM)
Computing the FINAL DISTANCE PHI φ^*
$\varphi^* = \underset{\varphi}{argmin}(PhiDistance)$
LIST with
(φ, PhiDistance)
values
φ^*

Figure 3.17: Computing the Φ^* distance using training data as hypernym-hyponym seed values

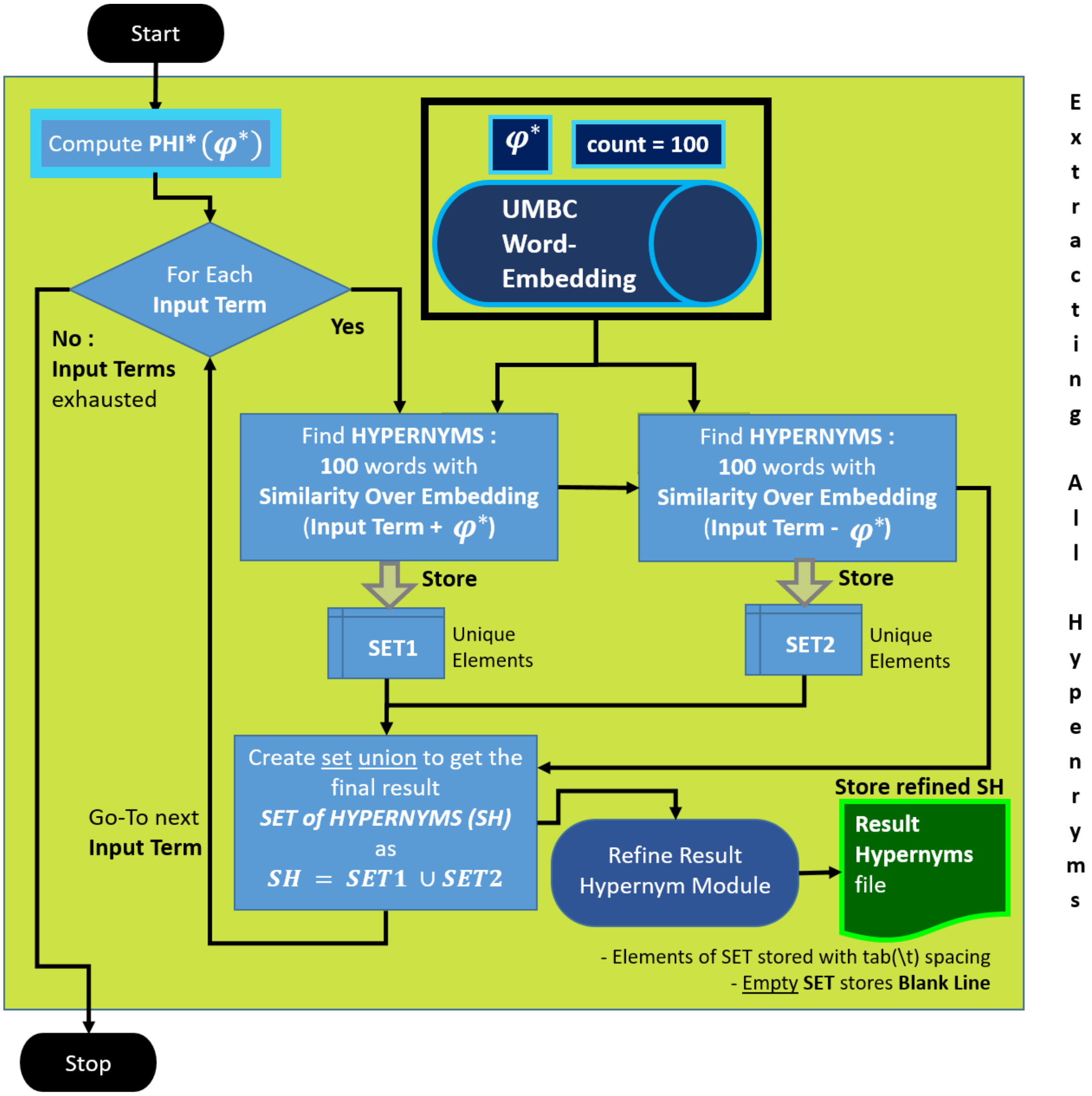

Start
Compute PHI* (φ^*)
For Each Input Term
Yes
No : Input Terms exhausted
φ^*
count = 100
UMBC Word-Embedding
Find HYPERNYMS : 100 words with Similarity Over Embedding (Input Term + φ^*)
Find HYPERNYMS : 100 words with Similarity Over Embedding (Input Term - φ^*)
Store
Store
SET1
Unique Elements
SET2
Unique Elements
Create set union to get the final result SET of HYPERNYMS (SH) as $SH = SET1 \cup SET2$
Go-To next Input Term
Refine Result Hypernym Module
Store refined SH
Result Hypernyms file
- Elements of SET stored with tab(\t) spacing
- Empty SET stores Blank Line
Stop
Extracting All Hypernyms

Figure 3.18: Extracting hypernyms from UMBC Word-Embedding using the pre-computed Φ^* value

3.2.4 Hypernymy Similarity Distance over the UMBC Word-Embedding

In order to extract *hypernyms* for a given input term from the UMBC Word-Embedding, we need to first determine a *distance* which represents the average hypernym-hyponym distance in this embedding. We call this distance the Φ^* *distance.*

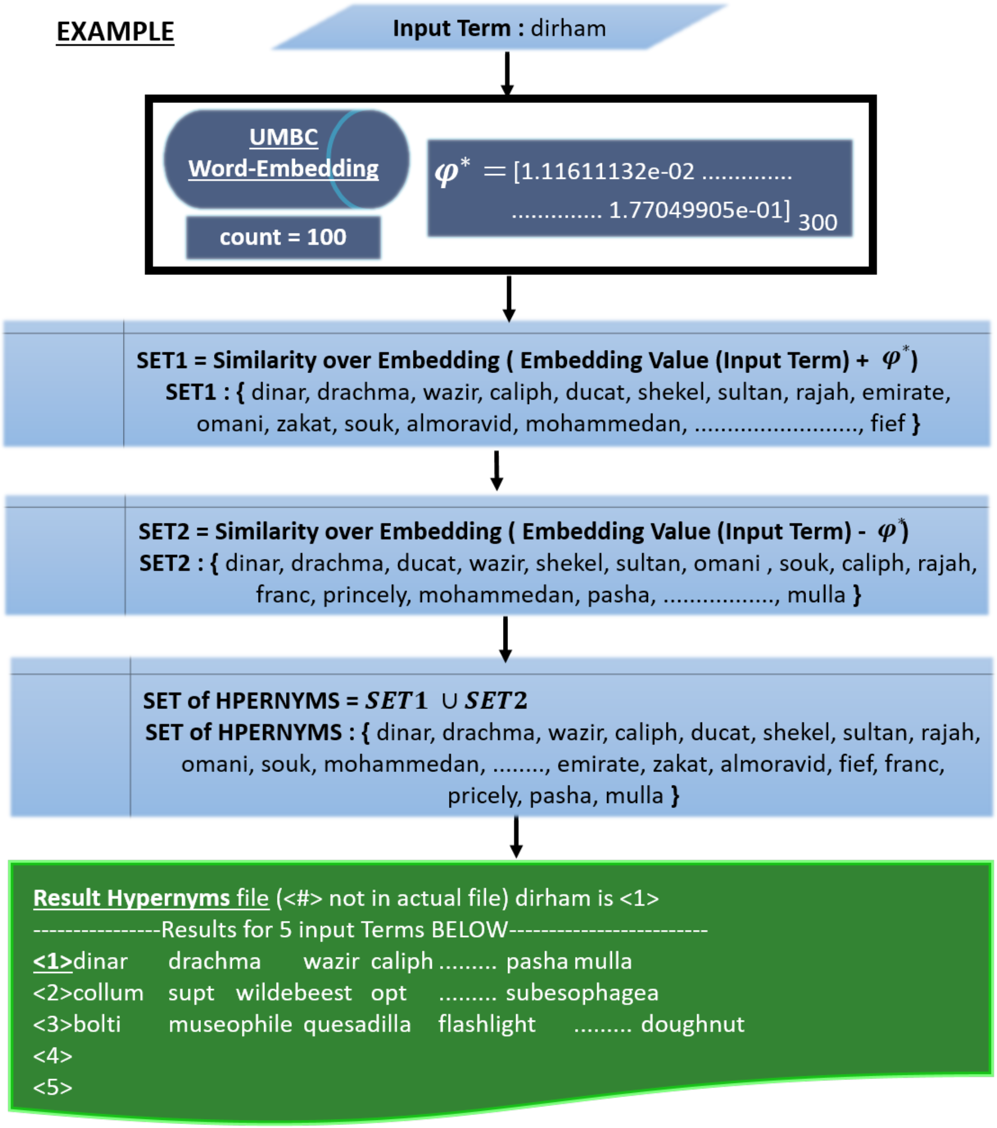

EXAMPLE
Input Term : dirham
UMBC Word-Embedding
count = 100
φ* = [1.11611132e-02 ....
.... 1.77049905e-01] 300
SET1 = Similarity over Embedding (Embedding Value (Input Term) + φ*)
SET1 : { dinar, drachma, wazir, caliph, ducat, shekel, sultan, rajah, emirate, omani, zakat, souk, almoravid, mohammedan, ..., fief }
SET2 = Similarity over Embedding (Embedding Value (Input Term) - φ*)
SET2 : { dinar, drachma, ducat, wazir, shekel, sultan, omani , souk, caliph, rajah, franc, princely, mohammedan, pasha, ..., mulla }
SET of HPERNYMS = SET1 ∪ SET2
SET of HPERNYMS : { dinar, drachma, wazir, caliph, ducat, shekel, sultan, rajah, omani, souk, mohammedan, ..., emirate, zakat, almoravid, fief, franc, pricely, pasha, mulla }
Result Hypernyms file (<#> not in actual file) dirham is <1>
----------------Results for 5 input Terms BELOW------------------------
<1>dinar drachma wazir caliph .... pasha mulla
<2>collum supt wildebeest opt .... subesophagea
<3>bolti museophile quesadilla flashlight .... doughnut
<4>
<5>

Figure 3.19: Using Φ* value over UMBC Word-Embedding to extract hypernyms for the input hyponym term *dirham*

To compute this distance, we first create *hypernym-hyponym seed pairs* from the training data's input and gold files. For example, if the input hyponym term from the training data is "*burger_king*" and its gold data hypernyms are "*eating_house, restaurant, chain*" then the seed pairs {*(eating_house, burger_king), (restaurant, burger_king), (chain, burger_king)*} are created. We use all such seed pairs from the training data to compute Φ^* by using the below formula. The details of this computation are explained in the *figure 3.17*.

$$\Phi^* = \text{argmin}_\Phi \frac{1}{N} \sum_{(x,y)} \|\Phi x - y\|^2 \tag{3.1}$$

Once we get this distance with respect to the UMBC Corpus Embedding, it is used to fetch candidate hypernyms for any new input hyponym term. For a new input term, the words which exist at Φ^* distance on either side of its vector value are the possible candidate hypernyms from this embedding. Hence we use this Φ^* value as both the positive and the negative value in the word2vec similarity module. Figure 3.18 shows how this distance value is used along with the input term over an embedding to fetch the candidate hypernym results. *Similarity over Embedding* in this figure represents the word2vec similarity function. The candidates fetched are refined by using the vocabulary file. Figure 3.16 shows an example of how this sub-system retrieves candidate hypernyms for the input term "*dirham*".

3.2.5 Hearst Patterns over the Definition of new Out-Of-Vocabulary(OOV) Lemma

This sub-system is applicable only to *SemEval 2016 Task 14 - Semantic Taxonomy Enrichment* (Chapter 2 Section 2.4.4). This module relies only on the new OOV's definition. We apply the **Hearst Patterns** over these definitions in a particular order to extract hypernyms which exist in WordNet. Figure 3.20 shows the order in which we apply these Hearst Patterns to identify a hypernym for a given new OOV lemma. More information about Hearst Patterns can be found in *Background Chapter*(Chapter2). We apply Hearst Patterns only when the OOV lemma is a noun as

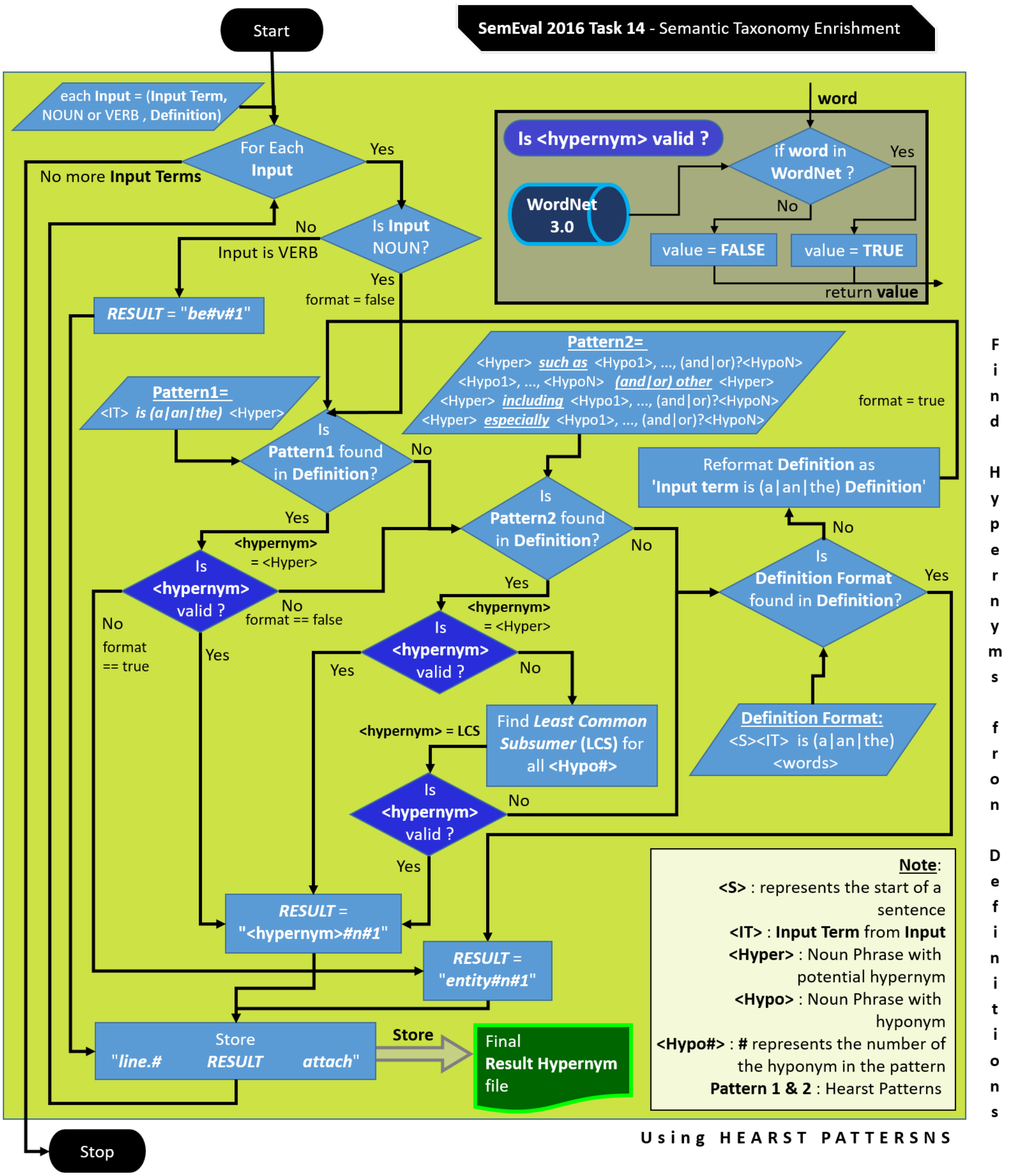

Start
SemEval 2016 Task 14 - Semantic Taxonomy Enrishment
each Input = (Input Term, NOUN or VERB , Definition)
For Each Input
No more Input Terms
Yes
No
Is Input NOUN?
Input is VERB
Yes
format = false
RESULT = "be#v#1"
word
Is <hypernym> valid ?
if word in WordNet ?
Yes
No
WordNet 3.0
value = FALSE
value = TRUE
return value
Pattern2=
<Hyper> such as <Hypo1>, ..., (and|or)?<HypoN>
<Hypo1>, ..., <HypoN> (and|or) other <Hyper>
<Hyper> including <Hypo1>, ..., (and|or)?<HypoN>
<Hyper> especially <Hypo1>, ..., (and|or)?<HypoN>
format = true
Pattern1=
<IT> is (a|an|the) <Hyper>
Is Pattern1 found in Definition?
No
Reformat Definition as 'Input term is (a|an|the) Definition'
Yes
<hypernym> = <Hyper>
Is Pattern2 found in Definition?
No
No
Is Definition Format found in Definition?
Yes
Is <hypernym> valid ?
No
format == false
No
format == true
Yes
Yes
<hypernym> = <Hyper>
Is <hypernym> valid ?
Yes
No
Find Least Common Subsumer (LCS) for all <Hypo#>
<hypernym> = LCS
Definition Format:
<S><IT> is (a|an|the) <words>
Is <hypernym> valid ?
No
Yes
RESULT = "<hypernym>#n#1"
RESULT = "entity#n#1"
Note:
<S> : represents the start of a sentence
<IT> : Input Term from Input
<Hyper> : Noun Phrase with potential hypernym
<Hypo> : Noun Phrase with hyponym
<Hypo#> : # represents the number of the hyponym in the pattern
Pattern 1 & 2 : Hearst Patterns
Store "line.# RESULT attach"
Store
Final Result Hypernym file
Stop
Using HEARST PATTERSNS
Find Hypernyms from Definitions

Figure 3.20: Extracting hypernyms from definitions using Hearst Patterns

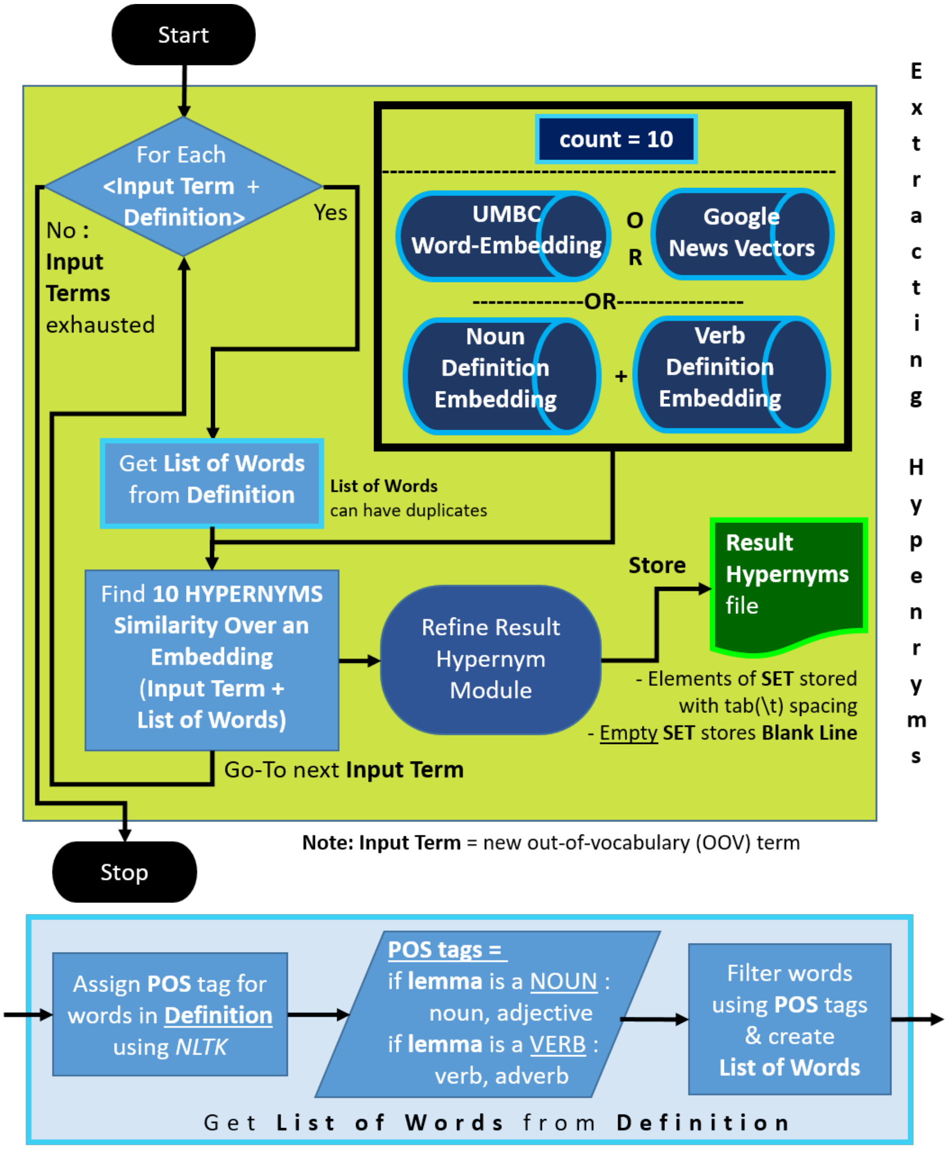
the patterns can be applied only to noun phrases. If *Pattern1* exists in the definition, then we search WordNet for the head word in the $\langle Hyper \rangle$ noun phrase. If this word exists in WordNet, then it is considered the result hypernym. If this word does not exist in WordNet or if *Pattern1* is not found in the definition, then we apply *Pattern2* to this definition. If this pattern is found, then we search for $\langle Hyper \rangle$'s headword in WordNet. If found, then this word is the result hypernym, otherwise we identify the *Least Common Subsumer (LCS)* for the $\langle Hypo1 \rangle, ..., \langle HypoN \rangle$ terms from WordNet. If we locate this *LCS*, then we would assign this as the result hypernym. If we do not locate a hypernym for this OOV lemma, we rewrite this definition as "*OOV Lemma* is a *old definition*" and re-apply *Pattern1*. Finally, if a hypernym is identified, then we consider it as our result. If we still cannot locate a valid hypernym even after redefining the OOV lemma, "*entity#n#1*" is added as the default result hypernym if the given input term is a *noun*.

If the given input term is a *verb*, this sub-system would add "*be#v#1*" as default hypernym.

3.2.6 Similarity with Definition over the Word-Embeddings

When a definition is provided with a new out-of-vocabulary (OOV) input term, we can use the keywords in this definition to extract candidate hypernyms from any word-embedding matrix. If the input term is a noun, then the nouns and adjectives in the input term's definition are the keywords. If this input term is a verb, then the verbs and adverbs in it's definition are the keywords. For example, if the noun input term is "*ger toshav*" and its definition is "*Lit. a resident stranger, a non-Jewish inhabitant of the Land of Israel who observes the Seven Laws of Noah and has repudiated all links with idolatry.*"[5], then the keywords from this definition are "*lit, resident, stranger,*

[5] http://www.reformjudaism.org/glossary-search?keyword=&page=8

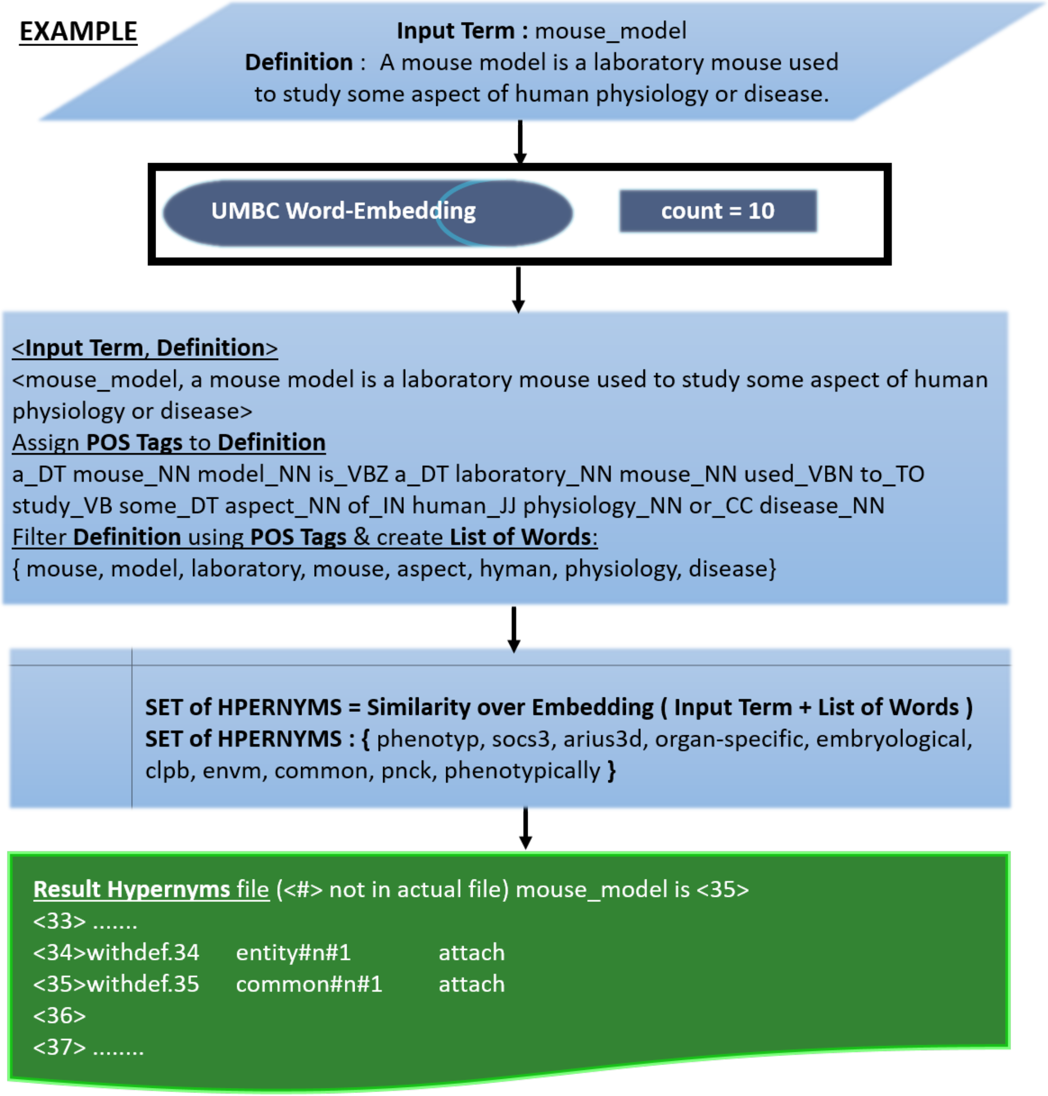


Start
For Each <Input Term + Definition>
Yes
No : Input Terms exhausted
count = 10
UMBC Word-Embedding
OR
Google News Vectors
-------------OR---------------
Noun Definition Embedding
+
Verb Definition Embedding
Get List of Words from Definition
List of Words can have duplicates
Store
Result Hypernyms file
Find 10 HYPERNYMS Similarity Over an Embedding (Input Term + List of Words)
Refine Result Hypernym Module
- Elements of SET stored with tab(\t) spacing
- Empty SET stores Blank Line
Go-To next Input Term
Extracting Hypernyms
Stop
Note: Input Term = new out-of-vocabulary (OOV) term
Assign POS tag for words in Definition using NLTK
POS tags =
if lemma is a NOUN : noun, adjective
if lemma is a VERB : verb, adverb
Filter words using POS tags & create List of Words
Get List of Words from Definition

Figure 3.21: Extracting hypernyms from Word-Embeddings using the keywords in the Definitions

non-jewish, inhabitant, land, israel, seven, laws, noah, links, idolatry". If the input term is a verb - "*evergreen*", and the definition is "*To set the repayment rate of a loan at or below the interest rate, so low that the principal will never be repaid.*"[6], then the keywords are "*set, so, never, be, repaid*". Figure 3.21 shows how these

[6] https://en.wiktionary.org/wiki/evergreen#English



Input Term : mouse_model
Definition : A mouse model is a laboratory mouse used to study some aspect of human physiology or disease.

<**Input Term, Definition**>
<mouse_model, a mouse model is a laboratory mouse used to study some aspect of human physiology or disease>
Assign **POS Tags** to **Definition**
a_DT mouse_NN model_NN is_VBZ a_DT laboratory_NN mouse_NN used_VBN to_TO study_VB some_DT aspect_NN of_IN human_JJ physiology_NN or_CC disease_NN
Filter **Definition** using **POS Tags** & create **List of Words**:
{ mouse, model, laboratory, mouse, aspect, hyman, physiology, disease}

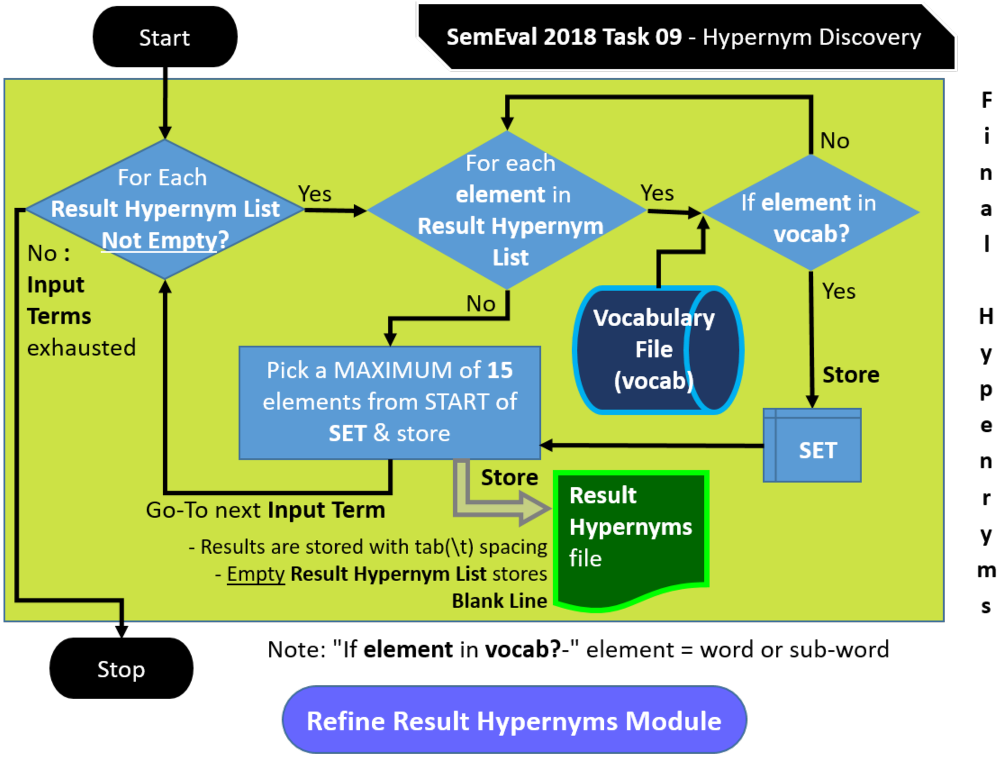
SET of HPERNYMS = Similarity over Embedding (Input Term + List of Words)
SET of HPERNYMS : { phenotyp, socs3, arius3d, organ-specific, embryological, clpb, envm, common, pnck, phenotypically **}**

Result Hypernyms file (<#> not in actual file) mouse_model is <35>
<33> ....
<34>withdef.34 entity#n#1 attach
<35>withdef.35 common#n#1 attach
<36>
<37> ....

Figure 3.22: Using Φ^* value over UMBC Word-Embedding to extract hypernyms for the input hyponym term *dirham*

keywords are used along with input in fetching a hypernym from any word-embedding matrix. *Similarity over an Embedding* in this figure represents the word2vec similarity function. We fetch the top **10** words which fall in this similarity range as candidate hypernyms. The first word from this hypernym set which exists in WordNet is the *result hypernym* with a default sense *#1*, and "*attach*" is the default operation for this sub-system. Please refer to the sub-section *Refine Result Hypernym(s)* (Section

3.2.7) for more details about refining the result hypernyms for *SemEval 2016 Task 14 - Semantic Taxonomy Detection*. We use this procedure to extract hypernyms from *UMBC Word-Embedding*, *Google News Vectors* and *WordNet Definition Embeddings*. Figure 3.22 shows an example of how this sub-system retrieves a hypernym for the input term "*Mouse Model*" using *UMBC Word-Embedding* matrix.



Start
SemEval 2018 Task 09 - Hypernym Discovery
For Each Result Hypernym List Not Empty?
Yes
For each element in Result Hypernym List
Yes
If element in vocab?
No
Yes
No : Input Terms exhausted
No
Vocabulary File (vocab)
Pick a MAXIMUM of 15 elements from START of SET & store
Store
SET
Store
Result Hypernyms file
Go-To next Input Term
- Results are stored with tab(\t) spacing
- Empty Result Hypernym List stores Blank Line
Final Hypernyms
Stop
Note: "If element in vocab?-" element = word or sub-word
Refine Result Hypernyms Module

Figure 3.23: Refining the Candidate Hypernyms using the *Vocabulary* file

3.2.7 Refine Result Hypernym(s)

From all the sub-systems - Co-occurrence Frequencies over the UMBC corpora (Normalized Corpus, IS-A Hearst Corpus, Other Hearst Corpus), Hypernymy Similarity Distance over the UMBC Word-Embedding and Similarity with Definition over the Word-Embeddings - we identify only valid hypernyms for all the input terms. The functionality of the last step in these sub-systems is to retain only valid hypernyms

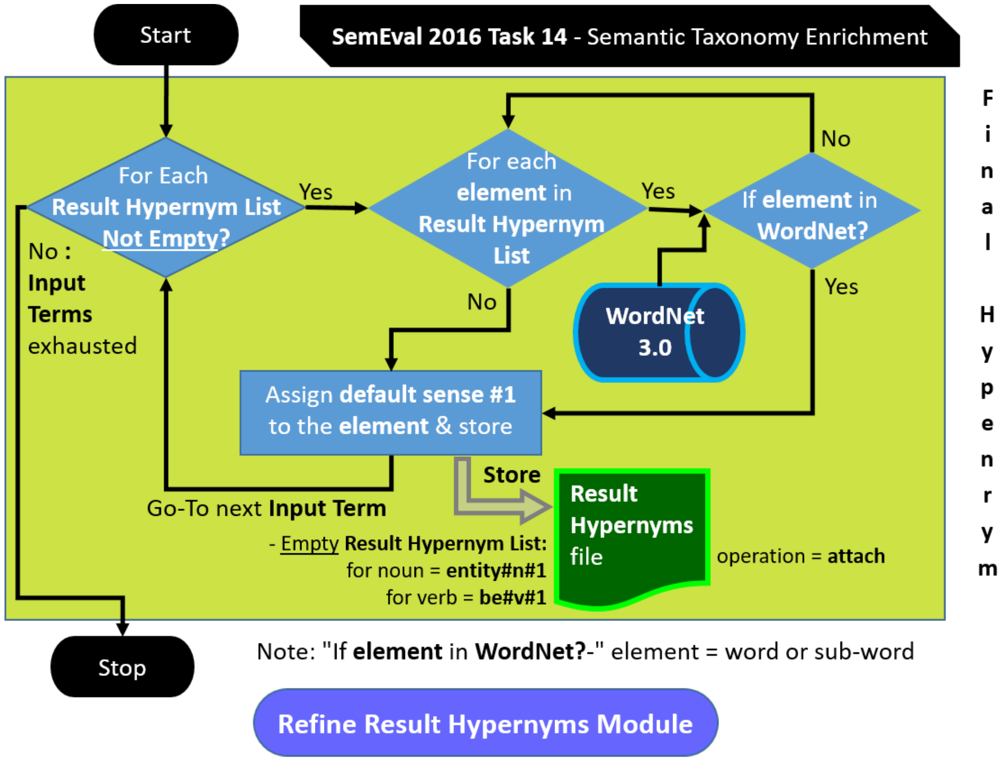

Start
SemEval 2016 Task 14 - Semantic Taxonomy Enrichment
For Each Result Hypernym List Not Empty?
Yes
For each element in Result Hypernym List
Yes
If element in WordNet?
No
Yes
No
No : Input Terms exhausted
WordNet 3.0
Assign default sense #1 to the element & store
Store
Result Hypernyms file
operation = attach
Go-To next Input Term
- Empty Result Hypernym List:
for noun = entity#n#1
for verb = be#v#1
Final Hypernym
Stop
Note: "If element in WordNet?-" element = word or sub-word
Refine Result Hypernyms Module

Figure 3.24: Identifying a one Hypernym from candidate hypernyms using *WordNet*

by eliminating the hypernyms which do not exist in either the *Vocabulary File* or *WordNet 3.0*. For *SemEval 2018 Task 09 - Hypernym Discovery* task, we use the *Vocabulary* file provided with this task to refine the results. *Figure 3.23* shows how this sub-system refines the identified hypernyms and produces the final results. The result hypernyms for the *SemEval 2018 Task 09 - Hypernym Discovery* are refined by using *WordNet 3.0*. *Figure 3.24* shows how this sub-system uses WordNet to determine a final hypernym from the candidate hypernyms.

3.3 Re-assign Sense

The *Re-assign Sense* module is applied only to the results of the Semantic Taxonomy Enrichment task (Chapter 2 Section 2.4.4). *Homographs* are words which have the same spelling but different meaning. Homographs are stored in WordNet by using a sense number. For example, a word "*soap*" could mean "A cleaning agent" or "the money which is offered as a bribe". These homographs are added to WordNet as "soap#n#1" with definition "a cleansing agent made from the salts of vegetable or animal fats" and "soap#n#2" with definition "money offered as a bribe"[7]. In the *Discovering Hypernym* phase, all the result hypernyms are assigned a default sense "*#1*". Sense "*#1*" in WordNet is often the most frequent sense for any given lemma in the corpus which is used to create WordNet. For a new out-of-vocabulary(OOV) noun lemma "buyoff", our system predicts "soap#n#1" as the hypernym. The definition provided along with this OOV lemma is "Money paid illegally to get some work done.". The predicted sense in this scenario (where soap represents a cleaning agent) is not an appropriate sense for the hypernym of the input term *buyoff.* So by using the OOV's definition and the definitions of lemmas "soap#n#1" and "soap#n#2" from WordNet, we apply *Lesk* and *Extended Gloss Overlaps* Word Sense Disambiguation algorithms to determine a more appropriate sense number for *soap.* Now our system is able to identify *soap#n#2* as the desired hypernym. The functionality of these algorithms is explained in the *Background Chapter*(Chapter 2). The sense with higher Word Sense Disambiguation score is the final sense of a hypernym.

3.4 Merging the Results for System Babbage

We have explained various sub-systems which work independently to identify candidate hypernyms for a given input hyponym term. A sub-system could fetch hypernyms for some specific input terms and fail for some other input terms. Each of these sub-systems can have its own set of input terms for which it succeeds. So in order

[7] http://wordnetweb.princeton.edu/perl/webwn?s=soap

to fetch hypernyms for a maximum number of input terms, we decided to merge the results of these sub-systems. Based on the type of the task and the format of the output expected for the task, the functionality of this merge module varies.

- If the task is *SemEval 2016 Task 14 - Semantic Taxonomy Enrichment*, we choose **Select One** merge technique which reports only one result hypernym per input term.

- If the task is *SemEval 2018 Task 09 - Hypernym Discovery*, we choose **Merge All** merge technique which reports a maximum of 15 hypernyms as the result for input term.

3.4.1 Select One: SemEval 2016 Task 14 - Semantic Taxonomy Enrichment

For this task, each input hyponym should be assigned only one hypernym from WordNet. For an input term, if more than one sub-system fetches a hypernym, then we need a priority order with which we could choose the result of one sub-system over the result of another sub-system. The following are the sub-systems which fetch results for this task. They are listed in the order of **priority**.

1. Hearst Patterns over the Definition of new Out-Of-Vocabulary (OOV) Lemma a.k.a. ***Definition Hearst Pattern*** (Section 3.2.5)

2. Similarity with Definition over the Word-Embedding - UMBC Word-Embedding a.k.a. ***UMBC Word Embedding*** (Section 3.2.6)

3. Similarity with Definition over the Word-Embedding - Google News Vectors a.k.a. ***Google News Vectors*** (Section 3.2.6)

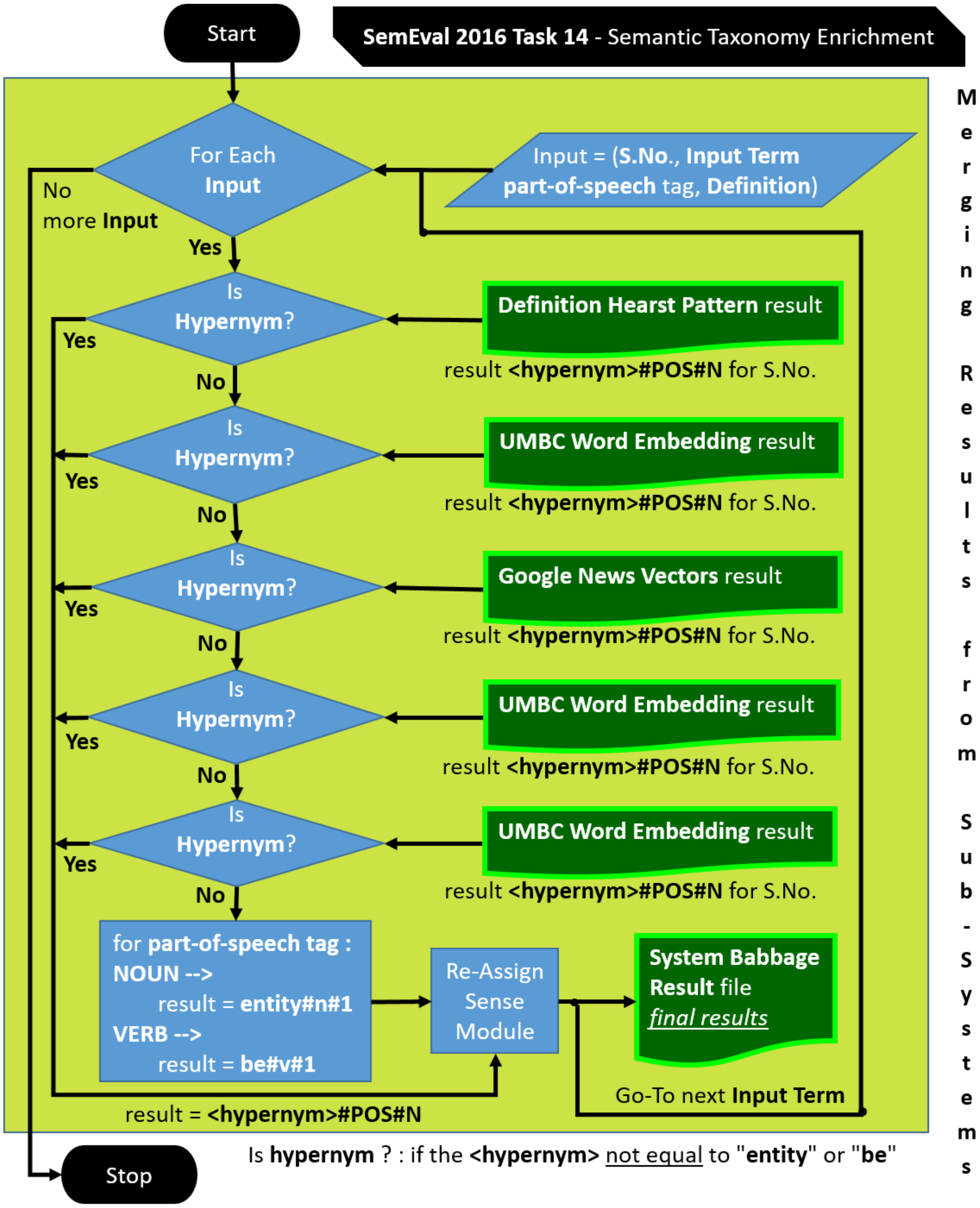

Start
SemEval 2016 Task 14 - Semantic Taxonomy Enrichment
For Each Input
Input = (S.No., Input Term part-of-speech tag, Definition)
No more Input
Yes
Is Hypernym?
Definition Hearst Pattern result
result <hypernym>#POS#N for S.No.
Yes
No
Is Hypernym?
UMBC Word Embedding result
result <hypernym>#POS#N for S.No.
Yes
No
Is Hypernym?
Google News Vectors result
result <hypernym>#POS#N for S.No.
Yes
No
Is Hypernym?
UMBC Word Embedding result
result <hypernym>#POS#N for S.No.
Yes
No
Is Hypernym?
UMBC Word Embedding result
result <hypernym>#POS#N for S.No.
Yes
No
for part-of-speech tag : NOUN --> result = entity#n#1 VERB --> result = be#v#1
Re-Assign Sense Module
System Babbage Result file final results
Go-To next Input Term
result = <hypernym>#POS#N
Stop
Is hypernym ? : if the <hypernym> not equal to "entity" or "be"
Merging Results from Sub-Systems

Figure 3.25: Final result file of System Babbage for SemEval 2016 Task 14 - Semantic Taxonomy Enrichment

4. Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus

   a.k.a. ***UMBC IS-A Hearst Patterns*** (Section 3.2.2)

5. Similarity with Definition over the Word-Embedding - WordNet Definition Em-

beddings

a.k.a. ***WordNet Definition Embeddings*** (Section 3.2.6)

Figure 3.25 represents of this merge order and the creation of the final result file, which we call the ***System Babbage*** result.

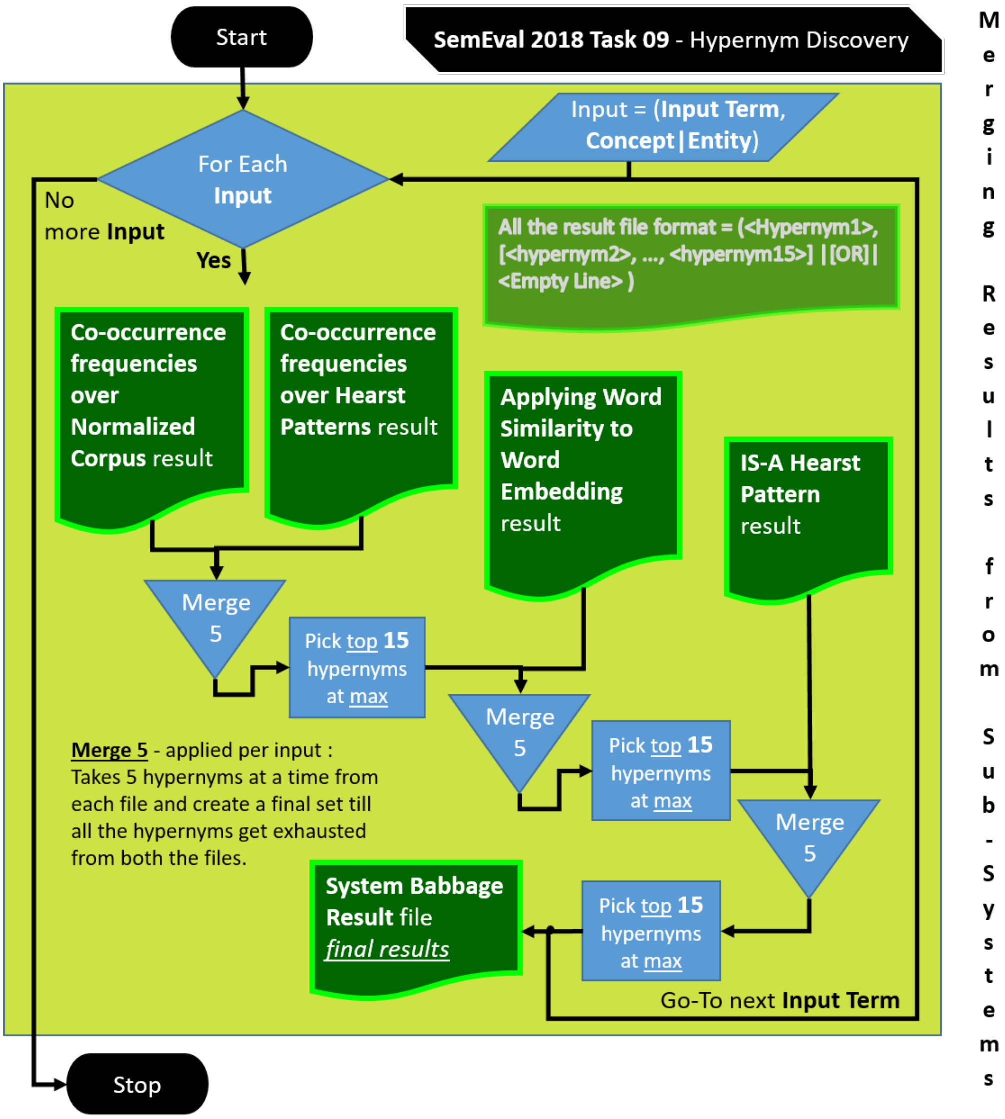

Start
SemEval 2018 Task 09 - Hypernym Discovery
Merging Results from Sub-Systems
Input = (Input Term, Concept|Entity)
For Each Input
No more Input
Yes
All the result file format = (<Hypernym1>, [<hypernym2>, ..., <hypernym15>] |[OR]| <Empty Line>)
Co-occurrence frequencies over Normalized Corpus result
Co-occurrence frequencies over Hearst Patterns result
Applying Word Similarity to Word Embedding result
IS-A Hearst Pattern result
Merge 5
Pick top 15 hypernyms at max
Merge 5
Pick top 15 hypernyms at max
Merge 5
Pick top 15 hypernyms at max
Merge 5 - applied per input : Takes 5 hypernyms at a time from each file and create a final set till all the hypernyms get exhausted from both the files.
System Babbage Result file final results
Go-To next Input Term
Stop

Figure 3.26: Final result file of System Babbage for SemEval 2018 Task 09 - Hypernym Discovery

3.4.2 Merge All: SemEval 2018 Task 09 - Hypernym Discovery

Unlike the previous task, each input hyponym in this task should be assigned multiple candidate hypernyms. The number of hypernyms retrieved per input hyponym should not exceed a count of **15**. For an input term, if more than one sub-system fetches the list of candidate hypernyms, instead of choosing based on sub-system's result, we choose to merge the results from all these sub-systems. We merge the results from only two sub-systems at a time. Once we merge results of two sub-systems, we merge these results with the results of another sub-system. We continue this process till all the sub-system results are merged to form the final system - **System Babbage** - results. Following are the sub-systems which fetch results for this task. We merge the results of sub-system **#4** with the merged results of sub-systems **#2** and **#3**. Then we finally merge this result with the results of sub-system **#1**. We choose this ordering by combining the results of the training data and choosing the order pattern which has the highest *Mean Average Reciprocal (MAR)* score.

1. Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus
   a.k.a. ***IS-A Hearst Pattern*** 3.2.2)

2. Co-occurrence Frequencies over the UMBC Normalized Corpus
   a.k.a. ***Co-occurrence frequencies over Normalized Corpus*** 3.2.1)

3. Co-occurrence Frequencies over the UMBC Other Hearst Corpus
   a.k.a. ***Co-occurrence frequencies over Hearst Patterns*** 3.2.3)

4. Hypernymy Similarity Distance over the UMBC Word-Embedding
   a.k.a. ***Applying Word Similarity to Word Embedding*** 3.2.4)

Figure 3.26 is the representation of this merge process and the creation of the final result file. We call this result file as the ***System Babbage*** result.

4 Results

In this section we explain the various evaluation measures and the baseline systems proposed by the organizers of the two SemEval tasks - *SemEval 2016 Task 14 Semantic Taxonomy Enrichment* and *SemEval 2018 Task 09 Hypernym Discovery*. Then we present the performance of our proposed methods when measured against these evaluation measures and baseline systems. Please refer Chapter 2 Section 2.4.4 for more details about these SemEval tasks.

4.1 Evaluation Measures

There are various evaluation measures proposed for these tasks to evaluate the systems submitted to them. Both the tasks have independent evaluation measures because the structure of the results submitted for these tasks is different.

The following are the Evaluation measures used for SemEval 2016 Task 14 Semantic Taxonomy Enrichment:

- Wu & Palmer Similarity score
- Lemma Match score
- Recall

The following are the Evaluation measures used for SemEval 2018 Task 9 Hypernym Discovery:

- Mean Reciprocal Rank (MRR) score
- Precision@k (P@k) scores with k = 1,3,5 and 15
- Mean Average Precision @15 (MAP) score

4.1.1 Wu & Palmer Similarity Score

The Wu & Palmer Similarity score is one of the many available structure based similarity measures. This measure uses the *is-a* hierarchy structure of WordNet to compute the similarity score. The Wu & Palmer Similarity score for a new out-of-vocabulary input term is calculated by measuring the similarity between the *actual sysnset* and *the predicted synset* for the new out-of-vocabulary lemma. This similarity could be calculated by using formula 4.1. In this equation, LCS stands for Least Common Subsumer and it represents the first common ancestor of Synset1 and Synset2 in the WordNet is-a hierarchy structure. *depth(synset)* is the distance between the root node (which is *entity#n#1* for noun hierarchy structure) and the given *synset*.

$$\begin{split} Wu\&Palmer(Synset1, Synset2) = 2 * depth(LCS(Synset1, Synset2)) \\ /(depth(Synset1) + depth(Synset2)) \end{split} \tag{4.1}$$

Figure 4.1 shows the WordNet *is-a* hierarchy structure with the two target synsets *feline#n#1*, *pet#n#1*. The LCS for *feline#n#1* and *pet#n#1* is *animal#n#1*. The depths of *feline#n#1*, *pet#n#1* and *animal#n#1* from the root node *entity#n#1* are **13**, **8** and **7** respectively. These depths include both the root node and the respective target synset. So the calculated Wu & Palmer Similarity score between *feline#n#1* and *pet#n#1* is **0.6667**.

The Wu & Palmer Similarity score ranges between *0 and 1*. A high similarity

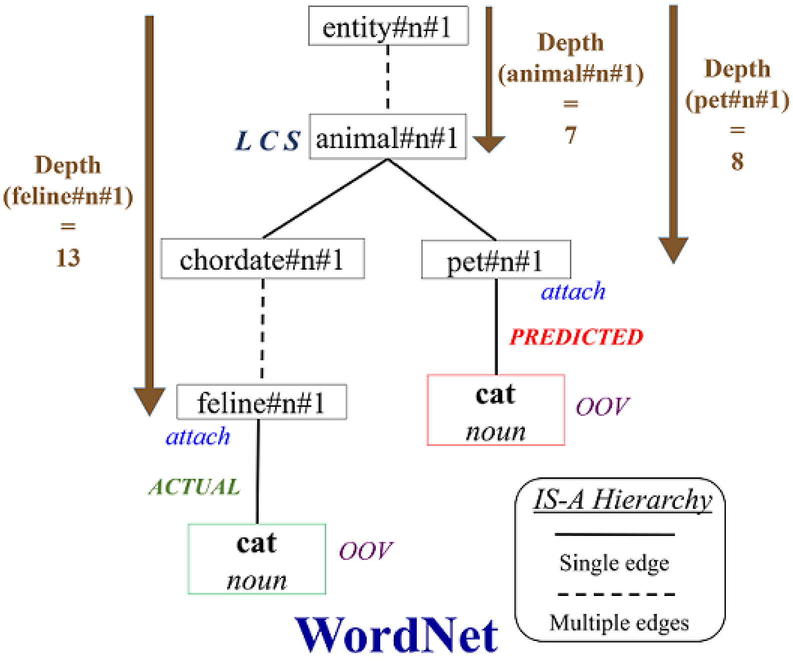


Figure 4.1: An example Wu & Palmer Similarity Measure using WordNet

score indicates that the two target synsets are more similar to one another. A lower similarity score indicates that these synsets are less similar to each other. If the score between the actual synset and the synset chosen by our system is high, then our system is capable of predicting a more precise hypernym[1]for a given lemma. Otherwise our system is unable to identify a potential hypernym for the given new lemma.

The Wu & Palmer Similarity measure not only scores the system based the synset predicted by the system but also scores the system based on the operation predicted by it. Our system is supposed to predict the operation - *attach* or *merge* along with the predicted result. When the chosen operation is attach, then the new out-of-vocabulary lemma is attached to the predicted synset as a new hyponym synset. If the operation is merge, then it is merged into the predicted synset as a synonym.

[1] Refer Chapter 2 Section 2.4.1 for more information

If the system predicted operation and the actual operation are different, then the similarity score would also be affected. For example, if both the system predicted result and the actual result is *feline#n#1 attach*, then the similarity score is equal to **1**. This is because both the predicted hypernym and the actual hypernym is the same synset from WordNet. But if our system predicted the result as *feline#n#1 attach* and the actual result is *feline#n#1 merge*, then the similarity score is *slightly less than* **1**. This is because the predicted synset is not equal to the actual synset but a direct hyponym of the actual synset. Finally, if the system predicted result and the actual result are *beast#n#1 attach* and *creature#n#1 attach* respectively, the similarity score between them is **1**. This is because both these lemmas belong to the same synset *animal#n#1.*

The operation which our systems chooses is attach by default for all the new out-of-vocabulary terms. This is because the key interest of this research is identifying hypernym(s) for a given input term.

4.1.2 Lemma Match score

The Lemma Match score, unlike the Wu & Palmer similarity score, evaluates our system based only on the synset predicted by the system and not on the operation chosen by it. It considers only the synsets of the actual and predicted lemma and does not consider the *is-a* hierarchy structure of WordNet. The lemma match score can either be *0 or 1.* If our system predicted synset is "*animal#n#1 merge*" and the actual synset is "*animal#n#1 merge*" the Lemma Match score is **1**. If our system predicted synset is "*animal#n#1 attach*" and the actual synset is "*animal#n#1 merge*" the Lemma Match score is still **1**. This is because this measure matches the predicted lemma with the actual lemma irrespective of whether the new out-of-vocabulary lemma is attached to the synset *animal#n#1* or merged with it. This means that the lemma score for "*animal#n#1 attach*" result and "*animal#n#1 merge*" result would still be the 1 as long as the lemma in the actual result is in the synset *animal#n#1.* Lemma match would also consider the synonyms of a given actual lemma as correct hypernyms. So if the predicted result is "*creature#n#1 attach*" the Lemma Match score is **1** as lemma *creature#n#1* is a synonym for *animal#n#1* as they belong to the same synset. The Lemma Match score is **0** if the lemmas in the actual result and the predicted result are different. For example, if the result predicted by our system is "*animal#n#1 attach*" and the actual result is "*feline#n#1 attach*", the Lemma Match score is **0**. Even though the Wu & Palmer Similarity score for this pair is **0.7**, the Lemma Match score would still be **0**. This shows that Lemma Match measure assigns only full credit when correct synset is predicted. It does not assign any partial score, unlike the Wu & Palmer similarity score when the synsets are similar to one another but not the same.

4.1.3 Recall

The Recall measure scores the system based on number of results retrieved by the system compared to the total number of inputs given to the system. For the SemEval 2016 task, the test and the training data contain 600 and 400 out-of-vocabulary (OOV) input terms respectively. The recall value ranges from *0* to *1.* If our system could retrieve hypernyms for 484 OOV lemmas out of 600, the recall score is **0.8067**. If the system could retrieve hypernyms for all the 600 OOV terms, then the recall value is **1**.

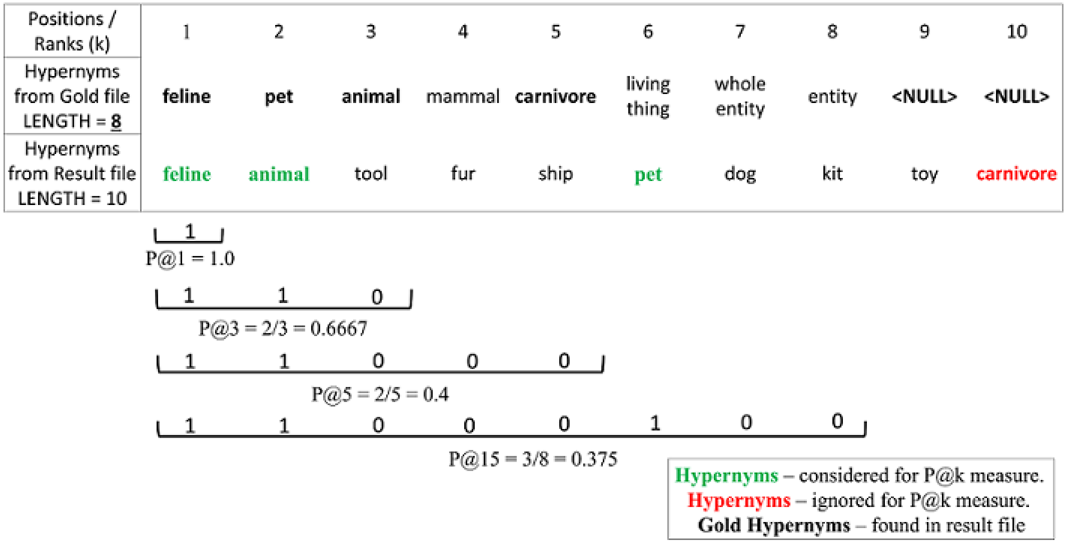
	Result Hypernyms in order	**Gold Hypernyms in order**	**Rank**	**Reciprocal Rank**	**MRR**
1	**feline**, animal, tool	feline, pet, animal	1	1	1
2	van, drive, **phone**	device, phone	2	0.5	0.5
3	fur, **animal**, tool, feline van, drive, **phone**	feline, pet, animal device, phone	2 3	0.5 0.333	0.4167

Table 4.1: Mean Reciprocal Rank scores with examples

4.1.4 Mean Reciprocal Rank (MRR) Score

The Mean Reciprocal Rank, also called MRR score is the primary evaluation measure for the SemEval 2018 Task 9 Hypernym Discovery. The reciprocal rank score for a given input term is the reciprocal of the rank of the first correctly retrieved hypernym from the top 15 candidate hypernyms predicted by our system. The maximum number of hypernyms that our system could predict as candidate hypernyms for any input term is 15. The hypernyms reported by our system should be listed in descending order of probability. The most probable candidate is reported as the first hypernym and the least probable candidate is reported as the last hypernym. The position of the first correct hypernym in the result list is assigned as the rank for our result and is used to calculate the reciprocal rank score. A hypernym in the result is a correct hypernym if it is present in the gold hypernym list. Table 4.1 shows some examples of reciprocal ranks and the resulting MRR score. From example 1, the hypernym *feline* is the first correct hypernym with position as **1** in the result hypernym list. So its reciprocal score is 11 which is **1**. Since there is only one result list, the mean reciprocal rank is also **1**. From example 2, the first correct hypernym is *phone* with rank **3** and both the reciprocal rank and mean reciprocal rank scores are **0.333**. In example 3, there are two lists of candidate hypernyms. The mean reciprocal rank in this case is the average of reciprocal ranks for hypernyms *animal* and *phone* which is **0.4167**.

4.1.5 Precision@k (P@k) Scores



Positions / Ranks (k)	1	2	3	4	5	6	7	8	9	10
Hypernyms from Gold file LENGTH = **8**	**feline**	**pet**	**animal**	mammal	**carnivore**	living thing	whole entity	entity	**<NULL>**	**<NULL>**
Hypernyms from Result file LENGTH = 10	feline	animal	tool	fur	ship	pet	dog	kit	toy	carnivore

Hypernyms – considered for P@k measure.
Hypernyms – ignored for P@k measure.
Gold Hypernyms – found in result file

Figure 4.2: Precisionk scores with an example

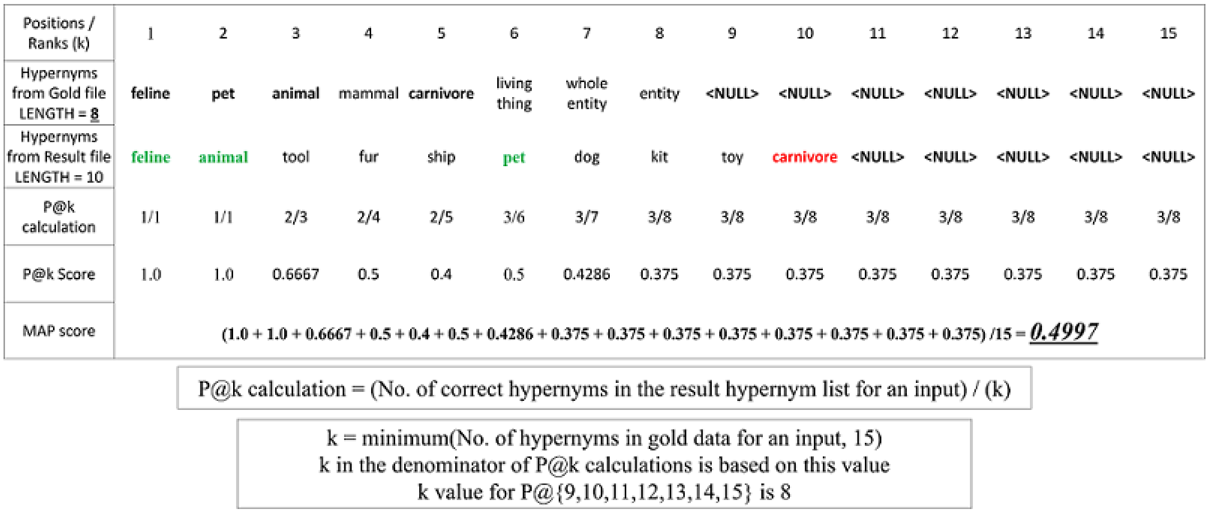
Precision@k is also called P@k measure and is an Information Retrieval measure which evaluates the precision of the top k results of a query. Precision@k score for an entry is the ratio of the number of correctly retrieved hypernyms in the first k results to the k value. This is given by formula

$$P@k = (Number\ of\ correct\ hypernyms\ in\ first\ k\ result\ hypernyms)/k$$

. The k value in the denominator is replaced by the minimum of two values the length of hypernym list in the gold data or the k value itself - for every entry. For example, let the number of hypernyms retrieved by our system for an entry be 10 and its respective gold hypernyms count be 8. To calculate P@15 score for this entry, the k value in the ratio $[(number\ of\ correct\ hypernyms\ in\ the\ result\ list)/(k = 15)]$ is changed from 15 to **8**. Figure 4.2 shows an example of P@1, 3, 5 and 15 scores. Here the number of hypernyms for one input term in the result file and the gold file are

10 and 8 respectively. The hypernym *carnivore* is ignored for the P@15 score as the k value in this score is changed from 15 to 8 and *carnivore* is present at location **10** which is not in the range [**1, 8**].

4.1.6 Mean Average Precision @15 (MAP) score



Positions / Ranks (k)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hypernyms from Gold file LENGTH = 8	feline	pet	animal	mammal	carnivore	living thing	whole entity	entity	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
Hypernyms from Result file LENGTH = 10	feline	animal	tool	fur	ship	pet	dog	kit	toy	carnivore	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
P@k calculation	1/1	1/1	2/3	2/4	2/5	3/6	3/7	3/8	3/8	3/8	3/8	3/8	3/8	3/8	3/8
P@k Score	1.0	1.0	0.6667	0.5	0.4	0.5	0.4286	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
MAP score	(1.0 + 1.0 + 0.6667 + 0.5 + 0.4 + 0.5 + 0.4286 + 0.375 + 0.375 + 0.375 + 0.375 + 0.375 + 0.375 + 0.375 + 0.375) /15 = ***0.4997***														

P@k calculation = (No. of correct hypernyms in the result hypernym list for an input) / (k)

k = minimum(No. of hypernyms in gold data for an input, 15)
k in the denominator of P@k calculations is based on this value
k value for P@{9,10,11,12,13,14,15} is 8

Figure 4.3: Mean Average Precision(@15) score with an example

The Mean Average Precision score is the main evaluation measure for the SemEval 2018 task 09 - Hypernym Discovery ranking. As the name indicates, for one input term, the average of all precision values from P@1 to P@15 is the Average Precision (AP) score. And the mean of AP scores of all entries in the givem sample is the Mean Average Precision score of that sample. Figure 4.3 shows a sample calculation of MAP@15 score for an input term. This average precision value is always computed over the number 15 irrespective of the number of hypernyms in gold data or the result data for any input term. For example, in the figure 4.3, the number of hypernyms of an entry in the gold data and the result data are 8 and 10 respectively. The task organizers chose to use all the 15-P@k scores ($1 \leq$ k $\geq$15) to compute the MAP score

Data Set	Wu & Palmer	Lemma Match	Recall
Training data	0.2179	0.0000	1.0000
Test data	0.2269	0.0000	1.0000

Table 4.2: The evaluation scores for the Random Baseline system

which is not the case with the other P@k dependent information retrieval measures.

4.2 Baseline Systems

Each SemEval task is provided with its own baseline system(s) and their respective scores for the training and test data. The following are the baseline systems used for the *SemEval 2016 Task 14 Semantic Taxonomy Enrichment*:

- The random sense baseline
- The first word first sense baseline
- The Default Hypernym Baseline

The first two baseline systems were proposed by the task organizers. We proposed the third baseline as part of our research. The evaluation scores from baseline systems act a lower bound for the evaluations scores of the participating systems.

4.2.1 The Random Sense Baseline

This baseline system selects a random synset from WordNet as the result of this system. If the new out-of-vocabulary lemma is a noun, then the random synset is chosen from the the noun synsets of WordNet. If it is verb, then the random synset is chosen from the verb synsets of WordNet. An operation is selected randomly from *attach* and *merge*. Table 4.2 shows the Random baseline scores for both the training and the test data. These scores are provided by the task organizers. WordNet has

Data Set	**Wu & Palmer**	**Lemma Match**	**Recall**
Training data	*0.4763*	*0.3250*	1.0000
Test data	*0.5140*	*0.4150*	1.0000

Table 4.3: The evaluation scores for the First-Word First-Sense Baseline system

82,115 noun synsets and **13,767** verb synsets. The probability of choosing a right sysnset for any given out-of-vocabulary(OOV) term is $1/82,115$ if the OOV lemma is a noun and is $1/13,767$ if it is a verb. And this probability is too low and hence the lemma match score is **0.0000** in both the cases.

4.2.2 The First-Word First-Sense Baseline

This baseline system selects the first word from the definition of the new out-of-vocabulary (OOV) lemma as the result. If the OOV lemma is a noun, then the first noun from the definition of the new OOV lemma is chosen as the result. Likewise, if the OOV lemma is a verb, then the first verb from the definition is chosen as the result. Here by *first word* we mean the first word other than the new OOV lemma. The first sense is assigned as the default sense for the selected word and *attach* is chosen as a default operation. Table 4.3 shows the scores of this baseline for both the training and the test data. These scores are also provided by the task organizers. This baseline has high Wu & Palmer similarity score and a high Lemma Match score when compared to the random baseline. This baseline is effective because the definitions of the new out-of-vocabulary terms contain the actual hypernym for this new lemma in itself. For example, if the new out-of-vocabulary term is *surya namaskar* and the definition provided with this term is *Surya namaskar is a yoga performed to attain a healthy life style*, then the hypernym predicted by this baseline is *yoga#n#1 attach*. It is important to consider that this definition holds the hypernym relation between *surya namaskar* and *yoga* with the ***is a*** Hearst Pattern. More information about

Data Set	**Wu & Palmer**	**Lemma Match**	**Recall**
Training data	*0.2495*	*0.0000*	1.0000
Test data	*0.2519*	*0.0000*	1.0000

Table 4.4: The evaluation scores for the Default Baseline system

other Hearst Patterns and its relation to hypernymy is described in the Background chapter (Chapter 2).

4.2.3 The Default Hypernym Baseline

This baseline system assigns *entity#n#1* as the default hypernym for a new out-of-vocabulary (OOV) lemma where the part-of-speech(POS) tag is *noun*. It assigns *be#v#1* as the default hypernym for a new OOV lemma where the POS tag is *verb*. Since we are assigning the default synset as a hypernym synset, the operation chosen here is *attach* by default. Table 4.4 shows the evaluation scores for this baseline system for both training and test data sets. This baseline similarity score is slightly higher than the random baseline system but lower than the first-word first-sense baseline system. The lemma match score for this system is **0** which is same as the random baseline. So this baseline as chosen to determine if a hypothesized strategy is a useful system to identify a hypernym or not.

4.3 Evaluation Scores of the Implemented Systems

We submitted two different systems to solve the problem of hypernym discovery for the two different SemEval tasks. In this section, a brief overview of the various sub-systems is presented. Detailed description of these systems is found in the Implementation (Chapter 3).

4.3.1 Evaluation of SemEval 2016 Task 14 : Semantic Taxonomy Enrichment system

For this task, we proposed a system which would identify an appropriate location to insert a new out-of-vocabulary (OOV) lemma in WordNet. The organizers provided a list of new OOV lemmas with their part of speech (POS) tags and definitions. By using the existing structure and information from WordNet, a hypernym synset or a synonym synset should be predicted by our system. For more information about the task description and the resources, please refer to Section 2.4 in Background Chapter 2.

The following are the sub-systems in the *SemEval 2016 Task 14 Semantic Taxonomy Enrichment* system. The Google News Vectors is the pre-trained word embedding matrix used by our sub-system. The UMBC Word Embedding and the WordNet Definition Embeddings are created as part of our research.

- **Definition Hearst Patterns** : A hypernym is obtained from the definition of the new out-of-vocabulary (OOV) term by applying the Identifying Hearst Patterns algorithm. These patterns fetch hypernyms only for noun OOV lemmas. The complete Description of this sub-system is available in Section 3.2.5 in Implementation Chapter 3.

- **UMBC Word Embedding** : A bag of words (CBOW) embedding is learnt from the UMBC corpus. The words with a noun part-of-speech (POS) tag from the definition of the new OOV noun lemma are used to fetch a hypernym from this UMBC embedding. Similarly, words with a verb POS tag are used to fetch a hypernym for a verb OOV lemma using this embedding. The complete Description of this sub-system is available in Section 3.2.6 in Implementation Chapter 3.

- **Google News Vectors** : The words with a noun part-of-speech (POS) tag from the definition of the new OOV noun lemma are used to fetch a hypernym from the Google News embedding. Similarly words with a verb POS tag are used to fetch a hypernym for a verb OOV lemma using this embedding. The complete Description of this sub-system is available in Section 3.2.6 in Implementation Chapter 3.

- **UMBC IS-A Hearst Patterns** : The IS-A Hearst Patterns from the UMBC corpus are pre-fetched and stored in a list in the $\langle$*Hyponym*$\rangle$ *:* $\langle$*Hypernym*$\rangle$ format. If the new OOV term is present in the $\langle$*Hyponym*$\rangle$ part of the list, then this is added to a temporary result list. Once all the candidate Hypernyms are identified, then the most frequent $\langle$*Hypernym*$\rangle$ in the temporary list is reported as the hypernym for this term. This sub-system also fetches hypernyms only for noun OOV lemma. The complete Description of this sub-system is available in Section 3.2.2 in Implementation Chapter 3.

- **WordNet Definition Embeddings** : Two bag of words (CBOW) embeddings are learnt from the definitions of nouns and the definitions of verbs from WordNet. We call them Noun WordNet Definition Embedding and Verb WordNet Definition Embedding. The words with a noun part-of-speech (POS) tag from the definition of the new OOV noun lemma are used to fetch a hypernym from the Noun WordNet Definition Embedding. Similarly, words with a verb POS tag from the definition of a new verb OOV lemma are used to fetch a hypernym from the Verb WordNet Definition Embedding. Complete Description of this sub-system is available in Section 3.2.6 in Implementation Chapter 3.

If all these sub-systems fail to fetch a hypernym for a new OOV lemma, a default lemma is added a hypernym. The default hypernym synsets chosen for noun and the verb OOV lemmas are *entity#n#1* and *be#v#1* respectively. This is done to improve the coverage of our sub-systems as well as maintain the recall value at **1.0**. The default root baseline system adds *entity#n#1* as default hypernym for all noun OOV lemmas and adds *be#n#1* for all verb OOV lemmas. Since this Default baseline system's Wu&Palmer similarity score is higher than the random baseline, adding default hypernyms for the new OOV lemmas where these sub-systems fail would still give a higher similarity score than the random baseline. The final system for this task merges the results from all the above modules without the default values (refer Section 3.4.1). After merging the results, the remaining OOV lemmas with no hypernyms are assigned the default values. We call this system - Babbage with default senses. Once the results from the sub-systems are merged, these hypernyms are run through the *Assign Sense module* (refer Section 3.3). This module re-assigns an appropriate sense for the chosen hypernym with the default sense*#1*. We call these results the Babbage results with re-assigned senses.

Table 4.5 and Table 4.6 show the results of all these sub-systems and the final system. The *With Default Hypernyms* column from these tables shows the scores of the sub-systems where the default hypernyms *entity#n#1* or *be#v#1* were added in case of a failed scenario. These tables also show the evaluation scores for these systems without the default hypernym values where the recall value is compromised. The individual baseline scores with respect to nouns and verbs are also listed in these tables.

From Table 4.5, the following observations can be made:

- By using the Wu&Palmer Similarly scores and recall values of the sub-systems (without default hypernyms), we create an optimal order to merge the results of these sub-systems to form a final system.The order of ranking the sub-system is

	Type	**With Default Hypernyms**			**Without Default Hypernyms**		
system or sub-system	**All, noun, verb**	**Wu & Palmer Score**	**Lemma Match Score**	**Recall**	**Wu & Palmer Score**	**Lemma Match Score**	**Recall**
Definition Hearst Patterns	all	0.4818	0.2600	1.0000	0.5594	0.3636	0.7150
	noun	0.5030	0.298	1.0000	**0.5594**	**0.3636**	**0.8194**
	verb	***0.3365***	0.0000	1.0000	**0.0000**	**0.0000**	*0.0000*
UMBC Word Embedding	all	0.2938	0.0200	1.0000	0.3675	0.0584	0.3425
	noun	0.2881	0.0229	1.0000	0.3717	0.0601	0.3810
	verb	0.3325	0.0000	1.0000	0.2279	0.0000	0.0784
Google News Vectors	all	0.2885	0.0225	1.0000	0.3757	0.0789	0.2850
	noun	0.2837	0.0258	1.0000	0.3818	0.0849	0.3037
	verb	0.3218	0.0000	1.0000	*0.3040*	0.0000	0.1569
UMBC IS-A Hearst Patterns	all	0.2686	0.0050	1.0000	0.3069	0.0185	0.2700
	noun	0.2587	0.0057	1.0000	0.3076	0.0185	0.3095
	verb	**0.3365**	0.0000	1.0000	*0.0000*	*0.0000*	*0.0000*
WordNet Definition Embeddings	all	0.2496	0.0000	1.0000	0.2477	0.0000	*0.5050*
	noun	0.2403	0.0000	1.0000	0.2481	0.0000	0.5358
	verb	0.3134	0.0000	1.0000	0.2422	0.0000	*0.2941*
Babbage Default Senses	all	0.4858	0.2625	1.0000	0.5196	0.3079	0.8525
	noun	0.5125	0.3009	1.0000	**0.5369**	0.3281	0.9169
	verb	0.3037	0.0000	1.0000	***0.2549***	0.0000	0.4118
Babbage Re-assign Senses	all	0.4821	0.2650	1.0000	0.5153	0.3108	0.8525
	noun	0.5088	0.3037	1.0000	**0.5329**	0.3312	0.9169
	verb	0.3000	0.0000	1.0000	***0.2459***	0.0000	0.4117
Random Baseline				all	0.2179	0.0000	1.0000
				noun	0.2148	0.0000	1.0000
				verb	0.2390	0.0000	1.0000
Default Baseline				all	0.2495	0.0000	1.0000
				noun	0.2368	0.0000	1.0000
				verb	0.3365	0.0000	1.0000
First-word First-sense Baseline				all	0.4763	0.3250	1.0000
				noun	0.4900	0.4150	1.0000
				verb	0.3824	0.4150	1.0000

Table 4.5: The evaluation scores for the SemEval 2016 task 14 - Taxonomy Enrichment systems against training data All the scores in this table are rounded to the 10,000th decimal place.

as follows: Definition Hearst Patterns, UMBC Word Embedding, Google News Vectors, UMBC IS-A Hearst Patterns and WordNet Definition Embeddings. If the result hypernyms for some new OOV lemma could not be identified by using any of these sub-systems, then the default value *entity#n#1* for noun OOV lemma and *be#v#1* for verb OOV lemma are assigned.

- The Wu&Palmer Similarly score and the Lemma Match score for the Definition Hearst Pattern module is higher than all three baselines. This indicates that hypernym discovery using Hearst Patterns over Definitions exceed the performance obtained from all other sub-systems especially when the new out-of-vocabulary term is provided along with its definition.

- Both the Definition Hearst Pattern and the UMBC IS-A Hearst Patterns sub-systems work only for the new OOV lemma with a noun part-of-speech tag. All the evaluation scores for a verb OOV lemmas are **0.0000**. We identified all possible Hearst Patterns by relying on the part-of-speech tags to include bi-gram and tri-gram phases. Since the original Hearst Patterns [Hearst 1992] were proposed only for Noun Phrases, we considered only the noun part-of-speech tag for pattern recognition. Hence this sub-system could not fetch results for verb OOV lemmas.

- Hypernym discovery for a verb OOV lemma using Google News Vectors has the best similarity score. So this module is included to fetch hypernyms for the new OOV lemma with verb as its POS tag.

- The final sub-system - discovering hypernyms using the WordNet Definition Embeddings - has the highest recall score. Though it retrieves more results, its similarity score is lower than the Random and the First-word First-sense

Baselines. This module is included in the final system because it was boosting the recall score of the entire system.

- After re-assigning more precise sense to some hypernyms with default sense, the similarity score is reduced by a very minute percentage ($< 0.5\%$). But the Lemma Match score was increased by a small percentage.

- From *Figure 4.7*, we noticed that for a few new OOV terms, more that one sub-system was able to fetch hypernyms from WordNet. On the other hand, hypernyms for some new OOV lemmas were retrieved by only one sub-system. This indicates that merging the results from various sub-systems would improve the overall recall of our proposed system. For example from *Figure 4.4 - b. Coverage of all sub-systems*, hypernyms for **89** OOV lemma could be retrieved by using the Hearst Pattern modules and the word embedding modules. But for the other **87** OOV lemmas only the Definition Hearst Pattern module was able to retrieve hypernyms.

- From *Figure 4.4* and *Figure 4.5*, we can see that all the results obtained from the Definition Hearst Patterns and the UMBC IS-A Hearst Patterns sub-systems were only nouns. Hence the evaluation scores of both these sub-systems for verbs with no default hypernyms are **0s** [Table 4.5]. Therefore, the hypernyms for verbs were retrieved from only the sub-systems which rely on the word embeddings.

- From *Figure 4.6* and Table 4.5, the following two unique observations could be made:
  - Most of the results for verb OOV lemmas from the training sample come from the WordNet Definition Embeddings. So this sub-system has the

highest recall value. The similarity score for this module is **0.2422** and this score is higher than the Random baseline score *0.239* but lower than and Default baseline score *0.3365* and the First-word, First-sense baseline score *0.3824*. Though this module fetches more hypernyms for the verb OOV lemmas when compared to the other modules, it cannot be ranked higher in the merge order as the hypernyms fetched are not very close to the original hypernyms or the hypernyms fetched from other sub-systems.

– The recall value for the sub-system Google News Vectors is lower than the WordNet Definition Embedding sub-system. The similarity score of this sub-system (**0.3040**) is higher than the Random baseline score (*0.239*) but not higher than the other baseline scores of Default baseline (*0.3365*) and First-word First-sense baseline (**0.3824**). This similarity score for verbs from this sub-system is higher than the other sub-systems. If we ignore the drop in the recall value and only consider the precision of individual results, this sub-system could be given a higher priority in the merge algorithm.

The Venn diagrams in Figures 4.4, 4.5 and 4.6 represent the recall coverage of the sub-systems of this task when applied on the training data. The training data consists of **400** new out-of-vocabulary lemmas out of which **349** are *Nouns* and **51** are *Verbs*.

The Venn diagrams Figures 4.7, Figure 4.8 and Figure 4.9 represent the recall coverage of the sub-systems of this task when applied on the test data. The test data consists of **600** new out-of-vocabulary lemmas out of which **517** are *Nouns* and **83** are *Verbs*.

Table 4.6 shows the results obtained after applying our system to the test data. The table also shows the breakdown of scores for various sub-systems and their in-

	Type	**With Default Hypernyms**			**Without Default Hypernyms**		
system or sub-system	**All, noun, verb**	**Wu & Palmer Score**	**Lemma Match Score**	**Recall**	**Wu & Palmer Score**	**Lemma Match Score**	**Recall**
Definition Hearst Patterns	all	0.4730	0.2883	1.0000	0.5682	0.4336	0.4336
	noun	0.4932	0.3346	1.0000	0.5682	0.4336	0.7717
	verb	***0.3473***	0.0000	1.0000	**0.0000**	**0.0000**	**0.0000**
UMBC Word Embedding	all	0.3026	0.0133	1.0000	0.3814	0.0370	0.3600
	noun	*0.2950*	0.0135	1.0000	0.3821	0.0265	0.4004
	verb	*0.3500*	0.0120	1.0000	0.3641	0.1111	0.1084
Google News Vectors	all	0.2847	0.0167	1.0000	0.3615	0.0629	0.2650
	noun	0.2721	0.0155	1.0000	0.3549	0.0556	0.2785
	verb	0.363	0.0241	1.0000	**0.4252**	0.1333	0.1807
UMBC IS-A Hearst Patterns	all	0.273	0.0100	1.0000	0.3194	0.0426	0.2350
	noun	0.2610	0.0116	1.0000	0.3194	0.0426	0.2727
	verb	***0.3473***	0.0000	1.0000	**0.0000**	**0.0000**	**0.0000**
WordNet Definition Embeddings	all	0.2500	0.0033	1.0000	*0.2393*	0.0061	**0.5433**
	noun	0.2383	0.0019	1.0000	0.2371	0.0033	**0.5783**
	verb	0.3228	0.0120	1.0000	0.2632	0.0370	**0.3253**
Babbage Default Senses	all	0.4821	0.2917	1.0000	0.5192	0.3500	0.8333
	noun	0.5055	0.3346	1.0000	**0.5355**	0.3728	0.8974
	verb	0.3366	0.0241	1.0000	***0.3102***	0.0556	0.4337
Babbage Re-assign Senses	all	0.4722	0.2917	1.0000	0.5073	0.3500	0.8333
	noun	0.4948	0.3346	1.0000	**0.5236**	0.3728	0.8975
	verb	0.3309	0.0240	1.0000	***0.2971***	0.0556	0.4337
Random Baseline				all	0.2269	0.0000	1.0000
				noun	0.2232	0.0000	1.0000
				verb	0.2496	0.0000	1.0000
Default Baseline				all	0.2519	0.0000	1.0000
				noun	0.2366	0.0000	1.0000
				verb	0.3473	0.0000	1.0000
First-word First-sense Baseline				all	0.5140	0.4150	1.0000
				noun	0.5273	0.3907	1.0000
				verb	0.4310	0.5662	1.0000

Table 4.6: The evaluation scores for the SemEval 2016 task 14 - Taxonomy Enrichment systems against test data. All the scores in this table are rounded to the 10,000th decimal place

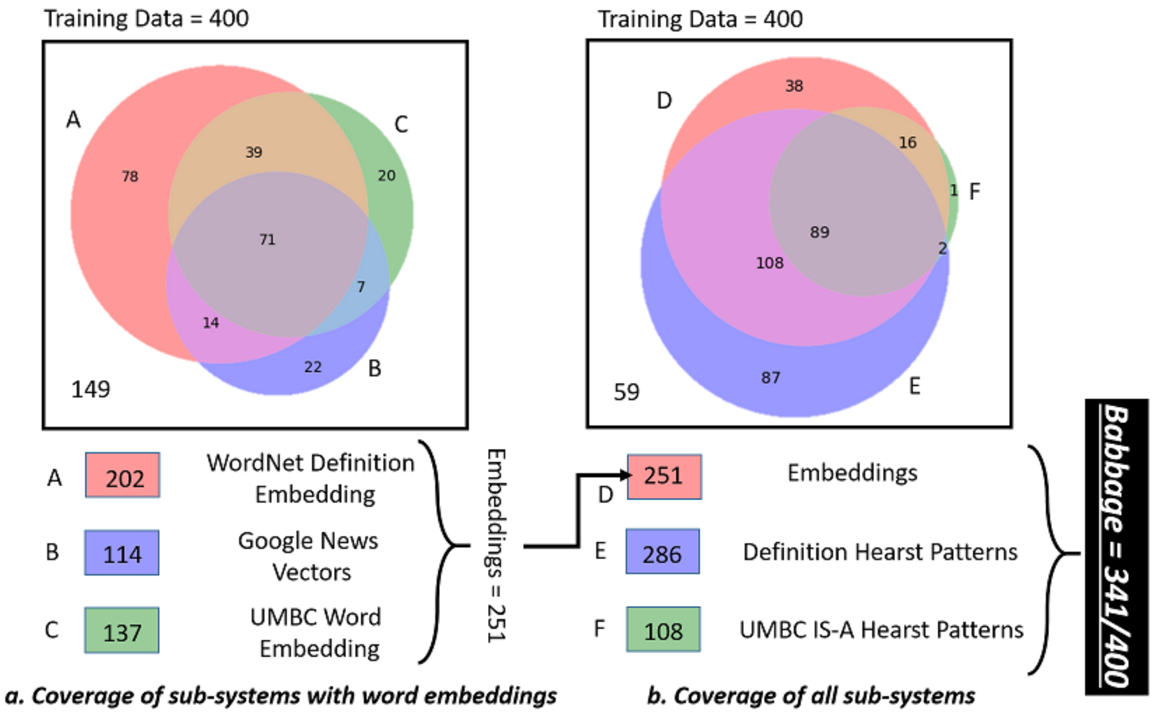
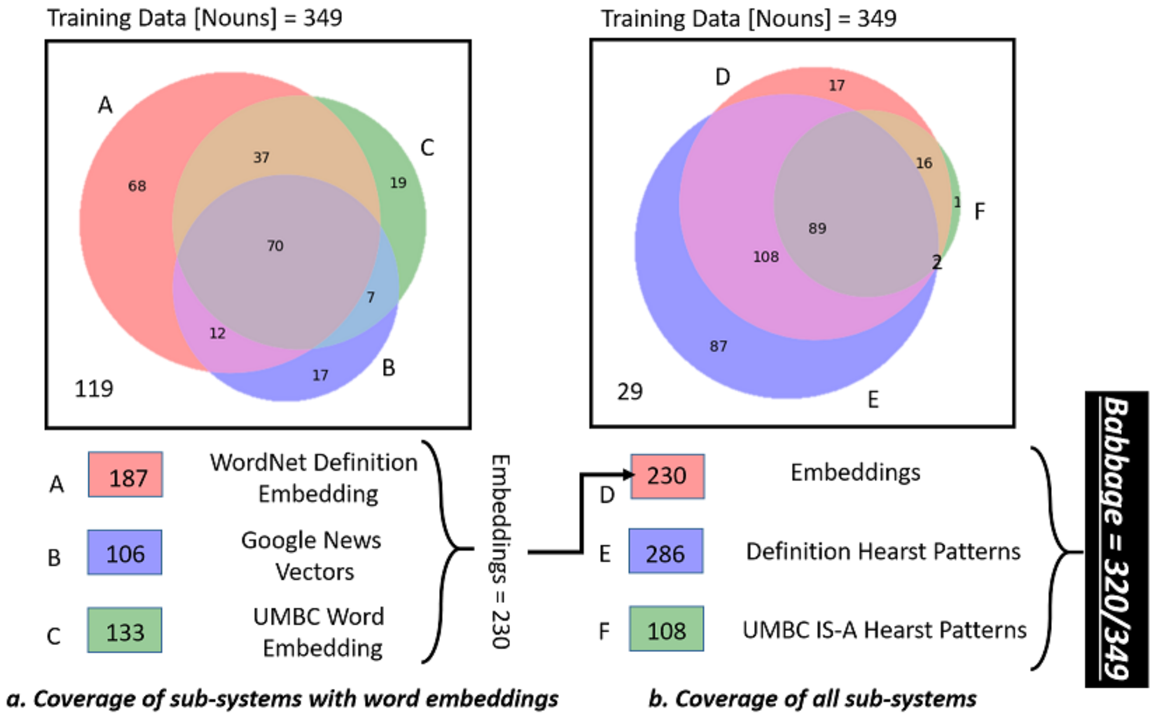


Figure 4.4: Venn Diagrams representing the recall values of the sub-systems for SemEval 2016 Task 14 Taxonomy Enrichment. The image represents the recall values for the entire training sample including nouns and verbs.

dependent nouns and verbs lemmas. Most of the observations made on the results obtained from the training data are applicable for the results of the test set data as well. The following observations can be made from this table.

- The Wu&Palmer Similarly score and the Lemma Match score for the Definition Hearst Pattern module is higher than all the three baselines even for the test data. This indicates that our decision to rank results of this Definition Hearst Patterns sub-system high based on the training data worked even for the test data.

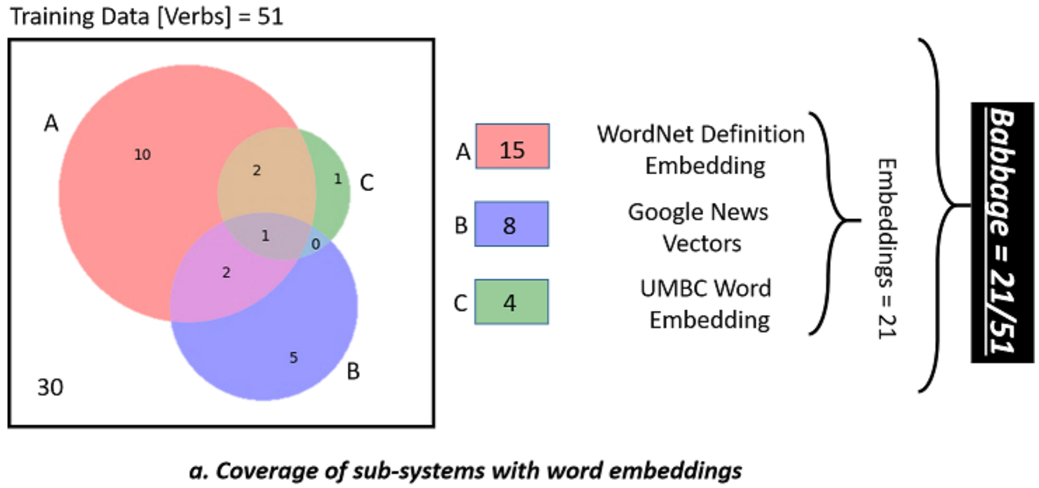
- The order of ranking the results from the sub-systems determined by looking at the training data worked well even for the test data. Higher ranked sub-systems


Training Data [Nouns] = 349
A
C
B
68
37
19
70
7
12
17
119
A 187 WordNet Definition Embedding
B 106 Google News Vectors
C 133 UMBC Word Embedding
Embeddings = 230
Training Data [Nouns] = 349
D
17
16
1
F
89
108
2
87
29
E
D 230 Embeddings
E 286 Definition Hearst Patterns
F 108 UMBC IS-A Hearst Patterns
Babbage = 320/349

a. Coverage of sub-systems with word embeddings

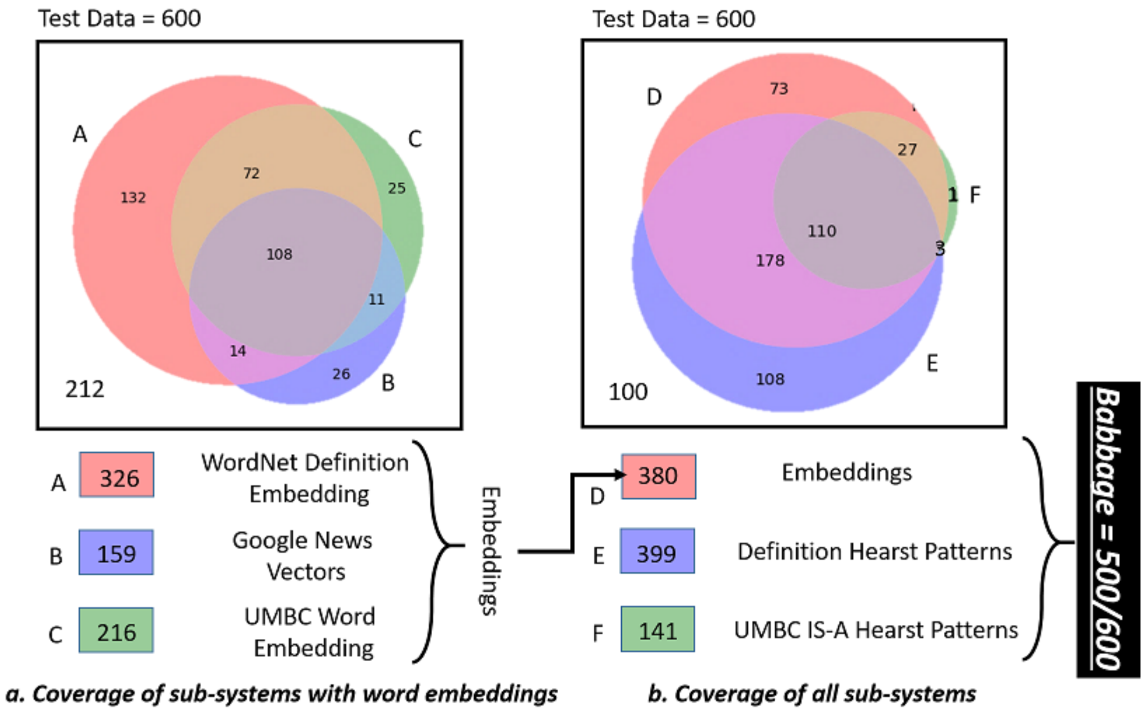
b. Coverage of all sub-systems

Figure 4.5: Venn Diagrams representing the recall values of the sub-systems for SemEval 2016 Task 14 Taxonomy Enrichment. The image represents the recall values for only the NOUN part-of-speech tag input lemmas from the training sample.



a. Coverage of sub-systems with word embeddings

Figure 4.6: Venn Diagrams representing the recall values of the sub-systems for SemEval 2016 Task 14 Taxonomy Enrichment. The image represents the recall values for only the VERB part-of-speech tag input lemmas from the training sample.


Test Data = 600
A
C
132
72
25
108
11
14
26
B
212
Test Data = 600
D
73
27
1 F
110
178
3
E
108
100
A 326 WordNet Definition Embedding
B 159 Google News Vectors
C 216 UMBC Word Embedding
Embeddings
D 380 Embeddings
E 399 Definition Hearst Patterns
F 141 UMBC IS-A Hearst Patterns
Babage = 500/600

a. Coverage of sub-systems with word embeddings ***b. Coverage of all sub-systems***

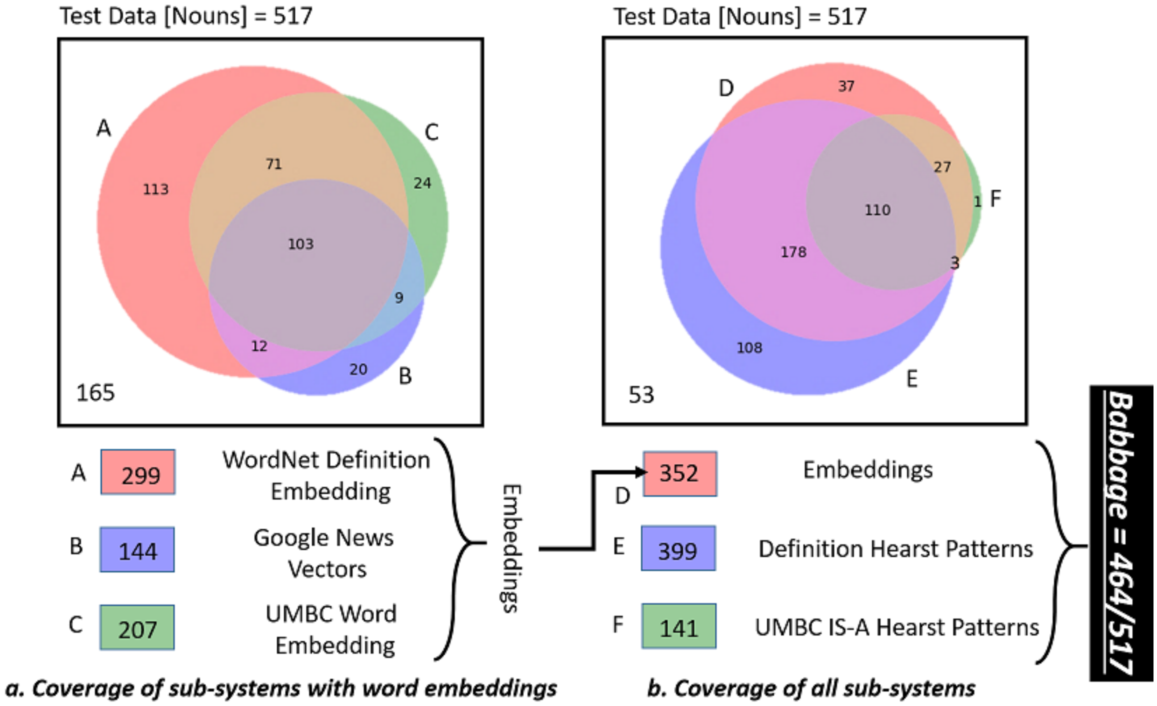
Figure 4.7: Venn Diagrams representing the recall values of the sub-systems for SemEval 2016 Task 14 Taxonomy Enrichment. The image represents the recall values for the entire test sample including nouns and verbs.

obtain higher similarity scores for the test data (Definition Hearst Patterns -*0.5682*, UMBC Word Embedding -*0.3814*, Google News Vectors -*0.3615*, UMBC IS-A Hearst Patterns -*0.3194* and WordNet Definition Embeddings -*0.2393*)

- Both the Definition Hearst Patterns and the UMBC IS-A Hearst Patterns sub-systems work only for the new OOV lemma with a noun part-of-speech tag. All the evaluation scores for verb OOV lemma are **0.0000**.

- Hypernym discovery for a verb OOV lemma using Google News Vectors has the best similarity score even for the test data.

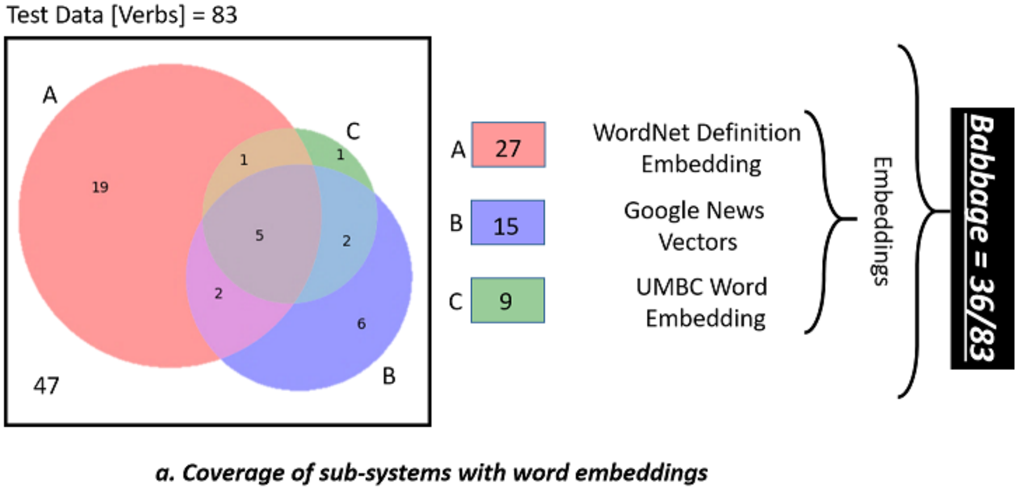
- The final sub-system - discovering hypernyms using the WordNet Definition Embeddings - has the highest recall score even for the test data. Though it



a. Coverage of sub-systems with word embeddings

b. Coverage of all sub-systems

Figure 4.8: Venn Diagrams representing the recall values of the sub-systems for SemEval 2016 Task 14 Taxonomy Enrichment. The image represents the recall values for only the NOUN part-of-speech tag input lemmas from the test sample.



a. Coverage of sub-systems with word embeddings

Figure 4.9: Venn Diagrams representing the recall values of the sub-systems for SemEval 2016 Task 14 Taxonomy Enrichment. The image represents the recall values for only the VERB part-of-speech tag input lemmas from the test sample.

retrieves more results, the similarity score is lower than the Random and the First-word First-sense Baselines.

- The Venn diagrams representing the recall coverage for the test data (4.7, 4.8 and 4.9) are similar to the Venn diagrams we obtained for the training data (4.4, 4.5 and 4.6).

- From *Table 4.6 and Figure 4.9*, the following two unique observations could be made:

  - The Definition Hearst Patterns have uniquely identified hypernyms for **108** OOV noun lemmas out of **600** test data lemmas. They have achieved an overall recall of **399/517** (or 0.7718) for the noun OOV lemmas from the entire test set.

  - The recall score for the WordNet definition subsystem is close to **50%**. The majority of the results for verb OOV lemmas of the test set also come from this module. But the similarity score of this module is lower than the default and the First-Word, First-Sense baseline.

  - The recall value for the Google News vectors system is close to half the recall score of WordNet Definition Embedding system. Its similarity score **0.4252** is higher than the Random baseline score *0.2496* and the default baseline *0.3473* but slightly lower than the First-word, First-sense baseline **0.4310**. This similarity score for the verb OOV lemmas is also higher than the scores of these verbs from other modules. This signifies that the results from this module should be ranked higher than the results from the WordNet Definition Embeddings. This module performed better for the test data verbs than the training data.

4.3.2 Evaluation of SemEval 2018 Task 09: Hypernym Discovery system

For this task, we proposed a system which identifies a set of hypernyms for a given input term by looking into a huge text based corpus. The organizers provided a list of input terms and the type of the input term (Concept or Entity). They also provided the participants with a huge text based part-of-speech tagged English corpus - UMBC WebBase Corpus. For more information about the task description and the resources, please refer to the Background Chapter (Chapter 2).

The following are the sub-systems in the SemEval 2018 Task 9 - Hypernym Discovery system. All these sub-systems are built based on UMBC WebBase Corpus only. No additional resources are used apart from those provided with this task. Each module reports a maximum of 15 results for each input term. In the following modules, when we mention the *top results*, the count should always be assumed to be a maximum count of 15 values.

- **IS-A Hearst Pattern** : The Hearst Pattern **Hyponym Noun Phrase *is (a | an | the)* Hypernym Noun Phrase** is used as the pattern of interest. All the phrases in UMBC Corpus matching this pattern are fetched and stored in a list of the form ⟨*Hyponym*⟩ *:* ⟨*Hypernym*⟩. If the target term is present in the ⟨*Hyponym*⟩ part of the list, then the ⟨Hypernym⟩ part is added to a temporary result list. This temporary result list can have duplicate values. Once all the candidate Hypernyms are identified, then this list is converted into a set ordered by the most frequent hypernyms to the least frequent ones in this list. Then the top hypernyms in this sorted set are reported as the result for this input term [Hearst 1992].

- **Co-occurrence frequencies over Normalized Corpus** : The huge 28.3 GB

UMBC corpus is normalized by removing all the words with a part of speech tag other than noun or noun phrases and is reduced to 17 GB. This sub-system iterates through the entire new corpus searching for each input term and all the nouns which co-occur with this input term are added to a temporary list. This list could have duplicates. Once the entire corpus is searched, this list is sorted in the descending order of the frequencies of hypernyms in this list. The top hypernyms in this final set are reported as the candidate hypernyms for the given input terms.

- **Co-occurrence frequencies over Hearst Patterns** : The Hearst Patterns mentioned in the Section 2.5.3 are the patterns of interest. The entire 28.3 GB UMBC corpus is filtered for phrases matching these patterns. The patterns found are stored in a file in the format *hypernym : hyponym-1 , hyponym-2, . . . , hyponym-n*. The total size of this pattern file is 180 MB (with **4,055,917** lines). If the target term is present in this *Hyponym part "hyponym-1 , hyponym-2, . . . , hyponym-n"* of the list, then *hypernym* part of this list is added to a temporary result list. This temporary result list can have duplicate values. Once all the candidate hypernyms are identified, then this list is converted into a set ordered by most to least frequent hypernyms in this list. Then the top hypernyms in this sorted set are reported as the result for this input term [Hearst 1992].

- **Applying Word Similarity to Word Embedding** : The huge 28.3 GB UMBC corpus is normalized by removing all the words with a part of speech tag other than noun or noun phrases and is reduced to 17 GB. A word embedding matrix is created over this Normalized Corpus. TensorFlow's word2vec model [Mikolov, Chen, Corrado, and Dean 2013] is used to build this embedding. The specifications of the model are as follows:

1. **Model :** ***Continuous Bag of Words (CBOW)*** - the vector values of all the words in the context window are modified based on the vector value of the word of interest in this context.

2. **Window Size :** ***10***. The context window size for a term which determines its embedding score. A maximum of 10 words including the target word are considered to calculate the vector score of a context word.

3. **Minimum Frequency Count :** ***5***. If a term occurs less than this number of times in the entire UMBC Corpus, the vector for this term is deleted from the embedding.

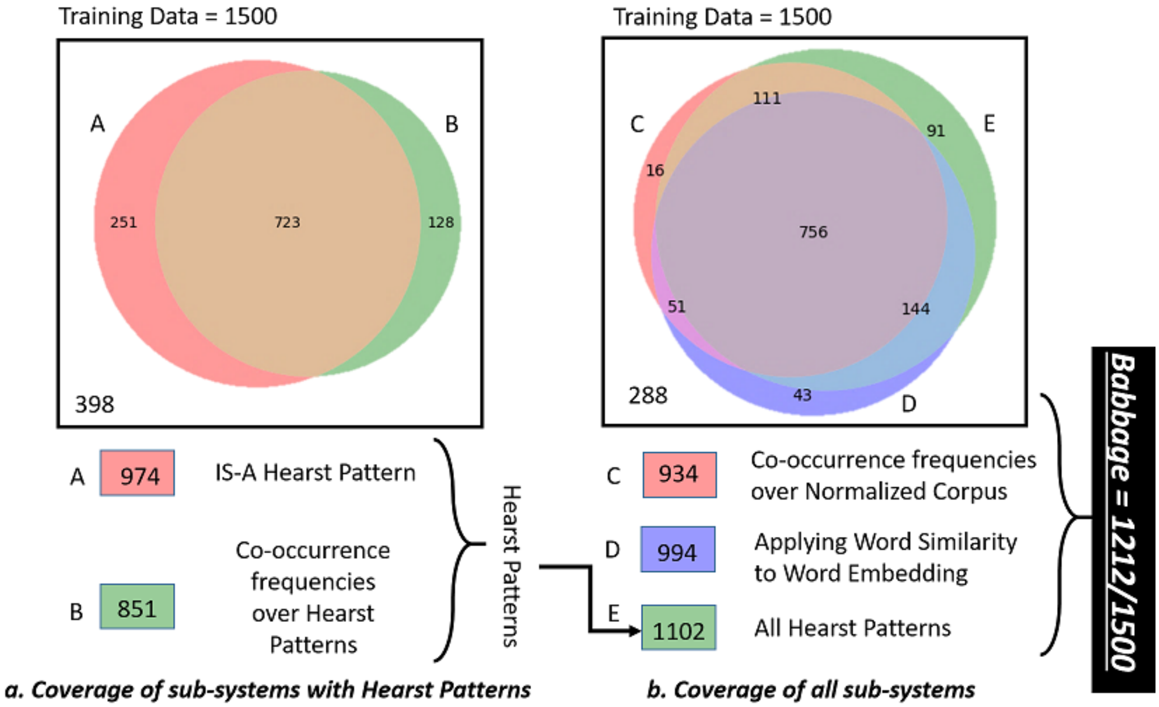
4. **Dimension Size** : ***300***.

More information about these specifications can be found in *Background Chapter*(Chapter 2). Once the embedding matrix is learnt, we used the training data hypernym-hyponyms pairs as seed values and find the most probable distance Φ^* which might define the distance between a general hypernym-hyponym pair in this UMBC word embedding. This value is found by using Formula 4.2. Here Φ is computed by using the equation y = Φ x. This Φ^* value is used to get candidate hypernyms from the UMBC word embedding matrix for any given input term. These candidate hypernyms are the terms on either side of the input term which lie at a distance of Φ^* in the UMBC embedding. The top hypernyms from these candidate hypernyms are listed as the result hypernyms from this sub-system. [Fu, Guo, Qin, Che, Wang, and Liu 2014]

$$\Phi^* = \text{argmin}_{\Phi} \frac{1}{N} \sum_{(x,y)} \|\Phi x - y\|^2 \tag{4.2}$$

The final system - Babbage merges the results from all the above sub-systems to get the final result. This is done to improve the recall of the system.

The Venn diagrams in Figures 4.10, 4.11 and 4.12 represent the recall coverage of the sub-systems of this task when applied on the training data. The training data consists of **1500** input terms with ***979*** *Concepts* and ***521*** *Entities*.


Training Data = 1500
A
B
251
723
128
398
A 974 IS-A Hearst Pattern
B 851 Co-occurrence frequencies over Hearst Patterns
Hearst Patterns
Training Data = 1500
C
E
111
91
16
756
51
144
43
288
D
C 934 Co-occurrence frequencies over Normalized Corpus
D 994 Applying Word Similarity to Word Embedding
E 1102 All Hearst Patterns
Babbage = 1212/1500

a. Coverage of sub-systems with Hearst Patterns

b. Coverage of all sub-systems

Figure 4.10: Venn Diagrams representing the recall values of the sub-systems for SemEval 2018 Task 9 Hypernym Discovery. The image represents the recall values for the entire training sample including concepts and entities.

Table 4.8 shows the results obtained after applying our **Babbage** system to the training data. Table 4.7 shows the breakdown of scores for various sub-systems and scores with respect to concepts and entities from this training data. The following are the key observations made from these tables:

- The recall values of these sub-systems (**<u>AR</u>**) are used to decide the merge order of these system to form system Babbage. So **5** values from each of the result files of subsystems -*IS-A Hearst Pattern (first)*, *Co-occurrence frequencies over*

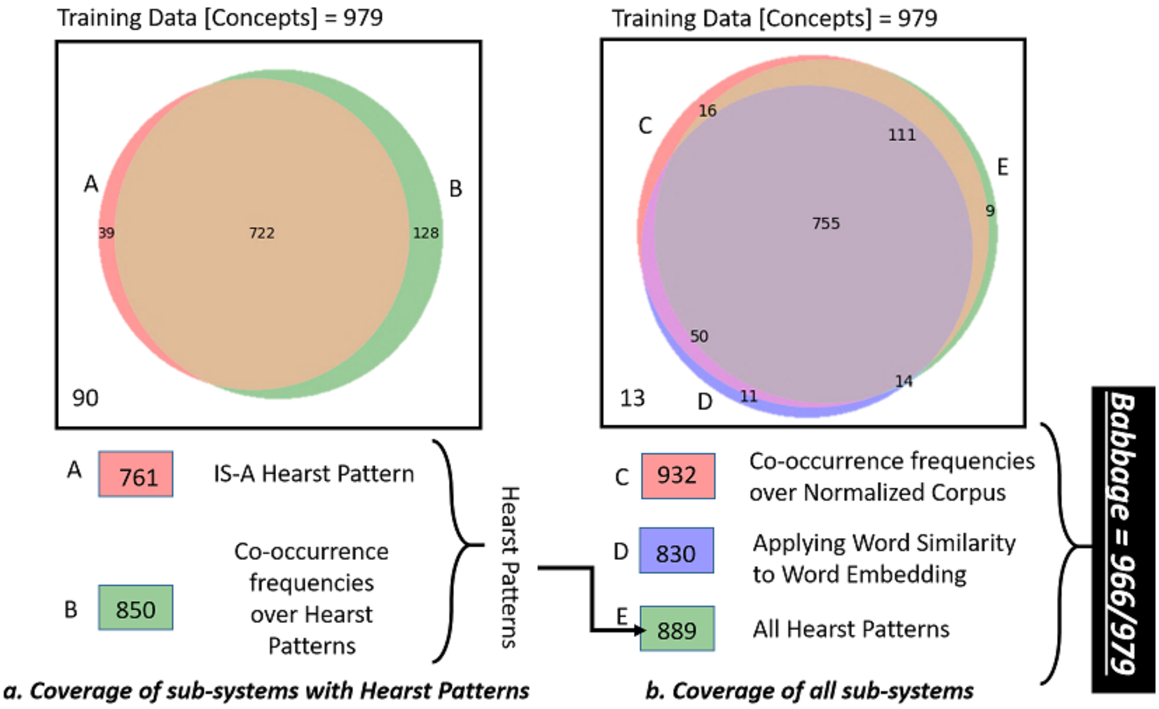

Training Data [Concepts] = 979
A
B
39
722
128
90
A 761 IS-A Hearst Pattern
B 850 Co-occurrence frequencies over Hearst Patterns
Hearst Patterns
a. Coverage of sub-systems with Hearst Patterns
Training Data [Concepts] = 979
C
16
111
E
9
755
50
14
13
D
11
C 932 Co-occurrence frequencies over Normalized Corpus
D 830 Applying Word Similarity to Word Embedding
E 889 All Hearst Patterns
b. Coverage of all sub-systems
Babbage = 966/979

Figure 4.11: Venn Diagrams representing the recall values of the sub-systems for SemEval 2018 Task 9 Hypernym Discovery. The image represents the recall values for only the concept input terms from the training sample.

Normalized Corpus + Hearst Patterns (second, third) and Applying Word Similarity to Word Embedding(last)- are merged till a maximum of **15** hypernyms are obtained .

- There are two strategies of merging the results:

   1. The first strategy is the one used by our system 4.3.2.

   2. In the second strategy, we simply choose all the hypernyms from first sub-system followed by the hypernyms from the second, third and the last sub-system.

The scores obtained by strategy **2** were slightly lower than the one's from strategy **1**.

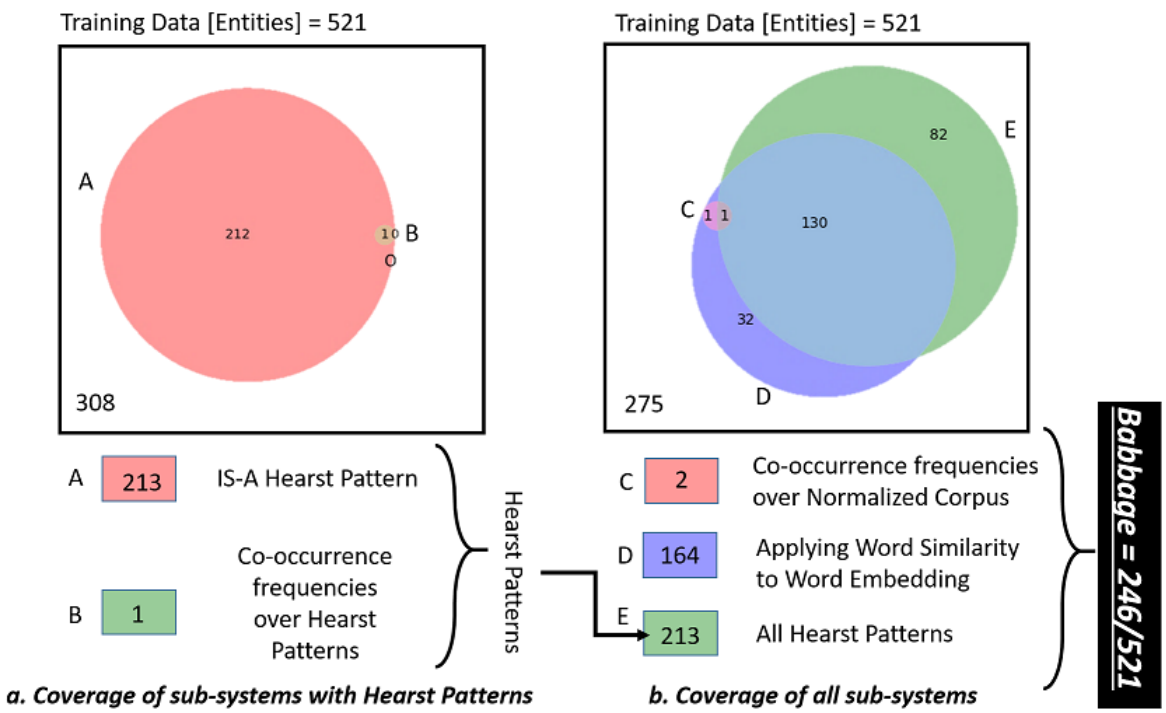

Training Data [Entities] = 521
A
212
1 0 B
0
308
Training Data [Entities] = 521
82
E
C
1 1
130
32
275
D
A
213
IS-A Hearst Pattern
B
1
Co-occurrence frequencies over Hearst Patterns
Hearst Patterns
C
2
Co-occurrence frequencies over Normalized Corpus
D
164
Applying Word Similarity to Word Embedding
E
213
All Hearst Patterns
Babbage = 246/521
a. Coverage of sub-systems with Hearst Patterns
b. Coverage of all sub-systems

Figure 4.12: Venn Diagrams representing the recall values of the sub-systems for SemEval 2018 Task 9 Hypernym Discovery. The image represents the recall values for only the entity input terms from the training sample.

- The sub-systems Co-occurrence frequencies over *Normalized Corpus* and *Hearst Patterns* were able to fetch results only for *concepts*. The evaluation scores for training data's entities from these modules are all *0's*. The recall values for training data entities for these modules are *0.0019* and *0.0038* (*1* and *2* out of *521* input entity terms).

- The *IS-A Hearst Pattern* module performs the best for entities both in terms of recall and the other evaluation scores. Though the recall value for the *Word Embedding* module is good, the other evaluation scores are not good.

- The *P1* score of Babbage system for entities is **0.1788**. And this is the best value for this measure. This signifies that for at least *one-fifth* of the entity

System Name	Type A, C, E	MRR	MAP	P@1	P@3	P@5	P@15	Recall
IS-A Hearst Pattern	A	0.0863	0.0329	0.0573	0.0427	0.0332	0.0247	1.0000
	AR	**0.1330**	0.0507	0.0882	0.0658	0.0512	0.0380	0.6273
	C	0.0783	0.0330	0.0429	0.0422	0.0338	0.0271	1.0000
	CR	0.1007	0.0424	0.0551	0.0543	0.0436	0.0348	0.7773
	E	0.1014	0.0328	0.0844	0.0438	0.0321	0.0201	1.0000
	ER	***0.2482***	0.0804	0.2065	0.1071	0.0786	0.0493	***0.4088***
Co-occurrence frequencies over Normalized Corpus	A	0.0738	0.0340	0.0346	0.0394	0.0361	0.0306	1.0000
	AR	**0.1185**	0.0546	0.0556	0.0633	0.0579	0.0492	0.6227
	C	0.1131	0.0521	0.0531	0.0604	0.0553	0.0469	1.0000
	CR	0.1188	0.0547	0.0557	0.0634	0.0581	0.0493	*0.952*
	E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
	ER	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	*0.0038*
Co-occurrence frequencies over Hearst Patterns	A	0.0119	0.0043	0.0066	0.0052	0.0040	0.0039	1.0000
	AR	**0.0211**	0.0077	0.0117	0.0092	0.0071	0.0068	0.5713
	C	0.0183	0.0067	0.0102	0.0080	0.0062	0.0059	1.0000
	CR	0.0211	0.0077	0.0117	0.0092	0.0071	0.0068	*0.8662*
	E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
	ER	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	*0.0019*
Word Similarity on Word Embeddings	A	0.0181	0.0073	0.0066	0.0082	0.0076	0.0068	1.0000
	AR	**0.0273**	0.0110	0.0100	0.0124	0.0115	0.0103	0.6627
	C	0.0259	0.0105	0.0102	0.0119	0.0109	0.0097	1.0000
	CR	0.0306	0.0124	0.0120	0.0140	0.0128	0.0114	0.8478
	E	0.0034	0.0012	0.0000	0.0012	0.0015	0.0015	1.0000
	ER	0.0109	0.0038	0.0000	0.0040	0.0048	0.0048	0.3148
Type	A (All)			concepts+entities				
	C			concepts only				
	E			entities only				
	R			compromising recall				

Table 4.7: Evaluation Scores for Training Data - Individual Sub-System Scores

values, the first hypernym fetched is the **correct hypernym**.

- The recall values for concepts is almost the same for all the the sub-systems. But the other scores like MAP, MRR were comparatively good for *IS-A Hearst Pattern* and *Co-occurrence frequencies over Normalized Corpus* only. The recall values signify that all the sub-systems fetch some candidate hypernyms for most

System Name	Type A, C, E	MRR	MAP	P@1	P@3	P@5	P@15	Recall
System Babbage	A	0.1259	0.0508	0.0846	0.0605	0.0503	0.0421	1.0000
	AR	**0.1558**	0.0628	**0.1047**	0.0749	0.0622	0.0521	0.7347
	C	0.1382	0.0602	0.0847	0.0687	0.0597	0.0534	1.0000
	CR	**0.1401**	0.0610	0.0859	0.0697	0.0605	0.0542	**0.9080**
	E	0.1026	0.0330	0.0844	0.0451	0.0325	0.0209	1.0000
	ER	**0.2174**	0.0699	**0.1788**	0.0955	0.0689	0.0442	**0.4088**
Type	A (All)			concepts+entities				
	C			concepts only				
	E			entities only				
	R			compromising recall				

Table 4.8: Evaluation Scores for Training Data - Final Babbage System Scores

of the *concept* input terms.

- System Babbage has a very high recall value of **0.9080** (close to 1.00) for concepts and only **0.4088** (less than 50%) for entities. We did not apply Named Entity Recognition while normalizing the UMBC corpus. Named Entity Recognizers help in retaining the Entities which our normalization module might discard. This could be the reason for such recall score difference.

- The **MRR** score of system Babbage for entities (**0.2174**) and concepts (**0.1401**) indicate that on an average, the first correct hypernym for the given input terms is found within the first 5 predicted hypernyms for entities and first 8 predicted hypernyms for concepts.

The Venn diagrams in Figures 4.13, Figure 4.14 and Figure 4.15 represent the recall coverage of the sub-systems of this task when applied on the test data. The test data consists of **1500** input terms with ***1057*** *Concepts* and ***433*** *Entities*.

Table 4.10 shows the results obtained after applying our **Babbage** system to the training data. Table 4.9 shows the breakdown of scores for various sub-systems and scores with respect to concepts and entities from this test data. The following are

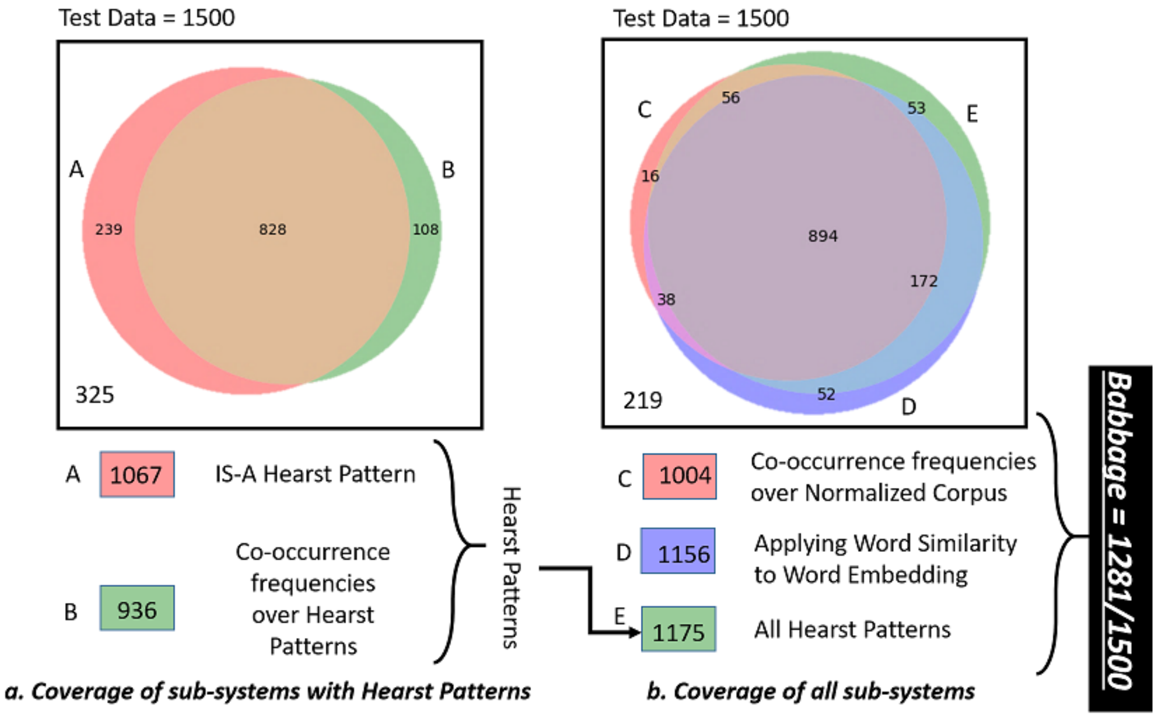

Test Data = 1500
A
B
239
828
108
325
Test Data = 1500
C
56
53
E
16
894
172
38
219
52
D
A
1067
IS-A Hearst Pattern
B
936
Co-occurrence frequencies over Hearst Patterns
Hearst Patterns
C
1004
Co-occurrence frequencies over Normalized Corpus
D
1156
Applying Word Similarity to Word Embedding
E
1175
All Hearst Patterns
a. Coverage of sub-systems with Hearst Patterns
b. Coverage of all sub-systems
Babbage = 1281/1500

Figure 4.13: Venn Diagrams representing the recall values of the sub-systems for SemEval 2018 Task 9 Hypernym Discovery. The image represents the recall values for the entire test sample including concepts and entities.

the key observations made form these tables and a brief comparison of these scores with the training data result scores:

- The order of merging results obtained from the training data is applied to the test data. The intuition for the merge order obtained from the training data proved to be true even for the test data. This intuition is based on the *AR - MRR* scores of the sub-systems.

- The *IS-A Hearst Pattern* module performs the best for entities both in terms of recall and the other evaluation scores even for the test data. Though the recall value for the *Word Embedding* module is good, the other evaluation scores are not good.

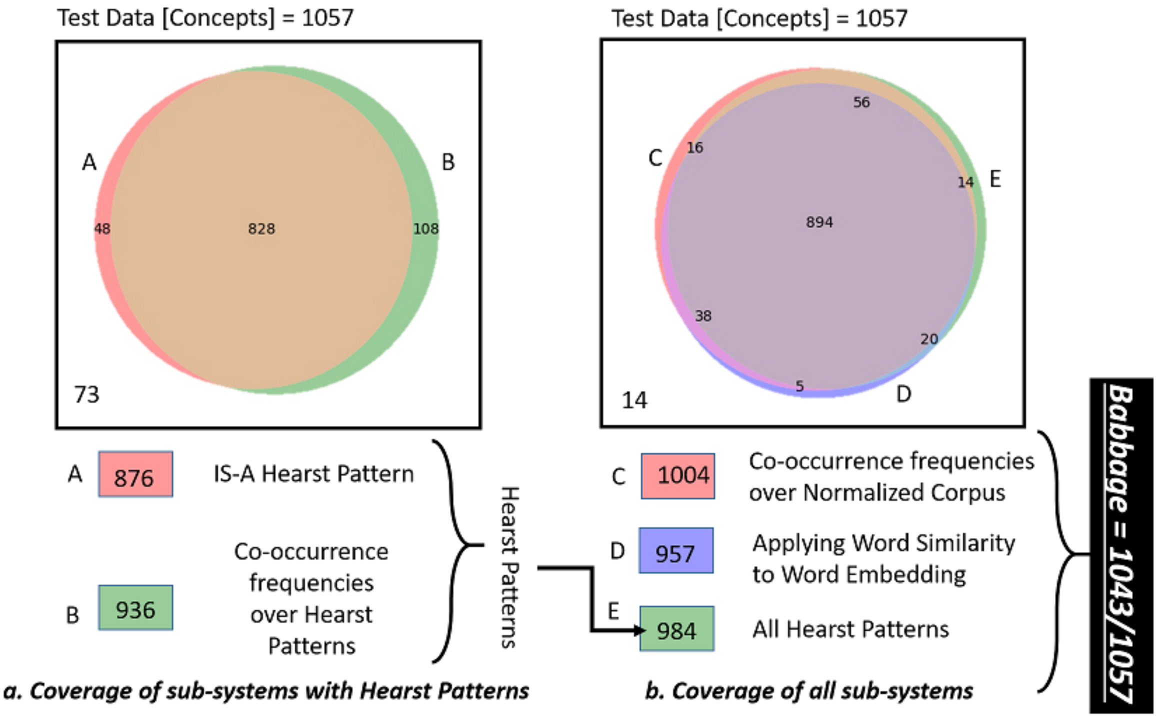

Test Data [Concepts] = 1057
A
B
48
828
108
73
A
876
IS-A Hearst Pattern
B
936
Co-occurrence frequencies over Hearst Patterns
Hearst Patterns
a. Coverage of sub-systems with Hearst Patterns
Test Data [Concepts] = 1057
C
16
56
14
E
894
38
20
5
D
14
C
1004
Co-occurrence frequencies over Normalized Corpus
D
957
Applying Word Similarity to Word Embedding
E
984
All Hearst Patterns
b. Coverage of all sub-systems
Babbage = 1043/1057

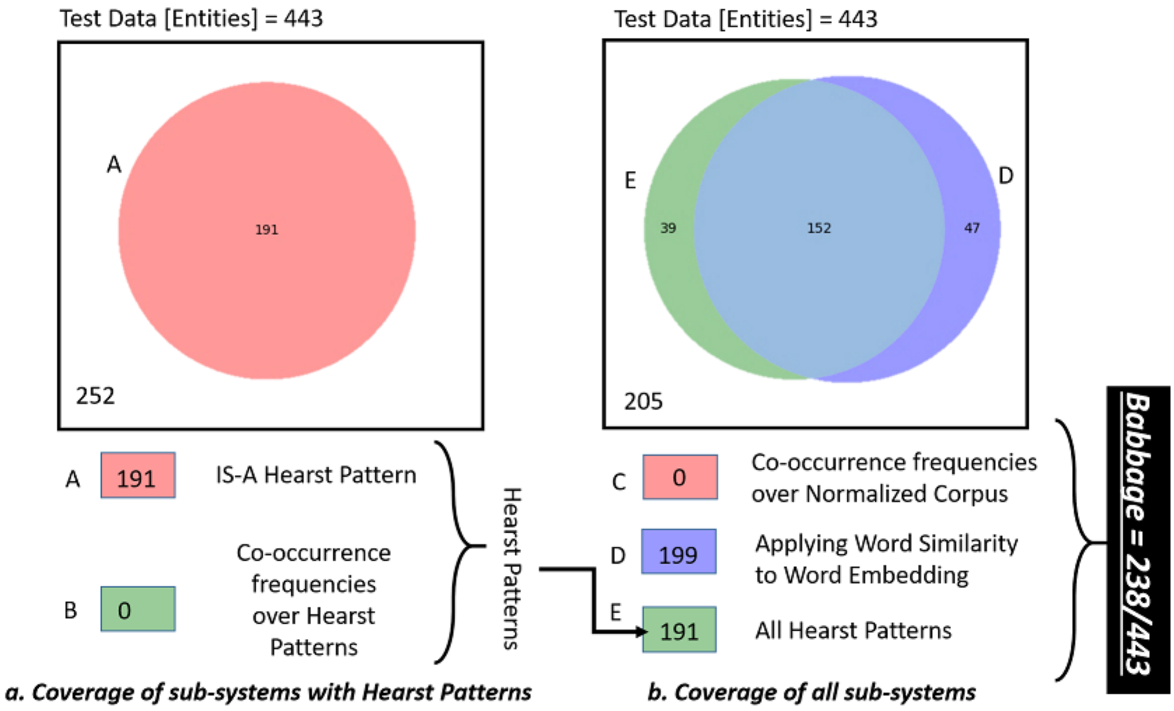
Figure 4.14: Venn Diagrams representing the recall values of the sub-systems for SemEval 2018 Task 9 Hypernym Discovery. The image represents the recall values for only Concept input terms from the test sample.

- Even for the test data, the *P1* score of Babbage system for entities is the best.

- The recall scores of all sub-systems are almost similar to one another when applied to concepts. This indicates that even for the test data, out sub-systems fetched hypernyms for most of the c*concept* input terms. And other scores like MAP, MRR were also good for *IS-A Hearst Pattern* and *Co-occurrence frequencies over Normalized Corpus* sub-systems only.

- The overall recall value of System Babbage is higher for test data than training data, but other evaluation scores like *MRR*, *MAP* and *P1* are lower than the training data.

- Since the **MRR** score for test data is lower than the training data, the prob-


Test Data [Entities] = 443
A
191
252
Test Data [Entities] = 443
E
39
152
47
D
205
A
191
IS-A Hearst Pattern
B
0
Co-occurrence frequencies over Hearst Patterns
Hearst Patterns
C
0
Co-occurrence frequencies over Normalized Corpus
D
199
Applying Word Similarity to Word Embedding
E
191
All Hearst Patterns
Babbage = 238/443

a. Coverage of sub-systems with Hearst Patterns

b. Coverage of all sub-systems

Figure 4.15: Venn Diagrams representing the recall values of the sub-systems for SemEval 2018 Task 9 Hypernym Discovery. The image represents the recall values for only Entity input terms from the test sample.

ability of fetching the correct hypernym within the first **5** and **8** predicted hypernyms for entities and concepts in training data change to **7** and **11** in test data.

System Name	Type A, C, E	MRR	MAP	P@1	P@3	P@5	P@15	Recall
IS-A Hearst Pattern	A	0.0883	0.0357	0.0653	0.0403	0.0340	0.0305	1.0000
	AR	**0.1241**	0.0503	0.0918	0.0567	0.0478	0.0429	0.7113
	C	0.0880	0.0359	0.0633	0.0405	0.0340	0.0310	1.0000
	CR	**0.1061**	0.0433	0.0764	0.0488	0.0411	0.0374	0.8288
	E	0.0890	0.0353	0.0699	0.0398	0.0338	0.0294	1.0000
	ER	**0.2064**	0.0820	0.1623	0.0924	0.0786	0.0683	0.4312
Co-occurrence frequencies over Normalized Corpus	A	0.0820	0.0388	0.0506	0.0428	0.0372	0.0359	1.0000
	AR	**0.1225**	0.0580	0.0756	0.0640	0.0556	0.0537	0.6693
	C	**0.1163**	0.0551	0.0719	0.0608	0.0529	0.0510	1.0000
	CR	**0.1225**	0.0580	0.0756	0.0640	0.0556	0.0537	0.9499
	E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
	ER	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Co-occurrence frequencies over Hearst Patterns	A	0.0137	0.0052	0.0093	0.0057	0.0050	0.0045	1.0000
	AR	**0.0220**	0.0084	0.0149	0.0092	0.0081	0.0072	0.6240
	C	0.0195	0.0074	0.0132	0.0081	0.0072	0.0063	1.0000
	CR	0.0220	0.0084	0.0149	0.0092	0.0081	0.0072	0.8855
	E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	ER	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Word Similarity on Word Embeddings	A	0.0192	0.0080	0.0100	0.0090	0.0078	0.0075	1.0000
	AR	**0.0250**	0.0104	0.0129	0.0116	0.0102	0.0097	0.7707
	C	0.0236	0.0100	0.0122	0.0115	0.0099	0.0094	1.0000
	CR	0.0260	0.0111	0.0135	0.0127	0.0109	0.0104	0.9054
	E	0.0089	0.0033	0.0045	0.0030	0.0030	0.0030	1.0000
	ER	0.0199	0.0073	0.0100	0.0067	0.0067	0.0068	0.4492
Type	A (All)			concepts+entities				
	C			concepts only				
	E			entities only				
	R			compromising recall				

Table 4.9: Evaluation Scores for Test Data - Individual Sub-System Scores

System Name	Type A, C, E	MRR	MAP	P@1	P@3	P@5	P@15	Recall
System Babbage	A	0.1276	0.0543	0.0913	0.0600	0.0519	0.0478	1.0000
	AR	0.1494	0.0636	0.1069	0.0702	0.0608	0.0560	0.7833
	C	0.1405	0.0617	0.0964	0.0671	0.0589	0.0556	1.0000
	CR	**0.1424**	0.0625	0.0977	0.0680	0.0597	0.0563	**0.9309**
	E	0.0969	0.0366	0.0790	0.0428	0.0352	0.0293	1.0000
	ER	**0.1803**	0.0681	0.1470	0.0798	0.0656	0.0546	**0.4312**
Type	A (All)			concepts+entities				
	C			concepts only				
	E			entities only				
	R			compromising recall				

Table 4.10: Evaluation Scores for Test Data - Final Babbage System Scores

5 Conclusions

This chapter explains a few important discoveries made while working on the two SemEval tasks *SemEval 2016 Task 14 - Semantic Taxonomy Enrichment* and *SemEval 2018 Task 09 - Hypernym Discovery*. While analyzing the results, we identified a few possible modifications and extensions to the existing sub-systems 3.2. We propose some improvements under the *Future Work* section of this chapter.

5.1 Contributions

Both the **Hypernym Discovery** task and **Semantic Taxonomy Enrichment** task (Chapter 2 Section 2.4.4) focus on identifying potential hypernym(s) for a given input lemma. *Hearst Patterns* and *Similarity over Word-Embeddings* (Chapter 3) are the key strategies applied to address this problem. The two major stages of our proposed systems are *pre-processing the UMBC WebBase Corpus* 3.1 and *identifying the hypernym(s) from the pre-processed corpora or WordNet* 3.1.

The following are the key contributions from this research towards ***SemEval 2018 Task 09: Hypernym Discovery for English***:

1. *Creating ready to use pre-processed corpora to speed up execution time* : Some participants [Onofrei, Hulub, Trandabat, and Gifu 2018] and [Aldine, Harzallah, Berio, Béchet, and Faour 2018] mentioned that they could not run their system against all the inputs from the test data. The size of the UMBC WebBase Data Corpus (28.3GB) was the reason for this failure. On the other hand, we were

able to run all our sub-systems on the entire test data. This could be achieved by reducing the size of UMBC WebBase Corpus to approximately **17** GB in the pre-processing phase.

2. *Validating the pre-processed UMBC corpora against a general purpose hypernym discovery task:* We used this corpus with another task - *SemEval 2016 Task 09 - Hypernym Discovery* whose underlying problem was also discovering a hypernym for a given term. We built two sub-systems *Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus* 3.2.2 and *Similarity with Definition over the Word-Embedding - UMBC Word-Embedding* 3.2.6 using this pre-processed corpora. The evaluation scores of these two sub-systems are higher than the Random 4.2.1 and Default 4.2.3 baselines. A sub-system which performs better than these baselines could be considered as a valid model for hypernym discovery. Table 5.1 shows few examples where these sub-systems could fetch more accurate hypernyms than all the baseline systems. The *Predicted Hypernym* is the result from our sub-system and the *Actual Hypernym* is the gold hypernym provided with the task. With this we could conclude that the pre-processed UMBC WebBase corpora could be used for any general purpose hypernym discovery task (in English).

3. From Table 4.10, the MRR evaluation score for the inputs *(Concepts + Entities)* of the sub-systems *Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus* (Co-occur over IS-A corpus) 3.2.2 and *Co-occurrence Frequencies over the UMBC Normalized Corpus* (Co-occcur Over Normalized Corpus) 3.2.1 are almost similar. And these sub-systems perform better than the other two sub-systems - *Co-occurrence Frequencies over the UMBC Other Hearst Corpus* (Chapter 3 Section 3.2.3) and *Hypernymy Similarity Distance over the UMBC*

<table>
<tr><td colspan="4">Similarity with Definition over the Word-Embedding - UMBC Word-Embedding</td></tr>
<tr><th>Input OOV lemma</th><th>Predicted Hypernym</th><th>Actual Hypernym</th><th>Wu & Palmer Score</th></tr>
<tr><td><b>graphics card</b></td><td>computer#n#1</td><td>circuit_board#n#1</td><td>0.6667</td></tr>
<tr><td><b>kill</b></td><td>hitting#n#1</td><td>stroke#n#1</td><td>0.5882</td></tr>
<tr><td><b>aquitard</b></td><td>lithosphere#n#1</td><td>stratum#n#1</td><td>0.8571</td></tr>
<tr><td><b>home care</b></td><td>care#n#1</td><td>healthcare#n#1</td><td>0.9412</td></tr>
<tr><td><b>laser surgery</b></td><td>microsurgery#n#1</td><td>surgery#n#1</td><td><b>0.9474</b></td></tr>
<tr><td><b>aeolian</b></td><td>semitone#n#1</td><td>musical_mode#n#1</td><td>0.7368</td></tr>
<tr><td><b>celestine</b></td><td>ecclesiastic#n#1</td><td>monk#n#1</td><td>0.5714</td></tr>
<tr><td><b>finger lake</b></td><td>rivulet#n#1</td><td>lake#n#1</td><td>0.7272</td></tr>
<tr><td><b>cost avoidance</b></td><td>cost#n#1</td><td>expense#n#1</td><td><b>0.9333</b></td></tr>
<tr><td colspan="4">Co-occurrence Frequencies over the UMBC IS-A Hearst Corpus</td></tr>
<tr><th>Input OOV lemma</th><th>Predicted Hypernym</th><th>Actual Hypernym</th><th>Wu & Palmer Score</th></tr>
<tr><td><b>fining</b></td><td>removal#n#1</td><td>clearing#n#1</td><td>0.7</td></tr>
<tr><td><b>immunoglobin</b></td><td>antibody#n#1</td><td>antibody#n#1</td><td><b>1.0</b></td></tr>
<tr><td><b>tenolysis</b></td><td>procedure#n#1</td><td>surgery#n#1</td><td>0.875</td></tr>
</table>

Table 5.1: Wu & Palmer similarity score with respect to individual results of test data. Refer Appendix A for the definitions of out-of-vocabulary (OOV) lemmas.

Word-Embedding (Chapter 2 Section 3.2.4). The scores for the *Co-occur Over Normalized Corpus* show that applying distributional semantics to text corpora could identify hypernymy relationship to some extent when given a text corpus. Similarly, some types of Hearst Patterns perform significantly better than the rest of these patterns.

4. Though the overall scores of the **Co-occur over *IS-A corpus* and *Normalized Corpus*** sub-systems are close to one another (0.1330 and 0.1185), the scores of Concepts only and Entities only results are significantly different in terms of *Recall* value. The **Co-occur over *IS-A corpus*** sub-system could fetch results for ≈ 43% Entities but the **Co-occur over *Normalized Corpus*** failed to get results when the input term is an Entity (From Table 5.2, the **ER** row for *Co-occur over Normalized Corpus* has all 0ś). All the results

		Training Data			Test Data		
System Name	Type A, C, E	MRR	MAP	Recall	MRR	MAP	Recall
IS-A Hearst Pattern	A	0.0863	0.0329	1.0000	0.0883	0.0357	1.0000
	AR	**0.1330**	0.0507	0.6273	0.1241	0.0503	0.7113
	C	0.0783	0.0330	1.0000	0.0880	0.0359	1.0000
	CR	0.1007	0.0424	0.7773	0.1061	0.0433	0.8288
	E	0.1014	0.0328	1.0000	0.0890	0.0353	1.0000
	ER	0.2482	0.0804	**0.4088**	0.2064	0.0820	**0.4312**
Co-occurrence frequencies over Normalized Corpus	A	0.0738	0.0340	1.0000	0.0820	0.0388	1.0000
	AR	**0.1185**	0.0546	0.6227	0.1225	0.0580	0.6693
	C	0.1131	0.0521	1.0000	0.1163	0.0551	1.0000
	CR	0.1188	0.0547	0.952	0.1225	0.0580	0.9499
	E	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
	ER	0.0000	0.0000	0.0038	0.0000	0.0000	0.0000
Type	A (All) C E R			concepts+entities concepts only entities only with recall			

Table 5.2: Evaluation scores for *IS-A Hearst Pattern* and *Co-occurrence frequencies over Normalized Corpus* sub-systems from Tables 4.7 and 4.9

obtained from the *Normalized corpus* are hypernyms for Concepts only (With recall value 0.9499 close to 1). Hence this module could be ranked higher than the **Co-occur over *IS-A corpus*** in the merge order for Concepts. So if we change the merge order based on the Concepts only and Entities only *Recall* score, the final scores for the System Babbage could increase for Concepts (Table 5.2 shows these scores).

5. *Creating independent corpora for different formats of Hearst Patterns is helpful to fetch more appropriate hypernyms.* There is a significant difference in the evaluation scores (shown in Table 5.3) between the sub-systems - *Co-occurrence Frequencies over the UMBC **IS-A Hearst Corpus** 3.2.2 or **Other Hearst Corpus** 3.2.3.* This signifies that *One-to-One Hearst Patterns* could be as-

System Name	Type A, C, E	MRR	MAP	P@1	P@3	P@5	P@15
IS-A Hearst Pattern	AR	<u>**0.1241**</u>	0.0503	0.0918	0.0567	0.0478	0.0429
Other Hearst Patterns	**AR**	<u>**0.0220**</u>	0.0084	0.0149	0.0092	0.0081	0.0072
A-(All Concepts and Entities) R-Compromising Recall							

Table 5.3: Evaluation Scores for Test - only Hearst Pattern based Sub-System Scores from Table 4.9

signed more weights (higher MRR score 0.1241 than Other Hearst Patterns 0.0220) than *Many-to-One Hearst Patterns* and they contain more information about hypernymy than the other sub-system.

The following are the key contributions from this research towards ***SemEval 2016 Task 14: Semantic Taxonomy Enrichment***:

1. *Need for a new baseline system - Default Baseline 4.2.3:* Initially when we applied Hearst Patterns over the Out-Of-Vocabulary (OOV) lemmas definitions 3.2.5, we realized that there existed a few cases where we have no results, especially for verbs. In order to improve the recall value, we had to come up with a default hypernym when the system cannot locate an appropriate hypernym from WordNet. Since the root node for the noun hypernym structure in WordNet is *entity#n#1*, we chose this as a default hypernym for all the noun OOV lemmas where a sub-system fails. Unlike nouns, for verbs there are several hundred verb hypernym tree structures in WordNet. The creators of WordNet had a philosophy that there is no one particular verb which could represents all verbs at an abstract level unlike for nouns where an *entity* could represent all nouns. So we choose a random verb synset *be#v#1* as the default hypernym. If we chose to add these defaults when a sub-system fails to fetch a hypernym,

system or sub-system	Type	**Wu & Palmer**	**Lemma Match**	**Recall**
Random Baseline	all	0.2269	0.0000	1.0000
Default Baseline	all	0.2519	0.0000	1.0000
Type : All, nouns, verbs				

Table 5.4: The evaluation scores for the SemEval 2016 task 14 - Taxonomy Enrichment systems against test data. All the scores in this table are rounded to the 10000th decimal place

we need a new baseline system with just these default values to evaluate this sub-system. We create this baseline by assigning "entity#n#1" for all noun OOV lemmas and "be#v#1" for all verb OOV lemmas as hypernyms. The similarity score for this baseline with the gold key is higher that the *Random* baseline (Refer Table 5.4). Hence by assigning defaults to the failed cases of any sub-system, we would still get a better similarity score for the entire sub-system than the Random baseline system.

2. We applied the following Hearst Patterns 3.13 to the definitions from the test data's Out-Of-Vocabulary (OOV) lemmas:

   - ⟨*Hypernym*⟩ ***is (a | an | the)*** ⟨*Hyponym*⟩
   - ⟨*Hypernym*⟩ ***such as*** ⟨*Hyponym*1⟩ ⟨*Hyponym*2⟩, ..., (and | or) ⟨*HyponymN*⟩
   - ⟨*Hyponym*1⟩ ⟨*Hyponym*2⟩, ..., (and | or) ⟨*HyponymN*⟩ ***(and | or) other*** ⟨*Hypernym*⟩
   - ⟨*Hypernym*⟩ ***(including | especially)*** ⟨*Hyponym*1⟩ ⟨*Hyponym*2⟩, ..., (and | or) ⟨*HyponymN*⟩

   The evaluation scores obtained for the hypernyms from these patterns (excluding the defaults *entity#n#1* and *be#v#1*) are higher than all the baselines. Table 5.5 is a snippet from Table 4.6. With Hearst Patterns, we obtain hypernyms only for nouns. So if we observe the recall score for noun OOV lemmas,

System Name (All, nouns, verbs)	**Type**	**Wu & Palmer**	**Lemma**	**Recall**
Definition Hearst Patterns	All	0.5682	0.4336	0.4336
	nouns	0.5682	0.4336	0.7717
	verbs	0.0000	0.0000	0.0000
Random Baseline	All	0.2269	0.0000	1.0000
Default Baseline	All	0.2519	0.0000	1.0000
First-word First-sense Baseline	All	0.5140	0.4150	1.0000

Table 5.5: The evaluation scores for the ***SemEval 2016 task 14 - Taxonomy Enrichment sub-system*** - **Definition Hearst Patterns** 3.2.5 All the scores in this table are rounded to the 10000th decimal place

this sub-system could get results for 399 nouns on 517 nouns, which is approximately *70%*. The overall score of this sub-system is above the baseline systems when applied to all OOV lemmas including verbs (with default *be#v#1*). This proves that Hearst Patterns have the potential to give information about hypernyms when applied to the definitions of nouns.

3. Hypernyms obtained from from **Definition Hearst Patterns** 3.2.5 for few Out-Of-Vocabulary lemma are almost similar to ones from the sub-system **UMBC Word-Embedding** 3.2.4. Table 5.6 shows a few examples which represent this similarity. The Wu & Palmer similarity between the hypernyms - *tint#n#1* and *color#n#1* - obtained from these two systems for the OOV lemma "streak" is very high (*0.9231*) and the synset *color#n#1* is the immediate hypernym for the synset *tint#n#1* in WordNet.This similarity score indicates that reformatting the definition as "OOV" lemma *is (a | an | the) definition* to apply Hearst Pattern is actually helping the system to extract a more accurate hypernym. Figure 5.1 shows the process of extracting hypernyms after re-formatting the definition.

4. Performance of the sub-systems proposed for *SemEval 2016 Task 14: Semantic*

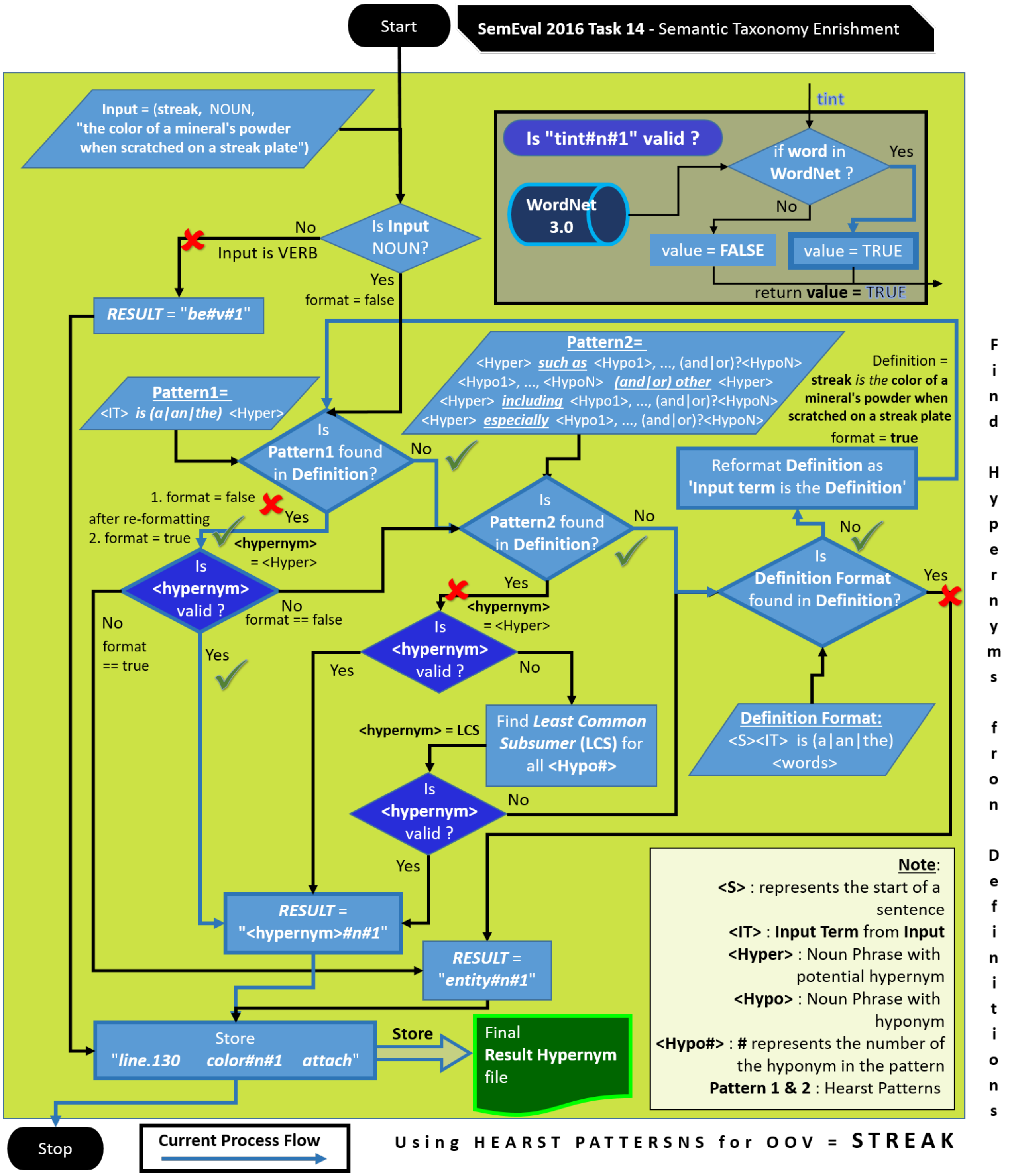

Start
SemEval 2016 Task 14 - Semantic Taxonomy Enrishment
Input = (streak, NOUN, "the color of a mineral's powder when scratched on a streak plate")
tint
Is "tint#n#1" valid ?
if word in WordNet ?
Yes
No
WordNet 3.0
value = FALSE
value = TRUE
return value = TRUE
Is Input NOUN?
No
Input is VERB
Yes
format = false
RESULT = "be#v#1"
Pattern2=
<Hyper> such as <Hypo1>, ..., (and|or)?<HypoN>
<Hypo1>, ..., <HypoN> (and|or) other <Hyper>
<Hyper> including <Hypo1>, ..., (and|or)?<HypoN>
<Hyper> especially <Hypo1>, ..., (and|or)?<HypoN>
Definition = streak is the color of a mineral's powder when scratched on a streak plate
format = true
Pattern1=
<IT> is (a|an|the) <Hyper>
Is Pattern1 found in Definition?
No
Reformat Definition as 'Input term is the Definition'
1. format = false
Yes
after re-formatting
2. format = true
<hypernym> = <Hyper>
Is Pattern2 found in Definition?
No
No
Is <hypernym> valid ?
No
format == false
Yes
<hypernym> = <Hyper>
Is Definition Format found in Definition?
Yes
No
format == true
Yes
Is <hypernym> valid ?
Yes
No
Definition Format:
<S><IT> is (a|an|the) <words>
<hypernym> = LCS
Find Least Common Subsumer (LCS) for all <Hypo#>
Is <hypernym> valid ?
No
Yes
Note:
<S> : represents the start of a sentence
<IT> : Input Term from Input
<Hyper> : Noun Phrase with potential hypernym
<Hypo> : Noun Phrase with hyponym
<Hypo#> : # represents the number of the hyponym in the pattern
Pattern 1 & 2 : Hearst Patterns
RESULT = "<hypernym>#n#1"
RESULT = "entity#n#1"
Store "line.130 color#n#1 attach"
Store
Final Result Hypernym file
Stop
Current Process Flow
Using HEARST PATTERSNS for OOV = STREAK
Find Hypernyms from Definitions

Figure 5.1: Applying Hearst Pattern to OOV lemma *streak* definition to extract a Hypernym

OOV Lemma	Hypernyms from Key	UMBC Word-Embedding		Definition Hearst Patterns	
		Score	Result	Score	Result
aquitard	stratum#n#1	0.8571	lithosphere#n#1	1.0000	stratum#n#1
streak	color#n#1	0.9231	tint#n#1	1.0000	color#n#1
inducement	statement#n#1	0.8000	information#n#1	1.0000	statement#n#1
bottle	nerve#n#2	0.8235	fortitude#n#1	0.875	courage#n#1
cost avoidance	expense#n#1	0.9333	cost#n#1	1.0	expense#n#1
deadlift	weightlift#n#1	0.8333	sit-up#n#1	0.9091	exercise#n#1
score = Wu & Palmer Similarity Score					

Table 5.6: The evaluation scores for the ***SemEval 2016 task 14 - Taxonomy Enrichment sub-systems*** - **UMBC Word-Embedding** 3.2.6 and **Definition Hearst Patterns** 3.2.5. All the scores in this table are rounded to the 10000th decimal place. Refer Appendix A for definitions of out-of-vocabulary (OOV) lemmas

Taxonomy Enrichment which fetch results also for **verbs** OOV lemmas: Modules which relied on Hearst Pattern recognition failed to fetch hypernyms for OOV verb lemmas (Table 4.6), so all the scores reported for verbs from these modules under ***With Default Hypernyms*** column are the scores for the default hypernym *be#v#1.* The only sub-system which identifies hypernyms for at least a few verb OOV lemmas is *Similarity with Definition over the Word-Embeddings* 3.2.6 So the inclusion of the embedding based hypernym discovery helped us address this problem for verb OOV lemmas.

5. *Ranking of our system with respect to the other systems proposed for this task :* We use the *F1 measure* shown in Formula 5.1 to rank our system against the system proposed for this task. The *Wu & Palmer score* and *Recall score* from the Table 5.7 are used to calculate the F1 Score. The F1 score for our entire system (all) is *0.641.* The F1 score for only noun inputs is *0.662* and verb input is *0.497.* Our system was ranked **4** when we consider all POS tags and stood 3^{rd} when considering only nouns and ignoring the verbs.

system or sub-system	Type	**Wu & Palmer**	**Lemma Match**	**Score**
	all	0.4722	0.2917	1.000
System Babbage	noun	0.4948	0.3346	1.000
	verb	.3309	0.0240	1.000

Table 5.7: Our final system proposed for SemEval 2016 task 14 - Taxonomy Enrichment task against test data. All the scores in this table are rounded to the 10000^{th} decimal place

$$F1 = 2(Wu\&Palmer * Recall)/(Wu\&Palmer + Recall) \tag{5.1}$$

5.2 Future Scope

Based on our results and contributions we propose the following improvements for the existing systems:

1. *Adding more Hearst Patterns to the Noun Hearst Corpora :* In this research we used Hearst Patterns with **4** keyword patterns which capture hypernymy 2. We created two corpora: UMBC IS-A Hearst Corpus 3.1.2 and UMBC Other Hearst Corpus 3.1.3, using these patterns. There are more patterns which could be used to extract hypernym-hyponym pairs from text corpus. An example of one such pattern is "*examples of ⟨Hypernym⟩ (is | are) ⟨Hyponym1⟩[, ⟨Hyponym2⟩, ..., ⟨HyponymN⟩]*". Please refer to the paper "A Large Database of Hypernymy Relations Extracted from the Web" [Seitner, Bizer, Eckert, Faralli, Meusel, Paulheim, and Ponzetto 2016] for more such patterns.

2. *Pattern based hypernymy detection for Verbs :* Hearst Patterns are applied over noun phrases which extract hypernyms for nouns from text based corpora. However the Hearst Pattern based sub-systems have a **0.0000** recall value for the sub-set of verbs. We could create new corpora from the UMBC WebBase Corpus which represent hypernym-hyponym relationship for verbs. In the pa-

per “VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations.” [Chklovski and Pantel 2004], the authors mentioned working on the following **5** semantic relationships between verbs:

(a) **similarity** : Verbs with similar meaning. An example is *kick_back#v#2*, *recoil#v#4*

(b) **strength** : One verb has more intense meaning than the other. An example is *slap#v#1*, *smack#v#1*

(c) **antonymy** : One verb is the opposite of another. An example is *front#v#1*, *back#v#6*

(d) **enablement** : One verb could be accomplished by another. An example is *participate#v#1*, *win#v#1*

(e) **happens-before** : One verb occurs before another in a given time-line. An example is *live#v#1*, *die#v#1*

We could use these relations to extract hypernymy relationship. For example, if the enablement relationship found for a new OOV lemma *mount* is *access#v#2*, then we could propose the hypernym of *access#v#2* as a hypernym for *mount*.

3. *Using similarity scores between the hypernyms of sub-systems to refine results :* In this research, we merged the final results using *Select One* merge module 3.4.1 where we rank results from one system higher than the other based on their training data evaluation scores. But we did not use the results of one sub-system to evaluate the accuracy of the results of another sub-system. If we could include this sort of evaluation as a filter before accepting a hypernym as a potential result, the overall Wu & Palmer similarity score of the entire system could increase. This might reduce the over-all recall score but could fetch

hypernyms with more accuracy. Table 5.8 shows a few hypernyms identified by the Definition Hearst pattern sub-system 3.2.5 whose Wu & Palmer similarity score (against gold standard hypernym) is greater than *0.9*. We calculate the similarity scores for these results against the results of another sub-system (considering these results as the gold standard hypernyms). These new similarity scores are also high (greater than *0.8*). This indicates that we could consider a result as a high accuracy hypernym when its similarity score with the result of the other sub-system is above some predefined threshold (say *0.8* for this example). We can call this *Sub-System Filtering*. This initial filtering worked to an extent and obtained results that support this strategy for both accept and reject cases. However we encountered some true negatives as well. With further investigation, these negative cases could also be converted into positive cases by adding more filters.

4. *Using hypernyms obtained from SemEval task to refine hypernym obtained from another SemEval task for an* ***Input Term****:* For a given New Out-Of-Vocabulary lemma, the current system extracts hypernyms by using System-Babbage of *SemEval 2016 Task 14 Semantic Taxonomy Enrichment* 3.4.1. In this process the first hypernym which exists in WordNet from the list of intermediate hypernyms is assigned as the result hypernym "*result1*". There could be a more precise hypernym from this list which is better than the current predicted result hypernym. So if we apply System-Babbage for *SemEval 2018 Task 09 Hypernym Discovery* 3.4.2 to this input term, we get a list of upto 15 possible hypernyms. We could use this list to construct a *Hypernym-tree* (as shown in figure 5.2). If "*result1*" exists in this Hypernym-tree, then we could choose a more precise synset from this tree which is also a hyponym to this "*result1*". For example,

OOV Lemma	**Hypernyms from Key**	**UMBC Word-Embedding**		**Definition Hearst Patterns [HP]**	
		Score with-HP	Result	Score with-key	Result
Positive cases where *Sub-System Filtering* could **accept** results of another sub-system					
aquitard	stratum#n#1	0.8571	lithosphere #n#1	1.0000	stratum#n#1
streak	color#n#1	0.9231	tint#n#1	1.0000	color#n#1
inducement	statement#n#1 #n#1	0.8000	information #n#1	1.0000	statement #n#1
bottle	nerve#n#2	0.9333	fortitude#n#1	0.875	courage#n#1
cost avoidance	expense#n#1	0.9333	cost#n#1	1.0	expense#n#1
deadlift	weightlift#n#1	0.9090	sit-up#n#1	0.9091	exercise#n#1
Positive cases where *Sub-System Filtering* could **reject** results of another sub-system					
adjuvant	immunogen #n#1	0.3077	cell#n#1	0.6	substance #n#1
tablet	portable_ computer#n#1	0.087 0.087	laptop#n#1	0.0909	type#n#1
palliative care	hospital_ room#n#1	0.1429	care#n#1	0.4	area#n#1
Negative cases where *Sub-System Filtering* could **reject** results of another sub-system					
lovibond	scale#n#1	0.0833	beef#n#1	1.0	scale#n#1
endograft	graft#n#1	0.1053	thrombectomy #n#1	1.0	graft#n#1
score = Wu & Palmer Similarity Score					

Table 5.8: *Similarity_score(⟨hypernym from UMBC Word-Embedding⟩, ⟨hypernym from Definition Hearst Patterns⟩)* : score to filter results of Definition Hearst Patterns sub-system. All the scores in this table are rounded to the 10000th decimal place. Refer Appendix A for definitions of out-of-vocabulary (OOV) lemmas

from figure 5.2, the "*result1*" for input term "*palliative_care*" after applying System-Babbage for *SemEval 2016 Task 14 Semantic Taxonomy Enrichment* is "*care#n#1*". The list of hypernyms for this input from the final system of *SemEval 2018 Task 09 hypernym Discovery* is { care, personal_care, nursing_care, primary_care }. Since our system reports all these words as candidate hypernyms, a hypernym which is a leaf node in the Hypernym-tree (constructed over this list) could be the most precise hypernym for a given term. In the cur-

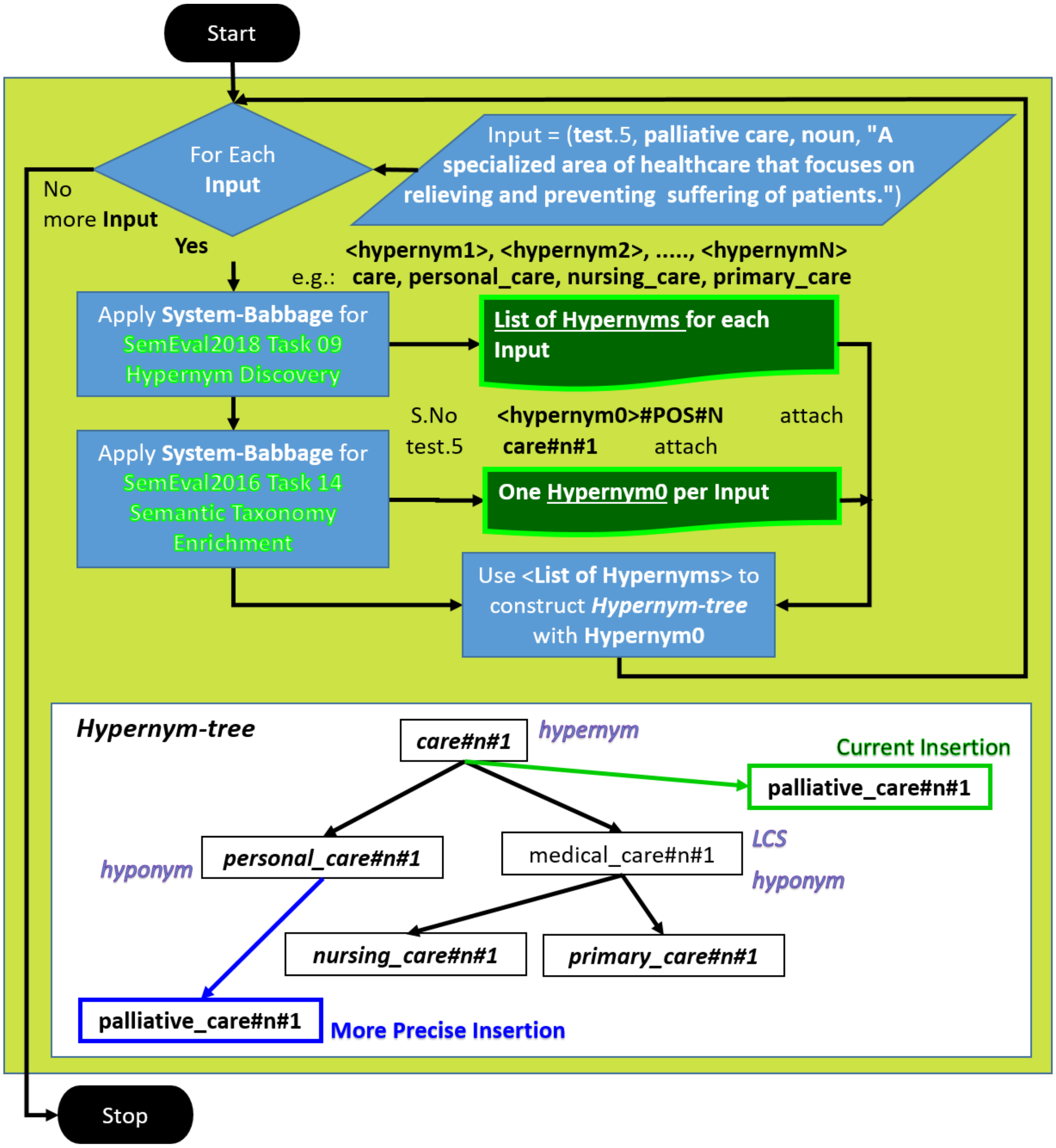
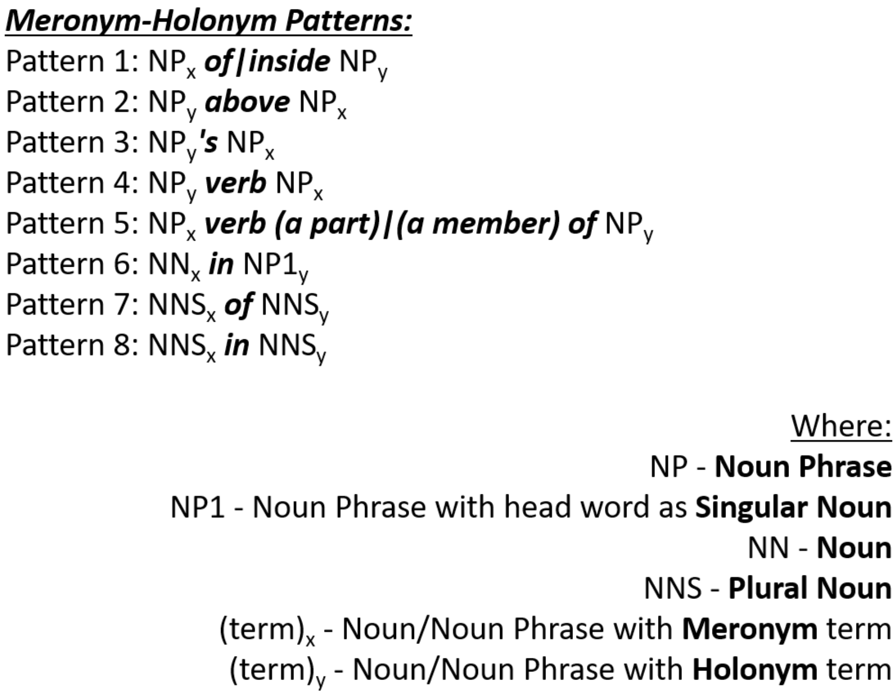

Start
For Each Input
Input = (test.5, palliative care, noun, "A specialized area of healthcare that focuses on relieving and preventing suffering of patients.")
No more Input
Yes
<hypernym1>, <hypernym2>, ….., <hypernymN>
e.g.: care, personal_care, nursing_care, primary_care
Apply System-Babbage for SemEval2018 Task 09 Hypernym Discovery
List of Hypernyms for each Input
S.No <hypernym0>#POS#N attach
test.5 care#n#1 attach
Apply System-Babbage for SemEval2016 Task 14 Semantic Taxonomy Enrichment
One Hypernym0 per Input
Use <List of Hypernyms> to construct Hypernym-tree with Hypernym0
Hypernym-tree
care#n#1
hypernym
Current Insertion
palliative_care#n#1
hyponym
personal_care#n#1
medical_care#n#1
LCS
hyponym
nursing_care#n#1
primary_care#n#1
palliative_care#n#1
More Precise Insertion
Stop

Figure 5.2: Using hypernyms from *System-Babbage for SemEval 2016 Task 14 Semantic Taxonomy Enrichment* to refine the result predicted by *System-Babbage for SemEval 2018 Task 09 Hypernym Discovery*

rent example, is also a hypernym for "*palliative_care*". The synset *care#n#1* is a super-ordinate for *personal_care#n#1*. So the input term "*palliative_care*" is closer to *personal_care#n#1* than *caren1* and the final result should be re-assigned to *personal_care#n#1*.



<u>Meronym-Holonym Patterns:</u>

Pattern 1: NP_x ***of|inside*** NP_y
Pattern 2: NP_y ***above*** NP_x
Pattern 3: NP_y***'s*** NP_x
Pattern 4: NP_y ***verb*** NP_x
Pattern 5: NP_x ***verb (a part)|(a member) of*** NP_y
Pattern 6: NN_x ***in*** $NP1_y$
Pattern 7: NNS_x ***of*** NNS_y
Pattern 8: NNS_x ***in*** NNS_y

<u>Where:</u>
NP - **Noun Phrase**
NP1 - Noun Phrase with head word as **Singular Noun**
NN - **Noun**
NNS - **Plural Noun**
$(term)_x$ - Noun/Noun Phrase with **Meronym** term
$(term)_y$ - Noun/Noun Phrase with **Holonym** term

Figure 5.3: Using hypernyms from *Patterns to identify meronym-holonym terms from a text corpus*

5. *Pattern based meronym-holonym relationship identification: Meronym-Holonym* relationship is a part-whole relationship where one word describes a part of another word. The words *nib* and *pen* hold a meronym-holonym relation where *nib* has the definition "the writing point of a pen"[1] and *pen* has the definition "a writing implement with a point from which ink flows"[2]. Similar to *Hearst Patterns*, a set of *Meronym Patterns* (shown in Figure 5.3) could be applied to the UMBC WebBase Corpus to learn the meronym-holonym terms from this corpus. These patterns are mentioned in the papers "Automatic Extraction of Hypernym Meronym Relations in English Sentences Using Dependency Parser" [Sheena, Jasmine, and Joseph 2016] and "Finding Parts in Very Large Cor-

[1] http://wordnetweb.princeton.edu/perl/webwn?s=nib
[2] http://wordnetweb.princeton.edu/perl/webwn?s=pen

pora" [Berland and Charniak 1999]. For example consider the instance "That keyboard is a part of a personal computer". If we apply the pattern "NP_x ***is a part of*** NP_y" to this instance, the meronym-holonym terms identified are *keyboard* and *personal computer*.

References

Aldine, A. I. A., M. Harzallah, G. Berio, N. Béchet, and A. Faour (2018). “EXPR at SemEval-2018 Task 9: A Combined Approach for Hypernym Discovery”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 919–923 (cit. on p. 104).

Banerjee, S. and T. Pedersen (2002). “An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet”. In: *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*. CICLing ’02. London, UK, UK: Springer-Verlag, pp. 136–145. ISBN: 3-540-43219-1. URL: http://dl.acm.org/citation.cfm?id=647344.724142 (cit. on p. 18).

— (2003). “Extended Gloss Overlaps As a Measure of Semantic Relatedness”. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI’03. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., pp. 805–810. URL: http://dl.acm.org/citation.cfm?id=1630659.1630775 (cit. on pp. 4, 19).

Berland, M. and E. Charniak (1999). “Finding Parts in Very Large Corpora”. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. ACL ’99. College Park, Maryland: Association for Computational Linguistics, pp. 57–64. ISBN: 1-55860-609-3. DOI: 10.

3115/1034678.1034697. URL: https://doi.org/10.3115/1034678.1034697 (cit. on p. 119).

Camacho-Collados, J., C. Delli Bovi, L. Espinosa-Anke, S. Oramas, T. Pasini, E. Santus, V. Shwartz, R. Navigli, and H. Saggion (2018). “SemEval-2018 Task 9: Hypernym Discovery”. In: *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*. New Orleans, LA, United States: Association for Computational Linguistics (cit. on pp. 2, 15).

Chklovski, T. and P. Pantel (2004). “VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations”. In: *Proceedings of EMNLP 2004*. Ed. by D. Lin and D. Wu. Barcelona, Spain: Association for Computational Linguistics, pp. 33–40 (cit. on p. 114).

Espinosa-Anke, L., J. Camacho-Collados, C. Delli Bovi, and H. Saggion (2016). “Supervised distributional hypernym discovery via domain adaptation”. In: *Conference on Empirical Methods in Natural Language Processing; 2016 Nov 1-5; Austin, TX. Red Hook (NY): ACL; 2016. p. 424-35.* ACL (Association for Computational Linguistics) (cit. on p. 5).

Fellbaum, C. (1998). *WordNet: An electronic lexical database (Language, Speech, and Communication). Cambridge, MA: The MIT Press* (cit. on p. 3).

Fu, R., J. Guo, B. Qin, W. Che, H. Wang, and T. Liu (2014). “Learning Semantic Hierarchies via Word Embeddings”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 1199–1209. URL: http://www.aclweb.org/anthology/P14-1113 (cit. on p. 93).

Han, L., A. L. Kashyap, T. Finin, J. Mayfield, and J. Weese (2013). “UMBC$_{E}BIQUITY-CORE: SemanticTextualSimilaritySystems$”. In: *Proceedings of the Second*

Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics (cit. on pp. 3, 10).

Hearst, M. A. (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora". In: *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*. COLING '92. Nantes, France: Association for Computational Linguistics, pp. 539–545. DOI: 10.3115/992133.992154. URL: https://doi.org/10.3115/992133.992154 (cit. on pp. 3, 22, 82, 91, 92).

Jurgens, D. and M. T. Pilehvar (2016). "SemEval-2016 Task 14: Semantic Taxonomy Enrichment". In: *Proceedings of the 1oth International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics: Association for Computational Linguistics (cit. on pp. 3, 12, 13).

Kilgarriff, A. and J. Rosenzweig (2000). "Framework and Results for English SENSEVAL". In: *Computers and the Humanities* 34.1, pp. 15–48. DOI: 10.1023/A:1002693207386. URL: https://doi.org/10.1023/A:1002693207386 (cit. on pp. 17, 42).

Lesk, M. (1986). "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone". In: *Proceedings of the 5th Annual International Conference on Systems Documentation*. SIGDOC '86. Toronto, Ontario, Canada: ACM, pp. 24–26. ISBN: 0-89791-224-1. DOI: 10.1145/318723.318728. URL: http://doi.acm.org/10.1145/318723.318728 (cit. on pp. 4, 18).

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781. arXiv: 1301.3781. URL: http://arxiv.org/abs/1301.3781 (cit. on pp. 3, 24, 92).

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *CoRR* abs/1310.4546. arXiv: 1310.4546. URL: http://arxiv.org/abs/1310.4546 (cit. on pp. 3, 27).

Onofrei, M., I. Hulub, D. Trandabat, and D. Gifu (2018). “Apollo at SemEval-2018 Task 9: Detecting Hypernymy Relations Using Syntactic Dependencies”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 898–902 (cit. on p. 104).

Rusert, J. and T. Pedersen (2016). “UMNDuluth at SemEval-2016 Task 14: WordNet’s Missing Lemmas”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1346–1350. URL: http://www.aclweb.org/anthology/S16-1211 (cit. on p. 3).

Seitner, J., C. Bizer, K. Eckert, S. Faralli, R. Meusel, H. Paulheim, and S. P. Ponzetto (2016). “A Large DataBase of Hypernymy Relations Extracted from the Web”. In: *LREC* (cit. on p. 113).

Sheena, N., S. M. Jasmine, and S. Joseph (2016). “Automatic Extraction of Hypernym Meronym Relations in English Sentences Using Dependency Parser”. In: *Procedia Computer Science* 93. Proceedings of the 6th International Conference on Advances in Computing and Communications, pp. 539–546. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2016.07.269. URL: http://www.sciencedirect.com/science/article/pii/S1877050916315150 (cit. on p. 118).

Toutanova, K. and C. D. Manning (2000). “Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger”. In: *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*. EMNLP ’00. Hong Kong:

Association for Computational Linguistics, pp. 63–70. DOI: 10.3115/1117794.1117802. URL: https://doi.org/10.3115/1117794.1117802 (cit. on p. 11).

A Appendix A

List of Definitions of OOV Lemmas from *SemEval 2016 Task 14 Semantic Taxonomy Enrichment*:

- **aeolian**: A mode used in Gregorian chant based upon the sixth tone of the major scale. In the key of C , the aeolian mode would be based on A , and would include A , B , C , D , E , F, G , A .
- **aquitard**: A geologic formation or stratum that significantly retards fluid movement.
- **bottle**: Nerve, courage.
- **celestine**: A member of a Roman Catholic monastic order, a branch of the Benedictines, founded in 1244.
- **cost avoidance**: An expense one has avoided incurring.
- **deadlift**: A weight training exercise where one lifts a loaded barbell off the ground from a stabilized bent-over position.
- **endograft**: An endoluminal graft.
- **finger lake**: an long, narrow lake occupying a glacial trough damned by morraine.

- **fining**: The process of adding clarifying agents such as isinglass, gelatin, silica gel, or Polyvinyl Polypyrrolidone (PVPP) to beer during secondary fermentation to hasten the precipitation of suspended matter, such as yeast, proteins or tannins.

- **graphics card**: A circuit board that controls and calculates the visuals on a computer monitor.

- **home care**: Health care provided in the patient's home by healthcare professionals.

- **immunoglobin**: Any protein that functions as an antibody.

- **inducement**: An introductory statement of facts or background information.

- **kill**: A hit immediately resulting in a point or out.

- **laser surgery**: Any surgery using a laser to cut tissue instead of a scalpel.

- **lovibond:** A scale used to measure color in grains and sometimes in beer.

- **palliative care**: A specialized area of healthcare that focuses on relieving and preventing the suffering of patients.

- **streak**: the color of a mineral's powder when scratched on a streak plate.

- **tablet**: A tablet computer, a type of portable computer.

- **tenolysis**: A surgical procedure in which a tendon is separated from its sheath.