HYPERNYM DISCOVERY

FINAL REPORT

Advisor : - Dr Manish Srivastava

Project Mentor : - Tanvi K

Team Number 19

Members : -
1) Akilesh Panicker 2021201081
2) Priyansh Upadhyay 2021201090
3) Subhra Chakravorty 2021201078

1. Project Overview:-

A hypernym describes a broader concept of the word, whereas, hyponym is more about defining finer granularity. In linguistics, a hyponym is a word or phrase whose semantic field is included within that of another word, its hyperonym or hypernym. In simpler terms, a hypernym “is a" type-of relationship with its hyponym. For instance, (orange,fruit) is a hypernymy relation, where orange is a hyponym and fruit is one of its hypernyms.

Due to its general representation ability of semantic relations, hypernymy becomes an essential concept in modern natural-language re- search, and hypernymy detection becomes a fundamental component in many of the applications, like Taxonomy construction, Semantic search, Textual entailment, and Question answering.Given a corpus and a target term (hyponym), the task of hypernym discovery consists of extracting a set of its most appropriate hypernyms from the corpus. For example, for the input word “dog”, some valid hypernyms would be “canine”, “mammal” or “animal”.

2. Datasets:-

i) Sem Eval 2018 Task 9 Dataset : - SemEval-2018 Task 9 is a sub-task of SemEval-2018 and focuses on Hypernym Discovery, which involves identifying a hypernym (i.e., a word that is more general than another word) for a given word. The task specifically involves identifying hypernyms for a given term from a list of candidate hypernyms.

The dataset for this task consists of training and test sets, each containing a list of target terms and a set of candidate hypernyms for each target term. The training set contains 14,900 target terms and 44,700 candidate hypernyms, while the test set contains 3,700 target terms and 11,100 candidate hypernyms. The data is available in several languages, including English, Spanish, Italian, and Dutch.

The goal of this task is to develop effective algorithms that can accurately identify hypernyms for a given term, which can be useful in various NLP applications such as text classification, information retrieval, and question answering.

ii) UMBC_Webbase Dataset :- The UMBC WebBase corpus is a large web-based corpus that was created at the University of Maryland, Baltimore County (UMBC) in the early 2000s. It contains over 3 billion words of text from a variety of online sources, including web pages, blogs, and online news articles. The corpus is intended to be representative of the language used on the internet, and it has been

used in a variety of research projects in natural language processing and computational linguistics.
The UMBC WebBase corpus is freely available for research purposes, and it has been used in a number of studies related to topics such as text classification, information retrieval, and semantic similarity. The corpus is often used as a benchmark dataset for evaluating the performance of natural language processing algorithms on web-based text data.

3. Implementation:-

The Sem Eval 2018 task 9 dataset had proper train,validation and test split so normal supervised approach was used for this dataset. But the umbc dataset was unlabelled so an unsupervised approach was used for this dataset. Given below are the implementation details for the two datasets:-

i) Sem Eval 2018:-

The steps involved in implementing this dataset was:-

- The english 1A dataset was used as train,test and validation split and a vocabulary of the words was created.
- The SkipGram model was used to generate embeddings. The SkipGram model of word2vec was used since this model combined with negative sampling is very efficient.
- The embeddings were trained using an Adam optimizer and it was stored for future use.
- We prepared the dataset in such a way that for each hyponym-hypernym pair we added a positive label 1 and for each positive pair found, we introduced 5 negative samples for training the data.
- Next this dataset was fed to the individual lstm and gru models and the best models were saved using the lowest validation loss.
- For evaluation of the model we assigned binary cross-entropy for each of the pair and discarded any score below 0.5. Then the top 15 pairs were selected and then compared with the test gold data for evaluation.

Hyperparameter Tuning:-

Model name	LSTM layers	FC layers	batch size	epochs	dropout	Accuracy
lstm_model_1	1	1	32	10	True	83.21
lstm_model_2	1	1	32	20	True	84.75
lstm_model_3	1	1	64	20	True	86.33
lstm_model_4	2	1	32	10	True	79.70
lstm_model_5	2	1	32	20	True	85.97
lstm_model_6	2	1	64	20	True	84.67
lstm_model_7	2	2	32	10	True	85.81
lstm_model_8	2	2	32	20	True	82.32
lstm_model_9	2	2	64	20	True	84.48

Model name	GRU layers	FC layers	batch size	epochs	dropout	Accuracy
gru_model_1	1	1	32	10	True	81.21
gru_model_2	1	1	32	20	True	82.75
gru_model_3	1	1	64	20	True	84.33
gru_model_4	2	1	32	10	True	77.70
gru_model_5	2	1	32	20	True	83.97
gru_model_6	2	1	64	20	True	82.67
gru_model_7	2	2	32	10	True	83.81
gru_model_8	2	2	32	20	True	80.32
gru_model_9	2	2	64	20	True	82.48

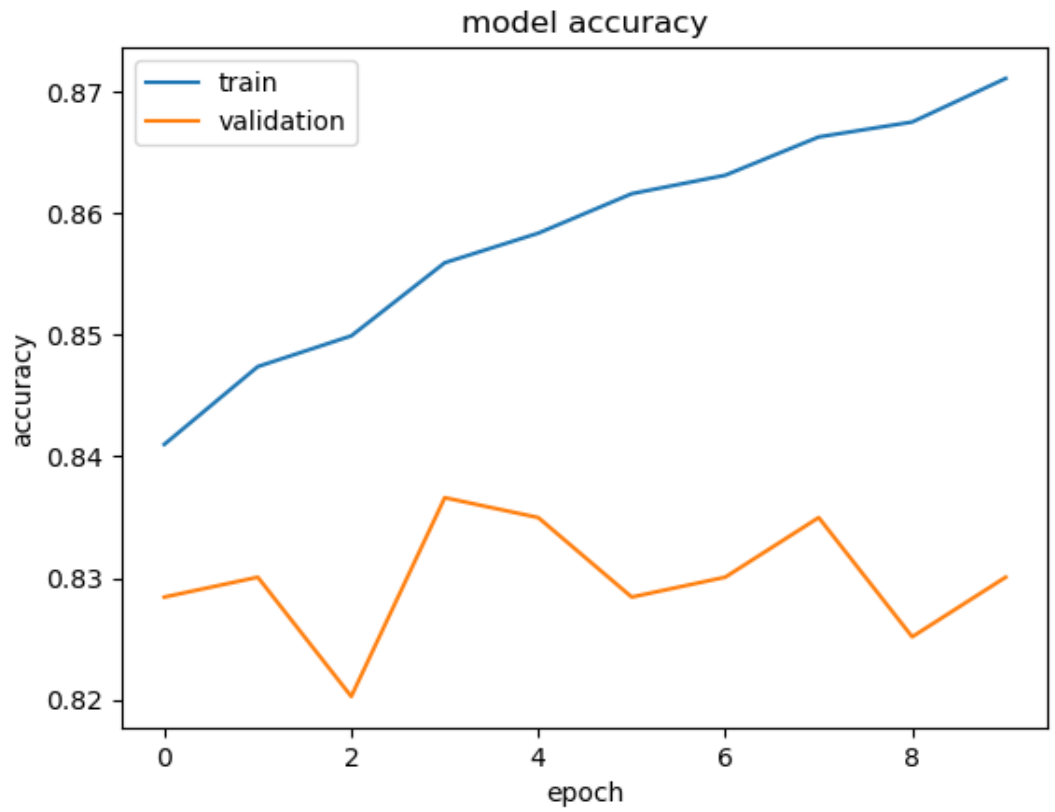
After the hyperparameter tuning the best models obtained were:-

a) **LSTM:-**
   LSTM Layers - 2
   Fully Connected Layers - 2
   Batch Size - 64
   Epochs - 20
   Accuracy - 86.54

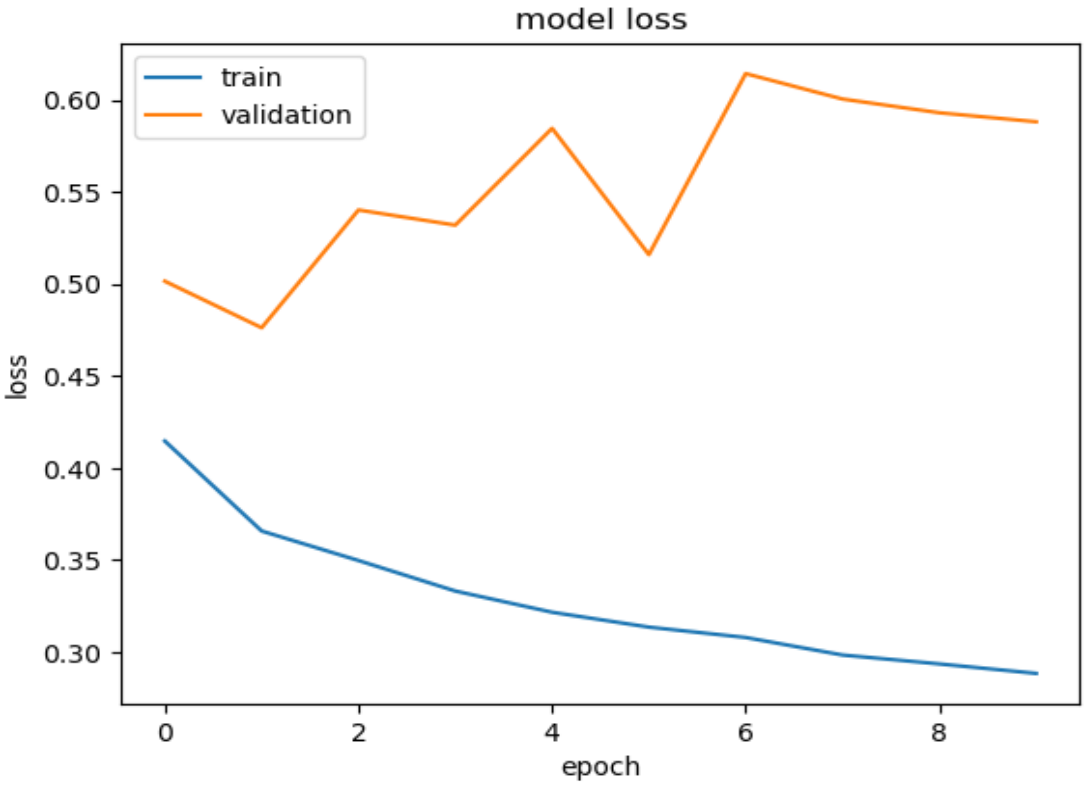
b) **GRU:-**
   GRU Layers - 1
   Fully Connected Layers - 1
   Batch Size - 64
   Epochs - 20
   Accuracy - 84.33

Accuracy Plots:-

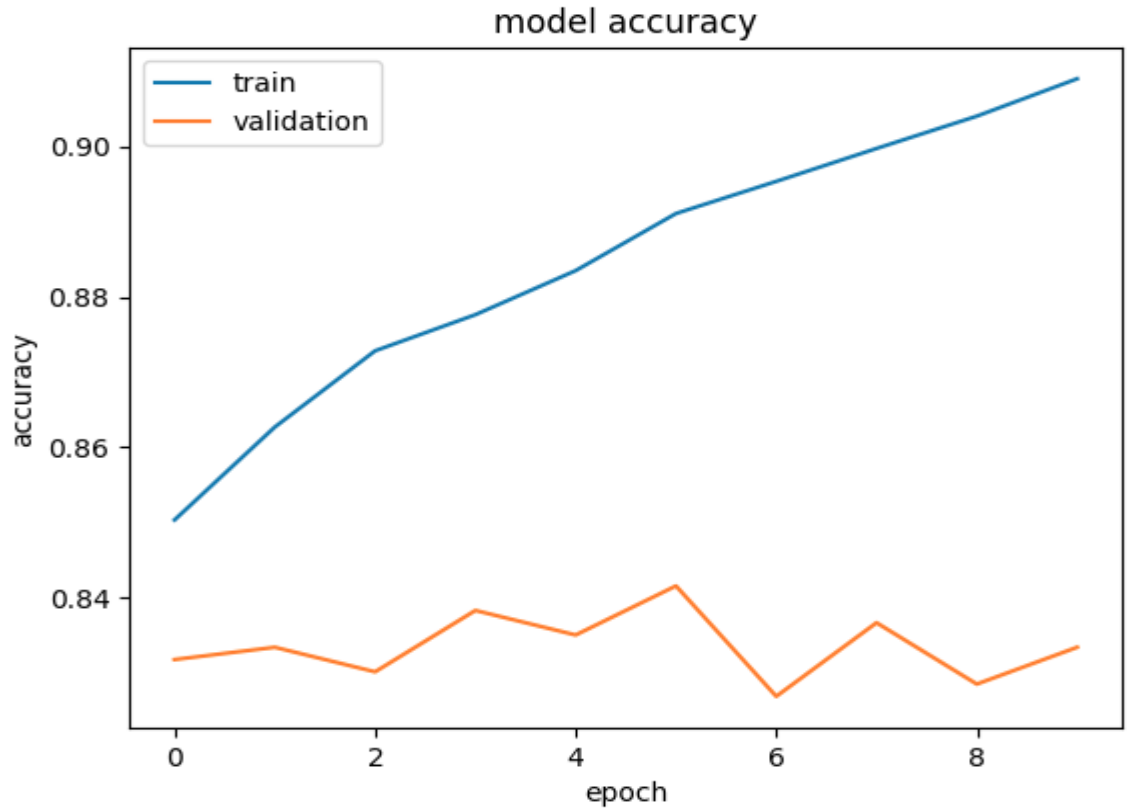
LSTM Accuracy Plot:-



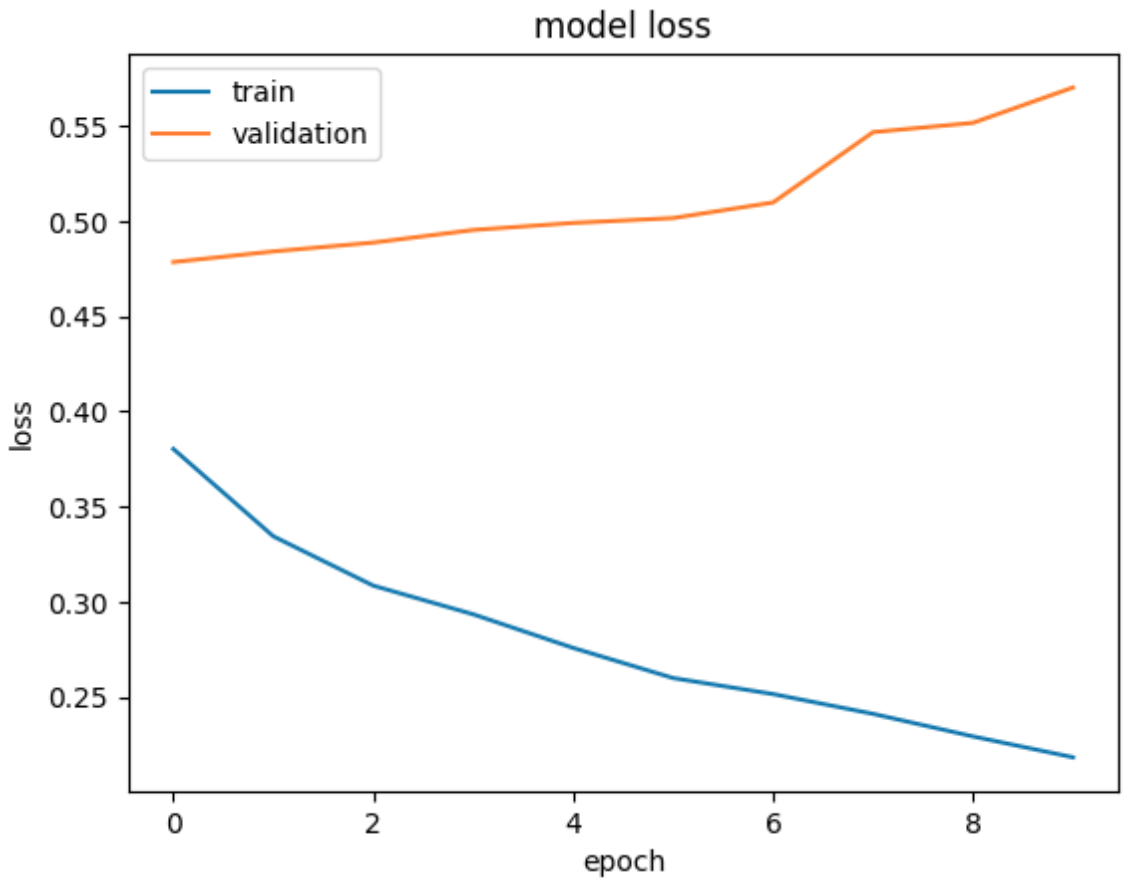
LSTM Loss Plot:-



GRU Accuracy Plot:-



GRU Loss Plot:-



ii) UMBC_Webbase:-

The steps involved in this unsupervised embeddings-based method of generating the hypernyms are as follows:

- Corpus preprocessing: The method begins by preprocessing a large text corpus to generate a vocabulary of words and their corresponding word embeddings. This can be done using popular pre-trained word embedding models such as Word2Vec or GloVe, or by training custom word embeddings on the corpus.

- Candidate hypernym generation: For each target word, the method creates a set of candidate hypernyms by identifying words that frequently co-occur with the target word in the corpus. This is done by calculating the pointwise mutual information (PMI) between the target word and all other words which co-occur in the same sentence in the corpus, and selecting the top N words with the highest PMI scores.

- Candidate hypernym filtering: To ensure that the candidate hypernyms are valid hypernyms of the target word, it removes any candidate hypernyms that are identical to the target word.

- Hypernym ranking: To rank the remaining candidate hypernyms, the method calculates the cosine similarity between their word embeddings and the embedding of the target word. The hypernym with the highest cosine similarity score is considered the most likely hypernym of the target word.

- Hypernym thresholding: To ensure that only meaningful hypernyms are selected, the method applies a threshold to the cosine similarity scores. Candidate hypernyms with a cosine similarity score below the threshold are discarded.

- Hypernym refinement: Finally, to improve the quality of the hypernym predictions, the method applies a refinement step that involves iteratively re-ranking the candidate hypernyms based on the cosine similarity scores between their word embeddings and the embeddings of the other hypernyms in the set. This ensures that the selected hypernyms are not only similar to the target word but also to other hypernyms in the set.

4. Challenges Faced:-

i) Semeval Dataset:-

- The main challenge faced for the SemEval Dataset was the very small size of the english 1A dataset. This meant that for most of the training steps, the model could not find any new hypernym pair since it had so few candidates to choose from. This meant that for both the models, the accuracy and loss values were pretty similar.

- Another challenge faced for this dataset was preparing this dataset for the model training. Even for this small vocabulary it took a lot time to prepare the dataset since for each positive hypernym pair we were finding 5 negative samples.

ii) UMBC Dataset:-

- A word may have multiple meanings depending on the context in which it appears. This can lead to the generation of incorrect candidate hypernyms or the ranking of the wrong hypernym.

- Since the method relies on co-occurrence statistics to identify candidate hypernyms, it may be prone to data sparsity issues, especially when dealing with rare or low-frequency words.

5. Analysis:-

i) Semeval Dataset:-

- The CRIM model developed by researchers at University of Montreal has the #1 benchmark for the model of this dataset. In the CRIM model, Convoluted Neural Network was used to learn the embeddings of the word pairs as feature maps. These were fed through a fully connected layer and were trained using a pairwise binary cross-entropy loss function. The CRIM model also incorporated several linguistic features such as part-of-speech tags and WordNet-based features to improve its performance.

- Some of the ideas we can implement to improve our own model are:-

    a) **Incorporating contextual information:** Instead of relying solely on the semantic similarity between the words, the model can be improved by considering the context in which the words appear. This can be done using contextualised word embeddings such as BERT or ELMo.

    b) **Combining multiple models:** Instead of relying on a single model, a combination of different models such as CRIM, LDA, and LSA can be used to improve performance. The output from each model can be combined to provide a more accurate and comprehensive set of hypernyms.
    c) **Use of multiple features:** The CRIM model uses multiple features such as word similarity, context similarity, and lexical patterns to identify hypernym relations. We can also consider using multiple features to improve the performance of our model.

ii) UMBC Dataset:-

- Unsupervised approach: The method is fully unsupervised, meaning it does not require any manually annotated data or external resources such as lexicons or knowledge bases. This makes it more scalable and applicable to a wider range of domains and languages.
- Distributional approach: The method uses a distributional approach to hypernym discovery, which involves analysing the distributional patterns of

words in a large text corpus. This approach has been shown to be effective for a wide range of natural language processing tasks.

- Embedding-based ranking: The method ranks the candidate hypernyms based on their similarity to the target word and other hypernyms in the set, using cosine similarity scores between their word embeddings. This is a common approach used in many natural language processing tasks.

6. References:-

1. Gabriel Bernier-Colborne and Caroline Barrière. 2018. CRIM at SemEval-2018 Task 9: A Hybrid Approach to Hypernym Discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 725–731, New Orleans, Louisiana. Association for Computational Linguistics.

2. Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 Task 9: Hypernym Discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana. Association for Computational Linguistics.

3. https://www.mdpi.com/2078-2489/11/5/268