**SageConv Graph Neural Network Model for Multimodal Regression using a Customised Datset**

Premanand Ghadekar1, Ankur Raut 2, Atharva Purohit3, Roshita Bhonsle4,

Vithika Pungliya5, Varad Ingale6

Vishwakarma Institute of Technology

**Abstract –** **Natural Language Understanding (NLU) has become increasingly important in recent years, with the advent of various applications such as chatbots, virtual assistants, and sentiment analysis. Graph neural networks (GNNs) have demonstrated promising results in NLU tasks because of their ability to grasp the intricate links between words and their contextual dependencies. In this system, making a graph structure from a customized dataset which is an input to a GNN and the effectiveness of the SAGEConv Graph Convolutional Network (GCN) in text classification has been explored. The proposed system uses text documents from Google Quest dataset and experiments with different configurations of the SAGEConv GCN to achieve the best performance. The results show that the SAGEConv GCN has an R2 score of 0.65. Findings highlight the potential of GNNs in NLU tasks and pave the way for further research in this area.**

**Keywords:** Graph Neural Network, Natural Language Understanding, MSE, SAGEConv,

**1. Introduction**

Graphs play a significant role in problem-solving because they represent the real-world entities in a way that makes them simple to analyze and study relationships from. In more general terms, a graph can be thought of as a mathematical structure that can be helpful to represent a set of objects (nodes) and the connections between them (edges) [1] G (V, E) is a common way to represent a graph, the nodes of the graph are represented by the symbol V and the edges are represented by the symbol E. Several real-world problems can be solved using graph-based solutions, such as prediction of drug-target interactions [2], classification of breast cancer subtypes [3], and even social network analysis [4].

A new technique known as the Graph Neural Network (GNN) has been introduced for analysis of data. This data can be structured or unstructured. Structured data, like a social network, contains inherent relationships and hence has a regular graph form. Unstructured data, like text and images, must be converted to structured data in order to find the underlying relationships.

Unlike traditional neural networks, GNNs preserve state information to capture the attributes of neighboring nodes, making them highly effective for graph-based tasks. GNNs can be used to predict class labels as well as the value of a node. This can be achieved by an iterative process of message passing and data exchanging [5]. Graph Encoders and Decoders, Graph LSTMs, GCNs (Graph Convolutional Networks), and GATs (Graph Attention Networks) are some other types of GNNs [6].

In the domain of NLP, GNNs have recently become a potent tool. RNNs and CNNs, two types of traditional NLP models, are good at identifying local associations within a phrase but fail to model more intricate structures like grammar and discourse effectively [7].

A question-answering system is a very common application of natural language processing (NLP) that aims to automatically respond to questions posed. Several useful uses of question-answering systems include chatbots, search engines and voice assistants [8]. In order to build robust question-answering systems, analyzing the subjective nature of human responses is necessary. In the proposed work, a predictive model is developed using Graph Neural Network to capture not only the intricate relationships between the words but also their context in the question-answer pairs. By leveraging this graphical structure of the data, the proposed GNN model aims to make more accurate predictions of subjective aspects of the question-answering task.

Following is the structure of the paper, Section 1- Introduction gives a quick overview of graphs, GNNs, and question-answering. The literature on the application of GNNs to natural language processing is reviewed in Section 2. Methodology of the proposed work including – dataset used, proposed flow diagram, creation of the graph structure from text data, training and architecture of the model, and model evaluation is discussed in Section 3. The analysis and results of the experiments are presented in Section 4. The work is briefly summarized in Section 5, which also identifies some of the most potential areas for further study in this area.

**2. Literature Review**

In order to find a default approach to handle NLP tasks, this paper introduces the BERT model and the traditional NLP strategy, where a machine learning model is trained using the features extracted with TF-IDF. It also makes predictions regarding the behavior of BERT with respect to these techniques. In four different NLP scenarios, it was shown that BERT performed better than the conventional NLP methodology, offering empirical support for its superiority in typical NLP challenges compared to traditional techniques [9].

BERT is used in this study done by the authors to address a part of the issue of moderating activities on Question-Answering websites. Twenty subjective parameters of Questions on Question Answering websites have been predicted using BERT.  Computers have a difficult time predicting subjective features, but research showed that transfer learning from transformers can benefit prediction. The BERT Model successfully predicted target values with an MSE of 0.046 [10].

Multiple DL Frameworks - RNN, CNN, LSTM, Bi-LSTM based autoencoders have been used to extract the features from the question-answer text. Relevant parts of the survey include - Question Quality prediction, Answer Quality Prediction. about the various community question-answering platforms - Yahoo answers!, Stack Exchange, Stack Overflow and analyzes the quality of the question and answers. Different textual features that have been extracted - answer length, objectivity, similarity, keyword density, sentiment polarity of the answer. A multi-modal deep belief network is proposed in one of the papers surveyed to evaluate the answer of a doctor/physician, achieving an AUC value of 97.8% [11].

A novel network structure that has considerable potential for text and word classification. The bipartite graph is used in this paper to convert raw data into the graph. The method that makes use of the GNN seeks to obtain intricate data that can be conveniently stored in the graph.. The benchmark CORA dataset was used in this study [12].

The most amazing feature of Transformers is the direct application of its beautiful model architecture to the original text sequence, which sets the inputs apart from the transformer model's graph data modeling. A major disadvantage is that transformers are difficult to employ on data that is complex in nature, such as graphs. However, GNNs are more open-ended model architectures that work with graph data directly, but they necessitate that the user construct the data using either knowledge specific to the domain of the problem or different techniques to model the graph structures. The most popular models today combine the advantages of GNNs and Transformers by implementing a structure-aware self-attention mechanism. [13].

**3. Methodology**

**3.1 Dataset Description**

The Google Quest Q&A Labeling Dataset has been used in the proposed work to build a predictive model using GNN that can predict the subjective aspects of the question-and-answer pairs. The dataset contains 41 columns of which 30 are the target columns containing different attributes of question and answers to be predicted [14].

**3.2 Flow Diagram of Proposed System**

Fig 1 below illustrates an overview of the working of the system using a flow diagram of the proposed system. This provides an overview of data creation, model building and evaluation.

**Diagram

Description automatically generated**

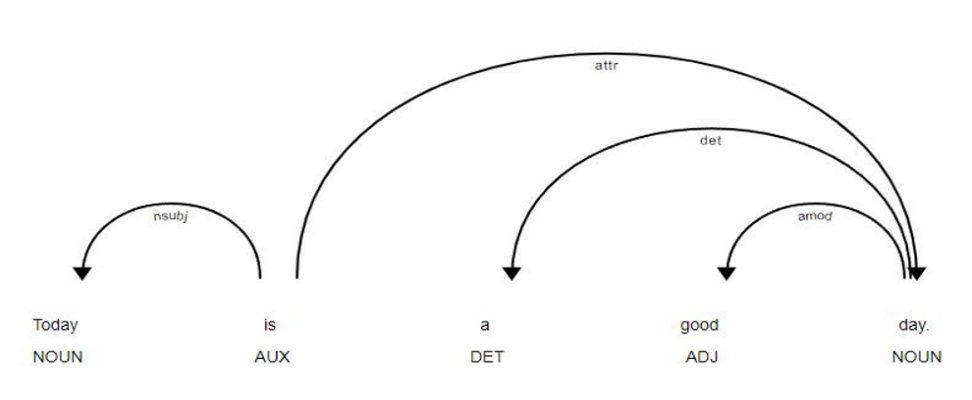
**Fig 1**. Proposed System of Natural Language understanding using GNN.

Algorithm of proposed system

1. Read the Google Quest Dataset in csv format .
2. Clean and preprocess the text data. Represent each text document as a graph, where each word is a node and2. edges are defined based on their co-occurrence in the document.
3. Convert each graph into a feature matrix X and adjacency matrix A, where X is a matrix of node features and A is a matrix of edge weights.
4. Define the SAGEConv model architecture.
5. Create training, validation, and testing sets from the dataset. Utilizing backpropagation and gradient descent to minimize the loss function, train the SAGEConv model using the training data.
6. Analyze validation set's performance using the model to tune the hyperparameters and prevent overfitting
7. To adjust the hyperparameters and avoid overfitting, analyze the validation set's performance using the model.
8. Once GNN model is trained, predictions can be made on new text data.

**3.3 Data Creation -**

This section discusses the detailed approach to create data suitable for a Graph Neural Network from a custom dataset. Initially the text is pre- processed by converting it into lowercase and removing newline characters. The data to a Graph Neural Network requires X, which are the input features, Y which are the labels and Edge Indices, which is the relationship between the graph’s nodes. To generate input features, the data corpus is tokenized to give every unique word a number token and using these tokens the words are converted into numeric sequences. Every sentence has 30 output labels.



**Fig 2**. Semantic Tree Structure

To find the edge indices, Spacy\_Doc object [15] is used to convert the data into a semantic tree structure as seen in Figure 2 above. The figure shows the relationship between the sentence's words using a tree data structure. Next this tree is converted to an adjacency matrix where 1 denotes the relationship between a child and parent node and 0 denotes that there is no such relationship. The Table 1 below shows the adjacency matrix for the semantic tree below.

**Table 1.** Adjacency matrix for the Semantic Tree

| **Tokens** | **Today** | **is** | **a** | **good** | **day** |
| --- | --- | --- | --- | --- | --- |
| **Today** | 0 | 1 | 0 | 0 | 0 |
| **is** | 1 | 0 | 0 | 0 | 1 |
| **a** | 0 | 0 | 0 | 0 | 1 |
| **good** | 0 | 0 | 0 | 0 | 1 |
| **day** | 0 | 1 | 1 | 1 | 0 |

Then the adjacency matrix is converted to a Co- Occurrence Matrix to find the edge indices. The Co- Occurrence matrix is in the form of [[X1, X2.. Xn], [Y1, Y2, .. Yn]], where Xn represents the row coordinate of 1 in the adjacency matrix and Yn represents the column coordinate of 1. [[0, 1, 1, 2, 3, 4, 4, 4], [1, 0, 4, 4, 4, 1, 2, 3]] is the Co-Occurrence matrix for the adjacency matrix above Table 1.

The X, Y and Edge Indices are then combined into a Data object. Example- DataBatch (x = [20015, 1], edge\_index = [2, 38776], y = [128, 30], batch = [20015], ptr = [129]).

**3.4 Model Architecture**

The model is initialized with a vocabulary size vocab\_size and an output dimen-sions output\_dims. The input data is first passed through an embedding layer, which maps the input data to a continuous vector space with dimension out-put\_dims. This continuous representation of the data is then passed through three GNN layers (gc1, gc2, and gc3), which use the SAGEConv operator to extract features from the graph structure.

) (1)

(2)

) (3)

In the above equations (1), (2), (3) [16], hvk represents the kth layer hidden representation of node v, which is calculated as the concatenation of the max-pooled neighbor representations and the concealed representation of node v in the preceding layer.. Here u belongs to the set of v. The pool layer computes the global representation of the graph by max-pooling the output representations of all nodes in the final layer. Finally, the output probability distribution over the labels is computed using a fully connected layer with weights ‘W’ and bias ‘b’, followed by a softmax activation function.

This is followed by three TopK pooling layers (pool1, pool2, pool3) that select the top K nodes and aggregate the node’s features. Then Concatenation is done with the pooling layers' output and passed through two fully connected linear layers (lin1 and lin2). The first linear layer lin1 maps the input features to 64 dimensions, while the second linear layer lin2 maps the features to the output dimension, which is 30 as the regression for 30 classes is being evaluated. The activation function used in both linear layers is ReLU. Dropout is also used after the first linear layer to prevent overfitting during training. The loss function used in training the mode is mean squared error.The optimizer used for training is Adam, which is a popular gradient-based optimization method. The learning rate scheduler used in the model is ReduceLROnPlateau. The figure 3 below shows the model architecture.

Diagram

Description automatically generated

**Fig 3**. Model Architecture of SAGEConv Model.

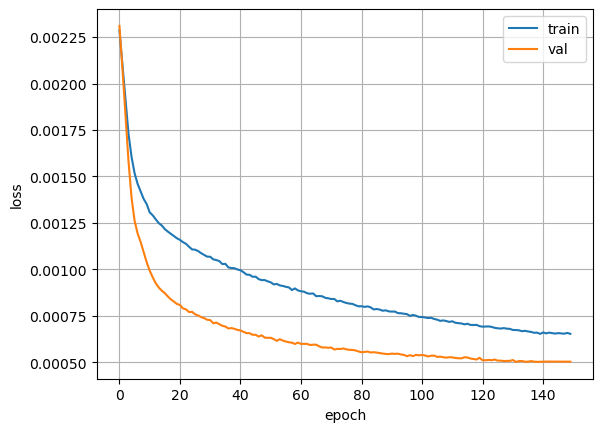
**4. Results And Discussion**

This section reports the proposed work's experimental results. The models have been evaluated using two different metrics namely - R2 Score and Mean Squared Error (MSE). Table 2. Below shows the comparison between the performance of the three different GNN models tried using R2 score as the metric.

**Table 2.** Comparison of the models based on R2 score and MSE.

| **Model** | **Training**  **R2 score** | **Validation**  **R2 score** | **Training**  **MSE** | **Validation**  **MSE** |
| --- | --- | --- | --- | --- |
| GCN | 0.63 | 0.60 | 0.000487 | 0.00055 |
| ResGatedConv | 0.40 | 0.44 | 0.00079 | 0.000786 |
| SAGEConv | 0.65 | 0.644 | 0.00045 | 0.00050 |

Based on the data presented in the tables above, it can be inferred that the SAGEConv model showed better performance in comparison to the other two models. Loss curves of training and validation of the SAGEConv model are presented below in Figure 4.

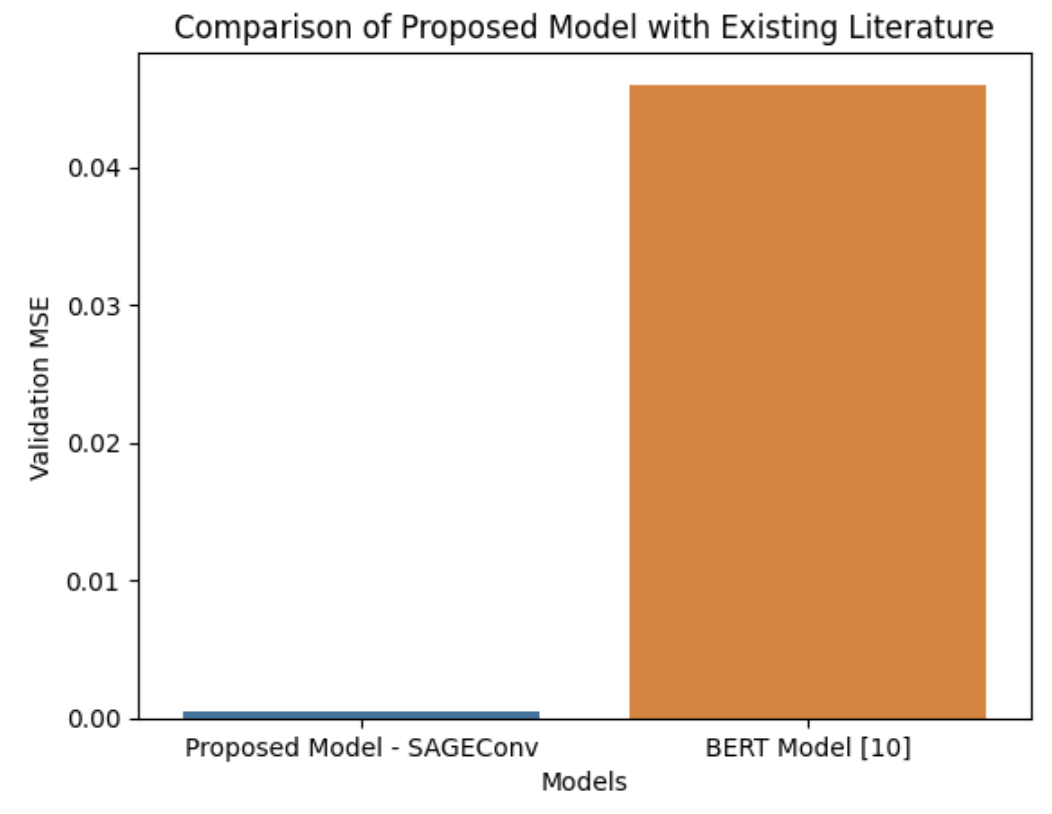


**Fig 4.** SAGEConv model training and validation loss curves.

Finally, in Table 3. A comparison is drawn between the BERT model proposed in [10] that was used to predict values for the target columns of the Google Quest Q&A Labeling Dataset.

**Table 3.** Comparison of the proposed model with Existing Literature

| **Model** | **Validation MSE** |
| --- | --- |
| **Proposed Model - SAGEConv** | 0.00055 |
| BERT Model [10] | 0.046 |



**Fig 5.** Bar plot of Validation MSE of the proposed SAGEConv model and the BERT model proposed in [10].

Based on the comparison of the MSE values between the proposed model and the architecture proposed in reference [10] as seen in Figure 5, it can be inferred that the proposed model outperforms the latter architecture.

**5. Conclusion**

This study explored the effectiveness of Graph Neural Networks (GNNs) in Natural Language Understanding (NLU) tasks. A customized dataset created from text documents was used in the Google Quest dataset and experimented with different configurations of the SAGEConv Graph Convolutional Network (GCN) for text classification. The results demonstrate the potential of GNNs in capturing complex relationships between words and their contextual dependencies in NLU tasks. The SAGEConv GCN performed well with an R2 score of 0.65 and an MSE score of 0.0004, indicating its effectiveness in text classification. These findings pave the way for further research and development of GNN-based models in NLU applications such as chatbots, virtual assistants, and sentiment analysis. The proposed system has significant implications in improving the accuracy and efficiency of NLU tasks, ultimately benefiting various industries and domains.

**References**

1. J. Zhou et al., ‘‘Graph Neural Networks: A Review of Methods and Applications,” pp. 1–22, 2018, [Online]. Available: http://arxiv.org/abs/ 1812.08434
2. Zhou, Y., Zhou, B., Liu, Z., & Guo, Z. (2018). Deep learning in drug discovery: successes, challenges, and opportunities. Computational and Structural Biotechnology Journal, 16, 325-338. doi: 10.1016/j.csbj.2018.06.001
3. Gao, Y., Wang, P., Mao, Z., Shen, X., & Shen, F. (2020). A graph convolutional neural network-based method for breast cancer subtype classification. Computers in Biology and Medicine, 126, 104041. doi: 10.1016/j.compbiomed.2020.104041
4. Newman, M. E. J., Girvan, M., & Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, 99(12), 7821-7826. doi: 10.1073/pnas.122653799
5. Gupta, A., Matta, P., & Pant, B. (2021). Graph neural network: Current state of Art, challenges and applications. *Materials Today: Proceedings*, *46*, 10927-10932.
6. Z. Zhang, P. Cui and W. Zhu, "Deep Learning on Graphs: A Survey," in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 249-270, 1 Jan. 2022, doi: 10.1109/TKDE.2020.2981333.
7. D. W. Otter, J. R. Medina and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 604-624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.
8. Soares, M. A. C., & Parreiras, F. S. (2020). A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, *32*(6), 635-646.
9. González-Carvajal, S., & Garrido-Merchán, E. C. (2020). Comparing BERT against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.
10. I. Annamoradnejad, M. Fazli and J. Habibi, "Predicting Subjective Features from Questions on QA Websites using BERT," 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 2020, pp. 240-244, doi: 10.1109/ICWR49608.2020.9122318.
11. Roy, Pradeep Kumar, Sunil Saumya, Jyoti Prakash Singh, Snehasish Banerjee, and Adnan Gutub. "Analysis of community question‐answering issues via machine learning and deep learning: State‐of‐the‐art review." *CAAI Transactions on Intelligence Technology* (2022).
12. Wang, Jingyu. "Text Classification Based on GNN." In *2020 International Workshop on Electronic Communication and* Artificial *Intelligence (IWECAI)*, pp. 94-97. IEEE, 2020.
13. Wu, Lingfei, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. "Graph neural networks for natural language processing: A survey." *Foundations and Trends® in Machine Learning* 16, no. 2 (2023): 119-328.
14. ‘Google QUEST Q&A Labeling’. Kaggle, 2019. <https://www.kaggle.com/competitions/google-quest-challenge/>
15. <https://spacy.io/api/doc>
16. Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Advances in neural information processing systems 30 (2017).