Contrastive Learning with Heterogeneous Graph Attention Networks on Short Text Classification

1st Zhongtian Sun
Department of Computer Science
Durham University
Durham, UK
zhongtian.sun@durham.ac.uk

2nd Anoushka Harit
Department of Computer Science
Durham University
Durham, UK
anoushka.harit@durham.ac.uk

3th Alexandra I. Cristea
Department of Computer Science
Durham University
Durham, UK
alexandra.i.cristea@durham.ac.uk

4rd Jialin Yu
Department of Computer Science
Durham University
Durham, UK
jialin.yu@durham.ac.uk

5th Lei Shi
Department of Computer Science
Durham University
Durham, UK
lei.shi@durham.ac.uk

6th Noura Al Moubayed
Department of Computer Science
Durham University
Durham, UK
noura.al-moubayed@durham.ac.uk

***Abstract*—Graph neural networks (GNNs) have attracted extensive interest in text classification tasks due to their expected superior performance in representation learning. However, most existing studies adopted the same semi-supervised learning setting as the vanilla Graph Convolution Network (GCN), which requires a large amount of labelled data during training and thus is less robust when dealing with large-scale graph data with fewer labels. Additionally, graph structure information is normally captured by direct information aggregation via network schema and is highly dependent on correct adjacency information. Therefore, any missing adjacency knowledge may hinder the performance. Addressing these problems, this paper thus proposes a novel method to learn a graph structure, NC-HGAT, by expanding a state-of-the-art self-supervised heterogeneous graph neural network model (HGAT) with simple neighbour contrastive learning. The new NC-HGAT considers the graph structure information from heterogeneous graphs with multilayer perceptrons (MLPs) and delivers consistent results, despite the corrupted neighbouring connections. Extensive experiments have been implemented on four benchmark short-text datasets. The results demonstrate that our proposed model NC-HGAT significantly outperforms state-of-the-art methods on three datasets and achieves competitive performance on the remaining dataset.**

***Index Terms*—Semi-supervised learning, graph neural network, contrastive learning, text classification**

I. Introduction

Text classification is a fundamental task in natural language processing (NLP), which can be applied to a variety of downstream tasks, such as question answering, machine translation and sentiment analysis [1]. The representation learning ability of textual features is a leading cause of the high performance of text classification models. Consequently, it is a pressing need to study how to extract textual features more effectively. Recently, graph neural networks (GNNs) have been increasingly applied to text classification tasks, due to their advantages of dealing with complex semantics and topological information, by modelling texts as graph structure [2]. Graphs in such studies [3], [4] usually consist of different types of nodes, which represent words or documents, and edges, to indicate relations. These graphs are known as heterogeneous information networks (HIN) or heterogeneous graphs. Different from most existing studies that focus on *long* text classification, we mainly focus on *short* text classification, as our daily communication is increasingly completed via short texts, such as tweets, messenger and online comments. Thus it has become more important to study this field.

On the one hand, most existing studies of GNNs on text classification tasks are trained in a semi-supervised manner, the same as the vanilla Graph Convolution Network (GCN) [5] requiring a large set of labelled data, which cannot be obtained in many real-life scenarios. Therefore, the shortage of labelled data may undermine the performances of graph neural network models on classification tasks, particularly with large scale data [6], [7]. On the other hand, although a GCN can encode local topological properties, it may fail to fully capture the global structural information [8]. To be more specific, existing methods for text classification mainly learn direct neighbourhoods and the associated textual features by supervised information aggregation. They may not be able to incorporate the high-order, rich relations among texts [9], and hence are not robust when the connections among nodes are noisy or missing [10], as is the case with the data considering only 40 labels known per class for training in this paper.

To address the above problems, we propose to integrate neighbouring contrastive learning (NC) with the heterogeneous graph attention network (HGAT), forming NC-HGAT. HGAT is the state-of-the-art work for text classification tasks, proposed by [6] to embed HIN with a dual-level attention mechanism for both nodes and relations. Contrastive learning can learn intrinsic and transferable topological information, enhance the performance of graph neural networks [11], and is widely applied in NLP tasks for pre-training [12]. NC learning enables our proposed model to transform k^{th} structural-aware features, without using direct message-passing modules, and hence improve robustness, despite missing connections between words during inference [10], when labelled data is limited.

The contributions of the paper are summarised as follows:

- To the best of our knowledge, this study is *the first attempt* to apply contrastive learning with a heterogeneous graph neural network to short text classification tasks.
- We propose to use a simple MLP to learn the neighbouring information without direct message-passing, which can be easily applied to existing graph neural network models [10] to text classification.
- Experimental results on three of the four datasets analysed show NC-HGAT outperforming the state-of-the-art on short text classification with limited labelled data, and it also delivers a competitive result on the fourth dataset.

II. Related Work

Text Classification. Extensive studies have been conducted on text classification, such as traditional machine learning using manually designed features [13], convolutional neural networks [14] and recurrent neural networks [15]. Recently, graph neural networks (GNNs) have shown promising performance on text classification, as text can be modelled as edges and nodes in a graph structure. For example, TextGCN [16] applied the vanilla GCN to heterogeneous graphs, on graphs built from a text corpus, and gained improved results. [6] proposed a novel heterogeneous graph attention networks model (HGAT) with a dual attention mechanism, to consider more relations between different nodes. Recently, [17] introduced an orphan category to HGAT, to remove unrelated stop-words, which improves classification accuracy. [9] also incorporated the attention mechanism with deep diffusion layers, to enrich the context information of texts. [18] constructed hypergraphs for text classification to capture high-order interaction between words. However, these methods all relied heavily on the direct message-passing function to learn node-wise feature transformation, and the performance decreased when labelled training data was limited. We thus propose, *for the first time*, to the best of our knowledge, to solve the problem by applying contrastive learning of graph structure in text classification tasks.

Contrastive Learning. Contrastive learning is a discriminative approach, which aims to learn embeddings of objects, shorten the distance between similar entities, and lengthen the gap among dissimilar entities [19]. It is naturally in line with the classification objective [20] and has increasingly been applied to computer vision and NLP tasks. Contrastive learning can be used in both self-supervised representation learning [21]–[23] and supervised learning [12], [24], [25]. In NLP tasks, [26] performed contrastive learning on adversarial samples, to improve text classification, and [27] applied it to obtain more effective embeddings of words, to mitigate the problem of data scarcity. For graph learning, contrastive learning between global and local objects can better capture structural information [28]. Comprehensive information about contrastive learning can be found in [29] and [19].

III. Methodology

In this section, we will introduce our NC-HGAT model, a merger of the HGAT model [6] and neighbouring contrastive (NC) learning adapted from the Graph-MLP model [10].

A. Construct Heterogeneous Graph from Data

Here we apply the same approach as in [6], to construct heterogeneous graphs from texts. This is briefly illustrated below, for clarity. Specifically, a heterogeneous graph can be denoted as $G = (V, \epsilon)$, where V is the union of entities E, topics T, and short texts S; and $\epsilon = \epsilon_1, \epsilon_2, ...$ is the set of edges representing the relations between them, as shown in Figure 1.

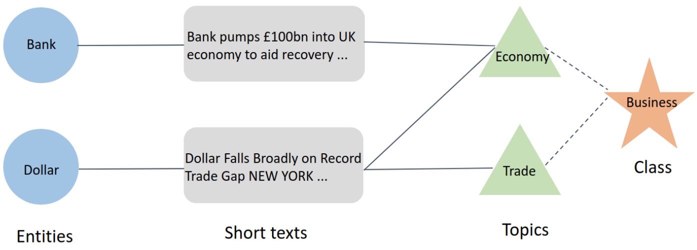
Bank
Bank pumps £100bn into UK economy to aid recovery ...
Economy
Business
Dollar
Dollar Falls Broadly on Record Trade Gap NEW YORK ...
Trade
Class
Entities
Short texts
Topics

Fig. 1. Example Snippet from a Heterogeneous Graph Structure

Each short text (document) is assigned to a number of top M possible topics $(t_1, t_2...t_M)$ using LDA [13] and the entities E are mapped to Wikipedia via TagME[1], an entity linking tool. Edges will be created if an entity e is contained in a document s or the document is assigned to a topic. Considering entities, short texts and topics in Figure 1, it is highly likely that the documents in the figure would be classified with their correct label as "Business". The overall structure of our model is shown in Figure 2, where we apply a HGAT model (circled by the orange color dash line) to construct text graphs and utilise an MLP-based model to update features, then calculate the similarity among nodes within the same input batch, based on an adjacency matrix. The details are explained in the later subsections III-B, III-C and III-D.

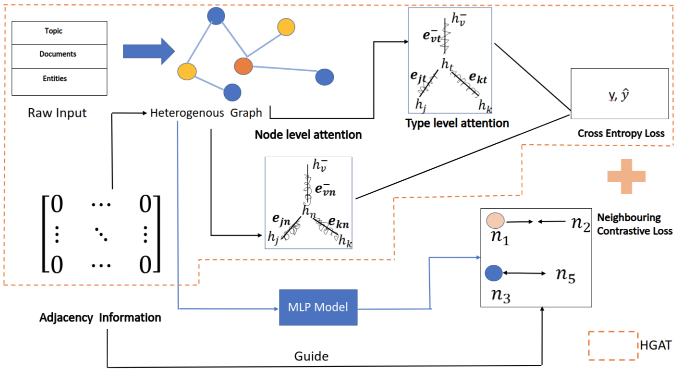

Fig. 2. Illustration of our NC-HGAT model

[1] https://sobigdata.d4science.org/web/tagme/tagme-help

B. HGAT

Compared with TextGCN [16], which directly applies GCN to different subgraphs, HGAT introduces a dual attention mechanism: type-level attention and node-level attention, to learn the relative influence of the different types and neighbouring nodes on the target node during information aggregation [6]. The type-level attention a_t is calculated as:

$$a_t = softmax(\sigma(\mu_t \cdot [h_i||h_t])) \tag{1}$$

where σ is a LeakyReLU activation, μ_t denotes the attention of the type t of the node, and operation $||$ is a concatenation. h_i and h_t are a specific node and type embedding, respectively. Then, a softmax function is applied, to normalise all types of neighbours of node i. The node level attention a_n is formulated based on the type level attention a_t from Equation (1):

$$a_n = softmax(\sigma(v^T \cdot a_t[h_i||h_{j'}])) \tag{2}$$

where v denotes the attention vector, and $h_{j'}$ is the neighbour embedding of node i with type t, is further concatenated with the central node h_i. The type attention weight a_t is obtained from 1. The two attention mechanisms are then integrated into the heterogeneous graph convolution, to update the embedding of nodes in the next layer:

$$H^{l+1} = \sigma(\sum \hat{A}_t \cdot H_t^l \cdot W_t^l) \tag{3}$$

where $\hat{A}$ is an adjacency matrix with type t edges, H_t^l represent the features of type t neighbouring nodes of the target node, and W_t^l is a weight matrix.

C. Neighbouring Contrastive Learning

The neighbouring contrastive learning is implemented by calculating the *contrastive loss* for node i. The idea behind it is that neighbouring documents are more likely to have the same class label. The node feature X will pass two linear layers with activation σ and layer normalisation LN, and a dropout in between to avoid over-fitting, as in [10]:

$$Z = W^1[Dropout(LN(\sigma(XW^0)))] \tag{4}$$

where W^1 and W^0 are the weight matrices of two layers. The number of linear layers could be set differently (from 1-7) as analysed in IV-D. Next, the embedding Z will be used to calculate the neighbouring *contrastive loss*:

$$loss_{NC} = -log\frac{\sum_j \lambda exp(sim(z_i, z_j)/\eta)}{\sum_k exp(sim(z_i, z_k)/\eta)} \tag{5}$$

where λ is a connection measure of node j and i and is nonzero only when the node j is within the k-hop neighbourhood of node i; sim is the cosine similarity, and η is the learning 'temperature' parameter.

D. Model Training

Considering the limited labelled data provided, we only use 40 labelled documents per class as training data, in line with previous work [6], [17]. We firstly use the HGAT model to build graphs from the text corpus and learn the representation of nodes with the dual-level attention mechanism. At the same time, we use the MLP-based model to learn more graph structure information, without an explicit message-passing function. To be more specific, the k-hop neighbours are considered more similar to the target node, where this k^{th} power of the neighbouring information is in the range of [1,2,3,4,5,6,7]. If the neighbouring node is not a k-hop of the target node, the neighbours' information is considered zero. Then, we calculate the neighbouring *contrastive loss*, $loss_{NC}$.

$$loss_{total} = loss_{NLL} + \beta * loss_{NC} \tag{6}$$

The total loss of our model 6 is the sum of the conventional negative log-likelihood loss $loss_{NLL}$ and the contrastive loss, $loss_{NC}$. β is a coefficient parameter to balance the total loss. The gradient descent algorithm is applied to optimise this total loss.

IV. Experiments

A. Dataset

We use the same four benchmark short text datasets as in [6], for a fair comparison to prior work. The movie review dataset (MR) [30] has 5,331 positive and 5,331 negative reviews, each of which is one sentence. Twitter, a sentiment classification dataset provided by the NLTK library of Python, contains 5,000 positive and negative tweets, respectively. Ohsumed is a bibliographic database provided by [31] where a graph convolution network model is applied for text classification. Each of the 7,000 documents we use is labelled with 23 types of diseases. We use 3,357 documents in the training set and the remaining in the test set. AGNews are randomly selected 6,000 news items from [32], which are classified into four classes: world, sports, business and sci/tech.

TABLE I
Dataset summary

Datasets	Docs	Token	Entities	Classes
MR	10662	7.6	1.8(76%)	2
Twitter	10000	3.5	1.1(63%)	2
AGNews	6000	18.4	0.9(72%)	4
Ohsumed	7400	6.8	3.1(96%)	23

B. Baselines and Experiment Settings

Baselines. We consider three widely applied NLP models and other three graph neural network models, applied as baselines for text classification. The parameter settings of all baseline models are the same as in [6], [17].

SVM + TFIDF and SVM + LDA are conventional machine learning classifiers, using classic features, including TF-IDF and LDA features [13], [33].

BERT, deploying a bidirectional Transformer encoder [34], is a widely-applied model in NLP. The model (Bert-base) has been fine-tuned as in [17].

TextGCN is the first study that applies GCN to text by building heterogeneous graphs from a text corpus [31].

HAN considers the importance of both node and meta-path, by introducing an attention mechanism into the heterogeneous graph neural network [16].

HGAT integrates a dual attention mechanism into a heterogeneous information network [6], [17], representing the current state-of-the-art on short text classification tasks.

Experimental Settings. We implement our model with Pytorch 1.10.2 and CUDA 10.2. The hyper-parameters of NC-HGAT are mainly borrowed from the experiments of HGAT [6] and Graph-MLP [10]. 40 labelled documents per class are randomly selected and split equally into training and validation sets. We use two layers and the number of hidden units is 512, learning rate 0.005, with an 80% dropout rate at each layer. The dimension of pre-trained word embeddings is set to 100. The k^{th} power of the adjacency matrix, the temperature parameter η and the coefficient balance parameter β are set using grid search. The range of η and β are [0,1,2] and [0.5, 1.0, 2.0, 3.0], respectively.

C. Experimental Results

Table II and Table III show the average classification performance of different models on the four benchmark datasets. The proposed model NC-HGAT outperforms all baselines on three datasets, demonstrating the effectiveness of the neighbouring contrastive learning with the heterogeneous graph attention network on short text classification. The improvement made by our proposed model with respect to the Ohsumed dataset in terms of the F1 score is statistically significant based on the t-test ($p < 0.05$).

One possible reason is that Ohsumed has the most classes from our four considered datasets (see Fig. 3). It is also not a uniform distribution, while the classes are equally distributed in the MR, Twitter and AGNews dataset. This shows that contrastive learning can enhance the ability to learn the structural node distribution and thus improve the classification accuracy, particularly when there are multi-classes.

The minor under-performance of NC-HGAT on the MR dataset may be due to the fact that it captures more background information or stop-words, which are unrelated to a specific class, thus diminishing the result. Another reason suggested by [31], who also found the under-performance of TextGCN model on MR dataset, is that edges in the constructed text graphs of MR are fewer than other datasets, as the documents are very short and thus limit the embedding learning ability.

TABLE II
MODEL ACCURACY ON DATASETS

Dataset	AGNews	MR	Ohsumed	Twitter
SVM+TFIDF	59.45%	54.29%	39.02%	53.69%
SVM+LDA	65.16%	54.40%	38.61%	54.34%
Bert	69.45%	53.48%	21.76%	52.00%
Text-GCN	67.61%	59.12%	41.56%	60.15%
HAN	62.64%	57.11%	36.97%	53.75%
HGAT	72.10%	**62.75%**	42.68%	63.21%
NC-HGAT	**73.15%**	62.46%	**43.27%**	**63.76%**

TABLE III
MODEL F1-SCORE ON DATASETS

Dataset	AGNews	MR	Ohsumed	Twitter
SVM+TFIDF	59.79%	48.13%	24.78%	52.45%
SVM+LDA	64.79%	48.39%	25.03%	53.97%
Bert	69.31%	46.99%	4.81%	43.34%
Text-GCN	67.12%	58.98%	27.43%	59.82%
HAN	61.23%	56.46%	26.88%	53.09%
HGAT	71.61%	**62.36%**	24.82%	62.48%
NC-HGAT	**72.06%**	62.14%	**27.98%**	**62.94%**

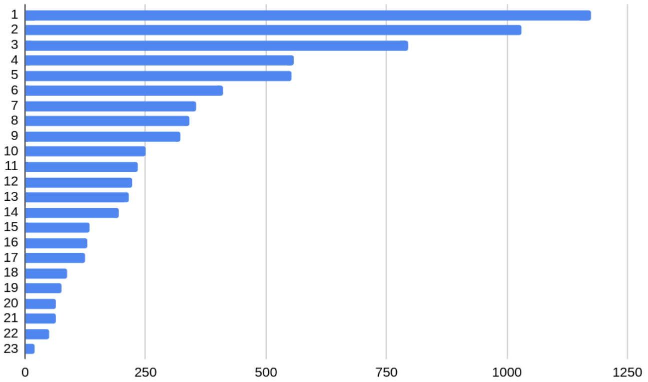

Fig. 3. Label distribution of the Ohsumed Dataset. Y-axis denotes the label of each document and X-axis represents the number of documents classified with that label. It is clear that the Ohsumed dataset is not uniform distributed

D. Impact of Layer Numbers of MLP

To investigate the impact of the MLP layer number deployed in section III-C, we evaluate our NC-HGAT model with 1-7 layers, inspired by [35], on the Twitter (very short sentence with less tokens seen in the Table I) and AGnews (relatively long sentence with more tokens) datasets as examples . As shown in Tables IV, V, the model with two layers performs better on the AGnews dataset; for the Twitter dataset, six layers perform the best. As for the AGNews dataset, the vanishing gradient and over-processed information will lead to an unstable model if the number of layers is excessive. The node representations may also become indistinguishable, known as the oversmoothing problem [36]. For the Twitter dataset, however, distant words may still be able to classify the document, and six layers can capture sufficient structural information.

TABLE IV
MODEL PERFORMANCE WITH DIFFERENT LAYERS ON THE TWITTER DATASET

Number of Layers	Accuracy	F1
1	63.04%	62.99%
2	61.86%	61.22%
3	61.05%	60.93%
4	63.66%	62.5%
5	63.28%	62.63%
6	**63.76%**	**62.9%**
7	62.79%	62.28%

TABLE V
MODEL PERFORMANCE WITH DIFFERENT LAYERS ON THE AGNEWS DATASET

Number of Layers	Accuracy	F1
1	73.00%	71.72%
2	**73.15%**	**72.06%**
3	72.50%	71.81%
4	72.85%	71.61%
5	72.50%	71.03%
6	72.60%	71.16%
7	72.3%	71.45%

V. CONCLUSION AND FUTURE WORK

In this paper, we propose, for the first time, to use contrastive learning to capture the topological information with HGAT on short text classification tasks. Extensive experiments on different datasets with different numbers of target classes illustrate that neighbour contrastive learning effectively learns and integrates structural information among entities and thus enhances the robustness of the existing model, particularly when there are limited labelled data.

Several potential extensions to our work could be addressed in future studies. The first is how to augment both feature and topology levels to improve the performance of graph learning models. For example, the links among documents could be directed; thus, hypergraphs, which allow one edge to link to more than two vertices, can be applied to capture more complex group-wise relationships. Second, fusing the method's expressive ability to capture global information with a multiple language model could also be insightful. Third, extending the model with other training methods, such as adversarial training, may improve the robustness more with fewer labels. Finally, the model can be applied to long text classification with multiple classes.

REFERENCES

[1] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," *arXiv preprint arXiv:2008.00364*, 2020.

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[3] Z. Wang, X. Liu, P. Yang, S. Liu, and Z. Wang, "Cross-lingual text classification with heterogeneous graph neural network," *arXiv preprint arXiv:2105.11246*, 2021.

[4] R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam, "Hetegcn: heterogeneous graph convolutional networks for text classification," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 860–868.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[6] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4821–4830.

[7] Z. Sun, A. Harit, J. Yu, A. I. Cristea, and N. Al Moubayed, "A generative bayesian graph attention network for semi-supervised classification on scarce data," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7.

[8] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, and Q. Wang, "Interpretable and efficient heterogeneous graph convolutional network," *arXiv preprint arXiv:2005.13183*, 2020.

[9] Y. Liu, R. Guan, F. Giunchiglia, Y. Liang, and X. Feng, "Deep attention diffusion graph neural networks for text classification," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 8142–8152.

[10] Y. Hu, H. You, Z. Wang, Z. Wang, E. Zhou, and Y. Gao, "Graph-mlp: Node classification without message passing in graph," *arXiv preprint arXiv:2106.04051*, 2021.

[11] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1150–1160.

[12] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," *arXiv preprint arXiv:2011.01403*, 2020.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[14] Y. Chen, "Convolutional neural network for sentence classification," Master's thesis, University of Waterloo, 2015.

[15] P. Liu, X. Qiu, X. Chen, S. Wu, and X.-J. Huang, "Multi-timescale long short-term memory neural network for modelling sentences and documents," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2326–2335.

[16] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[17] T. Yang, L. Hu, C. Shi, H. Ji, X. Li, and L. Nie, "Hgat: Heterogeneous graph attention networks for semi-supervised short text classification," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 3, pp. 1–29, 2021.

[18] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," *arXiv preprint arXiv:2011.00387*, 2020.

[19] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2021.

[20] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[21] W. Falcon and K. Cho, "A framework for contrastive self-supervised learning and designing a new approach," *arXiv preprint arXiv:2009.00104*, 2020.

[22] H. Fang, S. Wang, M. Zhou, J. Ding, and P. Xie, "Cert: Contrastive self-supervised learning for language understanding," *arXiv preprint arXiv:2005.12766*, 2020.

[23] M. Kim, J. Tack, and S. J. Hwang, “Adversarial self-supervised contrastive learning,” *arXiv preprint arXiv:2006.07589*, 2020.

[24] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *arXiv preprint arXiv:2004.11362*, 2020.

[25] M. Lopez-Martin, A. Sanchez-Esguevillas, J. I. Arribas, and B. Carro, “Supervised contrastive learning over prototype-label embeddings for network intrusion detection,” *Information Fusion*, vol. 79, pp. 200–228, 2022.

[26] L. Pan, C.-W. Hang, A. Sil, S. Potdar, and M. Yu, “Improved text classification via contrastive adversarial training,” *arXiv preprint arXiv:2107.10137*, 2021.

[27] Z. Guo, Z. Liu, Z. Ling, S. Wang, L. Jin, and Y. Li, “Text classification by contrastive learning and cross-lingual data augmentation for alzheimer’s disease detection,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6161–6171.

[28] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 5812–5823, 2020.

[29] P. H. Le-Khac, G. Healy, and A. F. Smeaton, “Contrastive representation learning: A framework and review,” *IEEE Access*, 2020.

[30] B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” *arXiv preprint cs/0506075*, 2005.

[31] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 7370–7377.

[32] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” *Advances in neural information processing systems*, vol. 28, pp. 649–657, 2015.

[33] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[35] L. Melas-Kyriazi, “Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet,” *arXiv preprint arXiv:2105.02723*, 2021.

[36] C. Yang, R. Wang, S. Yao, S. Liu, and T. Abdelzaher, “Revisiting over-smoothing in deep gcns,” *arXiv preprint arXiv:2003.13663*, 2020.