Contrastive Graph Learning with Graph Convolutional Networks

G Nagendar and Ramachandrula Sitaram

247.ai, Bangalore, India
{Gattigorla.Nagendar,NagaVenkata.R}@247.ai

Abstract. We introduce a new approach for graph representation learning, which will improve the performance of graph-based methods for the problem of key information extraction (KIE) from document images. The existing methods either use a fixed graph representation or learn a graph representation for the given problem. However, the methods which learn graph representation do not consider the nodes label information. It may result in sub-optimal graph representation and also can lead to slow convergence. In this paper, we propose a novel contrastive learning framework for learning the graph representation by leveraging the information of node labels. We present a contrastive graph learning convolutional network (CGLCN), where the contrastive graph learning framework is used along with the graph convolutional network (GCN) in a unified network architecture. In addition to this, we also create a labeled data set of receipt images (Receipt dataset), where we do the annotations at the word level rather than at the sentence/group of words level. The Receipt dataset is well suited for evaluating the KIE models. The results on this dataset show the superiority of the proposed contrastive graph learning framework over other baseline methods.

Keywords: Contrastive Learning · Graph Neural Networks (GNN) · Key Information Extraction · Graph Convolutional Networks (GCN).

1 Introduction

Optical character recognition (OCR) is the process of identifying text characters from scanned documents, which includes text detection and recognition. The scanned documents can be historical documents, receipts, bills, and invoices. Unlike only identifying the text characters from the document images, the key information extraction (KIE) from document images recently gained a lot of attention [1, 2]. It involves associating a label to each of the recognized word. Key information (entities) extraction from document images can play an important role in many applications like converting document information into a structured format, efficient archiving, fast indexing, document analytics, and so on. For example, in invoices, information like invoice number, total amount, and date gives richer and meaningful information about the invoice. Recently, there are many deep learning-based KIE approaches [8, 9, 12, 11, 40] that have emerged and outperformed traditional rule-based [3] and template-based [4, 5] methods. These traditional methods [3–5] are not robust against real-world settings, like the document images that are captured using mobile devices.

The deep learning-based KIE methods can be roughly categorized into three types: sequence-based [8, 9], graph-based [11, 12], and grid-based [10]. Among these methods, the graph-based methods, in particular, graph convolutional networks (GCN) [6, 7] are popularly used for the KIE problem. These methods [11, 12] model each document image as a graph, where the text segments (words or group of words) are represented as nodes. GCNs capture the structural information within the node's neighborhood by following a neighborhood aggregation scheme. However, the performance of graph-based methods [11, 12] majorly rely on the graph structure representation of given data. The node's neighborhood aggregation is computed using the graph structure. Recently, few methods have been proposed to learn a graph structure for the given data [13–15]. Henaff *et al.* [13] proposed a graph learning framework using a fully connected network. In [14], Li *et al.* proposed another framework for graph learning using a distance metric. Jiang *et al.* [15] proposed a graph learning convolutional network (GLCN) framework, which integrates both graph learning and graph convolution in a unified network architecture. Along with graph convolution, it also learns graph structure. Among these methods, GLCN [15] significantly outperforms other graph learning frameworks [13, 14].

The current graph representation learning methods [13–15] use the distance between the node features for checking whether two nodes are nearer or farther in the graph learning. These methods do not use the label information of the nodes in learning the graph representation. However, since the node features obtained in the initial steps are not specific to the given problem, these methods [13–15] result in sub-optimal graph representation and also lead to slow convergence. This is mainly because of the fact that, in the initial steps, the nodes that belong to different entities/classes may be nearer in the initial feature space. It can be avoided by leveraging the label information of the nodes.

In this paper, we propose a novel contrastive learning [18–20] framework for learning the graph representation, which assist graph-based methods [11, 12] to improve performance. Compared to other graph learning techniques [13–15], where the label information is not used in graph construction, the proposed contrastive learning framework allows to leverage this label information effectively. The proposed graph learning framework improves the performance of GCN based key information extraction methods [12, 11, 15]. We use contrastive loss [18, 19] to learn a class discriminative node representations by projecting them into an embedding space. Since we are considering the label information, the proposed framework results in an embedding space where elements of the same class are more closely aligned compared to other graph learning methods [13–15]. The graph representation is learned in this embedding space. The main idea for the proposed contrastive learning framework is to learn a projection network by pulling the set of nodes belonging to the same class (entity) together in an embedding space while simultaneously pushing apart clusters of nodes from different classes by leveraging the label information. In the embedding space, the samples in the same class will get a higher edge weight, while the samples belonging to different classes will get a smaller edge weight. In this way, an optimal graph representation is learned using contrastive loss in the embedding space. We present both supervised and semi-supervised contrastive loss functions for learning the graph representation. In addition to this, we also present a contrastive graph learning convolutional network (CGLCN), where both contrastive graph learning and graph convolution are integrated into a unified network architecture. It can be trained in a single optimization manner. Also, different from other KIE models [12, 11], we use multi-modal feature fusion using block super diagonal tensor decomposition [31] for combining the textual, visual, and positional features. We also created a Receipt dataset of grocery bills, with annotations at the word level. The images are captured in different view points and contain some amount of perspective distortion.

The main contributions of this paper can be summarized as follows

- We propose a novel graph representation learning framework using contrastive learning. Both graph learning and graph convolution are integrated into a unified network architecture.
- We also present a supervised and semi-supervised contrastive loss functions for the graph learning.
- The proposed CGLCN is evaluated on a real-world dataset, which shows the superiority of our model over other baseline models.

2 Relatedwork

In recent years, with the advantage of deep learning methods, key information extraction (KIE) from document images have gained encouraging improvement. Before the deep learning methods, the early works mainly used rule-based [3] or template matching methods [4, 5]. In general, these methods tend to fail on unseen templates and might lead to poor performance on real application scenarios. In real applications, the captured document images contain non-frontal views and some amount of perspective distortion. In KIE, significant progress has been made with the development of deep learning techniques. Most of these approaches formulate it as a token classification problem. The sequence-based approaches serialize the document into a 1D text sequence, then use sequence tagging methods [42, 43]. The sequence-based methods like LayoutLM [8] and LAMBERT [9] model the layout structure of documents based on the pre-training process of a BERT-like model. These methods pre-train on a large-scale dataset, making them less sensitive to the serialization step. The graph-based methods [41, 11, 12], represent the document image as a graph with the text segments as nodes in the graph. Several representation techniques [13–15] are used for the graph representation. The KIE problem is characterised as a node classification problem [11, 12]. In SPADE [41], a dependency graph is constructed for the given document. In VRD [11] and PICK [12], graph convolutional networks (GCN) [6]

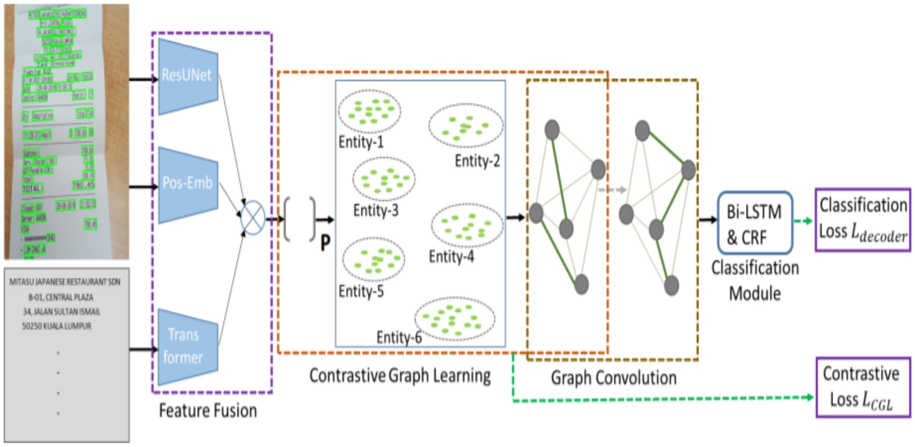

Fig. 1. The overall architecture of the proposed CGLCN model. The input for our model is an image containing text bounding boxes and their corresponding transcript. It contains graph learning, graph convolution, and classification modules. The features are combined using feature fusion techniques.

are used to get a richer representation for the nodes using message passing techniques. Our proposed contrastive graph learning framework can be used in the graph-based KIE methods [11, 12] for the improved graph representation.

Contrastive Learning Contrastive learning enables learning representations by contrasting positive against negative pairs [17]. It allows to learn class discriminative features. Contrastive learning is popularly used in self-supervised learning [23] and unsupervised visual representation learning [24, 25]. Recently, it has been successfully used for the self-supervised learning on graph structure data [21, 22]. Contrastive learning techniques are not previously explored for graph representation learning. In this work, we use contrastive learning techniques [19, 20, 18] for learning the graph representation, where contrastive loss is used to train the discriminative node embeddings.

3 Methodology

We present a contrastive graph learning framework for the graph-based key information extraction (KIE) methods [11, 12]. The overall architecture of the proposed contrastive graph learning convolutional network (CGLCN) is shown in Figure 1. In the first module, we extract the textual, visual, and positional features from the text bounding boxes. These features correspond to the node attributes in the graph representation. The extracted features are combined using the feature fusion techniques [31]. The node attributes are used for learning the graph representation using contrastive learning. We then use GCN for computing the node embeddings. The GCN contains one input layer, multiple hidden (convolutional) layers and the node embeddings are obtained from the final convolutional layer. Finally, in the classification module, we use Bi-LSTM [27] and CRF [28] for assigning the entity labels to the node attributes.

3.1 Graph Representation

We model the problem of key information extraction from document images as a graph node classification task. Here, we represent a document image as a undirected dynamic graph $G = \{V, E\}$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $E = [e_{ij}]$ is the edge (adjacent) matrix, which contains the edge weights between the nodes. $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{n \times m}$ is the feature matrix, where x_i is the m dimensional feature vector corresponding to the node v_i. The detected text regions (text bounding boxes) tr_i in the document image are represented as nodes in the graph, where, $tr_i = (x_1^i, y_1^i, x_2^i, y_2^i)$, (x_1^i, y_1^i) is the top left most coordinate and (x_2^i, y_2^i) is the bottom right most coordinate of the bounding box. We extract the following visual, textual, and positional features from the text bounding boxes, which are represented as the node attributes,

- *Visual Features*: The CNN features obtained from the cropped text bounding box region are used as the corresponding visual features of the node. We use ResUnet [33, 34] for extracting the visual features.
- *Textual Features*: The text features are extracted from the recognized text obtained from the cropped text bounding box. We use transformers [26] for extracting the text features.
- *positional Features*: For positional features, we use top left most coordinate of the text bounding box and its height. We use positional embedding of the bounding box location.

Now, each node in the graph is represented as $v_i = (vf_i, tf_i, pf_i)$, where vf_i, tf_i, and pf_i represents the visual, textual, and positional features respectively.

Multimodal Feature Fusion The node attributes play an important role in learning the graph representation. As discussed in the previous section, we are using textual, visual, and positional features for representing the nodes in a graph. Each of these features carries a different type of information. The standard way of feature fusion techniques like concatenation and weighted sum may not yield the best results. The fusion techniques should leverage the interactions between these feature representations. In this paper, we use the feature fusion technique proposed in BLOCK [31], which is a multimodal fusion framework based on the block-superdiagonal tensor decomposition [32]. It is primarily used for the fusion of textual and visual features for Visual Question Answering (VQA) and Visual Relationship Detection (VRD).

3.2 Contrastive Graph Learning

In this section, we present a framework for learning the graph representation using contrastive learning. The objective is to learn a projection network using contrastive loss [18–20] such that the cluster of nodes belonging to the same class (entity) will be pulled together in the projection space, while the cluster of nodes from different classes (entities) will be simultaneously pushed apart. This is done by effectively leveraging label information. The contrastive loss is applied to the outputs of the projection network. We normalize the outputs of the projection network to lie on a unit hypersphere. This enables to use the inner product as the similarity measure between the nodes in the projection space and we use it as the corresponding edge weight in the edge matrix E. The edge matrix gives an optimal graph representation for the given nodes. Since the nodes which belong to the same entity are closer to each other in the projection space, their corresponding edge weight (inner product) will be higher compared to the nodes which belong to different entities (clusters).

Projection Network P, We use a projection network $P : \mathbb{R}^m \to \mathbb{R}^l$, where m is the dimension of the node attributes, for projecting the node attributes into a projection space. l is the dimension of projection space and $l < m$. We use a multi-layer perceptron [16] with a single hidden layer as the projection network, and GeLU [35] is used as the activation function. The projection network is trained using a contrastive loss function.

The main advantage of the proposed approach is that we can use our framework for learning graph representation under supervised and semi-supervised settings. We now present the supervised and semi-supervised contrastive loss functions for training the projection network. The semi-supervised contrastive loss function enables to use both labeled and unlabeled nodes for training the projection network.

Supervised Contrastive Loss Function Let $V_P = \{P(x_1), P(x_2), \ldots, P(x_n)\}$ be the projection of node attributes in V for the projection network P. For supervised contrastive learning [18], the contrastive loss function is given as,

$$L_{sup} = \sum_{x \in V_P} \frac{-1}{|\pi(x)|} \sum_{u \in \pi(x)} log \frac{exp(\langle x, u \rangle / \tau)}{\sum_{v \in V_P \setminus \{x\}} exp(\langle x, v \rangle / \tau)} \tag{1}$$

where, $\pi(x) = \{z \in V; y_z = y_x\}$, y_x is the class label for the node x, $|\pi(x)|$ is the cardinality of $\pi(x)$ and $\tau \in \mathbb{R}^+$ is the temperature parameter. $\langle ., . \rangle$ denotes the inner product between node attributes to measure their similarity in the projection space.

Semi-Supervised Contrastive Loss Function In semi-supervised contrastive learning [19, 20], in addition to labeled document images, we also use unlabeled document images for learning the graph representation. Here, we use unlabeled data only for learning the graph representation. Using semi-supervised contrastive learning, we are able to achieve similar performance as supervised contrastive learning (Table 5) with a fewer number of labeled documents. In the first step, we cluster the nodes (text bounding boxes) from the unlabeled document images using a clustering algorithm. Since the densities of actual clusters vary a lot, the clusters obtained from this step may not be reliable clusters. There is a possibility of outliers in these clusters, which means the class labels for these nodes are different from the cluster representative label. These outliers are treated as unclustered nodes, and we remove these outlier nodes from their cluster by leveraging their context in the graph. We introduce a reliability criterion for recognizing the outliers in the clusters. In a cluster, a node is considered as an outlier if none of its neighbors in the graph are present in that cluster. We use the graph representation obtained from the previous iteration for computing the neighbors of a node.

For semi-supervised contrastive learning [19, 20], the contrastive loss function is given as,

$$L_{semi-sup} = - \sum_{x \in V_P^l \cup V_P^{ul} \setminus V_P^{ol}} log \frac{exp(\langle x, c \rangle / \tau)}{\sum_{i=1}^{n_l} exp(\langle x, m_i \rangle / \tau) + \sum_{i=1}^{n_{ul}} exp(\langle x, m_i^{ul} \rangle / \tau)} \tag{2}$$

where, V_P^l and V_P^{ul} are the projection of node attributes for the projection network P from the labeled and unlabeled documents, respectively. V_P^{ol} is the projection of outlier node attributes, n_l is the number of classes (entities) in the labeled data and n_{ul} is the number of clusters obtained from the clustering algorithm over the unlabeled data. $\tau \in \mathbb{R}^+$ is the temperature parameter. m_i and m_i^{ul} are the i^{th} class/cluster mean of labeled and unlabeled nodes respectively. If $x \in V_P^l$ belongs to k^{th} class, then $c = m_i$ and if $x \in V_P^{ul}$ belongs to k^{th} cluster, then $c = m_i^{ul}$. The major difference between the supervised (Eq 1) and semi-supervised (Eq 2) contrastive loss functions is the addition of a new term in the denominator ($\sum_{i=1}^{n_{ul}} exp(\langle x, m_i^{ul} \rangle / \tau)$) for the semi-supervised formulation, which corresponds to the unlabeled data. This term acts as a regularizer in the semi-supervised formulation.

4 Graph Convolution

The graph representation learned using contrastive learning (Section 3.2) is used along with the graph convolutional network (GCN) [6] for computing the node embeddings. The node embeddings obtained from the GCN are the problem specific node attributes. These node attributes are fed into the classification (decoder) module for classifying the nodes into one of the information categories. We use Bi-LSTM [27] and CRF [28] as the decoder and, negative log-likelihood as the loss function in the decoder module.

4.1 Loss

The overall loss in CGLCN constitutes the loss from the graph learning module (Eq 1 or Eq 2) and the loss from the classification (decoder) module. We take the weighted combination of these 2 losses as the final loss, which is given as,

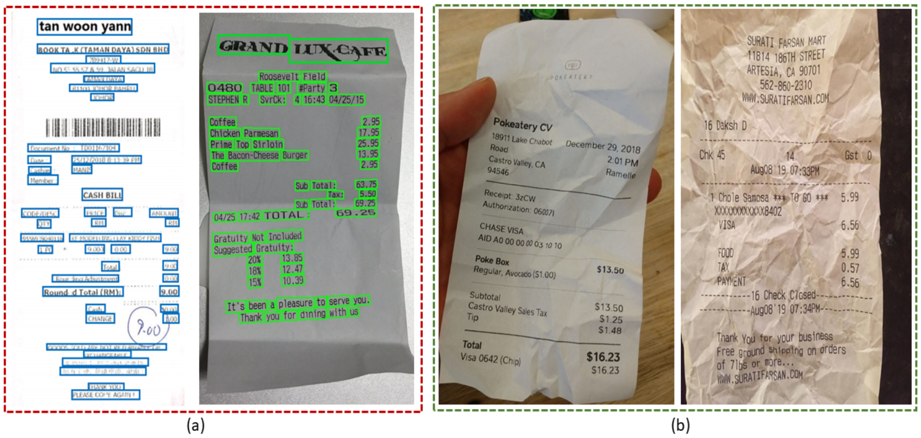

Fig. 2. (a) Annotations from SROIE (left) and Receipt datasets (right). (b) Few sample images from the Receipt dataset.

$$L_{CGLCN} = L_{CGL} + \lambda L_{decoder} \tag{3}$$

where, L_{CGL} is the loss from the contrastive graph learning module, which can either supervised or semi-supervised contrastive loss function. $L_{decoder}$ is the loss from the final classification module. λ is the trade off parameter.

5 Experiments

In this section, we validate the performance of proposed contrastive graph representation learning framework for the key information extraction task.

5.1 Datasets

To evaluate various components of the proposed contrastive learning framework, we test it on the following datasets.

SROIE Dataset [1], This dataset is part of the 2019 ICDAR-SROIE [1] competition. It contains the scanned receipt images, which do not have any perspective distortions. The training set contains 626 images, and the test set contains 347 images. All the receipts are labeled with 4 entities (categories), company, address, date, and total. It mainly contains English characters and digits. This dataset provides the text bounding boxes and their corresponding transcript. Some of the text bounding boxes are at a word level, and some of them contain multiple words. An example receipt image and the overlaid text bounding boxes are shown in Figure 2(a) (left).

Receipt Dataset, The receipt dataset is our in-house dataset, which contains 1612 images. These images are downloaded from the internet using different keywords. These are captured from different views and contain some amount of perspective distortion. We randomly selected 1300 images for training and 312 images for testing. The templates for test images differ from the training images. For the SROIE dataset [1], the text bounding boxes are not at the word level (Figure 2(a) (left)), and the key information extraction from those types of text bounding boxes is easier compared to the word level text bounding boxes. Also, producing those types of bounding boxes in real world setting is not quite easy. The document images captured in real application setting contains different types of layouts, perspective distortions, and capture from different views. This limits the applicability of SROIE dataset for evaluating the key information extraction methods.

For the Receipt dataset, we do the annotation at the word level. A word level bounding box is drawn for all the text in each document image and annotated their corresponding text. We then label each bounding box to one of 12 key information categories, including company name, address, date, item name, item price, total, subtotal, quantity, phone number, time, tax and individual item price. The annotated text bounding boxes for the Receipt dataset are given in Figure 2(a) (right). Few sample images containing non-frontal views and folds are shown in Figure 2(b). Since our dataset has word level annotations and is captured from different views containing perspective distortions, it is well suited for evaluating the performance of key information extraction methods. We use the same evaluation settings described in [1].

CORD Dataset [39], It includes images of Indonesian receipts and consists of 30 fields under 4 categories. We use the train/dev/test split of 800/100/100 respectively as proposed in [39].

5.2 Implementation Details

The proposed framework is implemented in PyTorch and trained on a NVIDIA GPU with 12 GB memory. Our model is trained from scratch using Adam [38] optimizer. We use a batch size of 4 during training. The learning rate is set to 10^{-4}. The encoder module of transformers [26] is used for text embeddings. The dimensionality for visual, textual and positional features are set to 256. We set the number of convolutional layers in CGLCN to 2 and train CGLCN for a maximum of 60 epochs. The averaged F1 score is used as the evaluative metric.

5.3 Baseline Methods

To verify the performance of the proposed contrastive graph learning framework for the key information extraction (KIE), we compare CGLCN with the graph-based methods GLCN [15], VRD [11] and PICK [12]. All these graph-based methods use graph convolutional networks [6] for computing the node embeddings. In these methods, the text bounding boxes in the document image are modeled as nodes in a graph. In both VRD [11] and PICK [12], nodes are represented using textual and visual features. These methods are summarized as follows,

- VRD [11]. It uses a fully connected graph for the graph representation. The node embeddings obtained from the GCN are fed into a standard BILSTM as the decoder for information extraction.
- GLCN [15]. Graph representation is learnt from the given data. We use textual, visual, and positional features as node attributes. It use MLP as the decoder.
- PICK [12]. The graph representation used in this work is inspired by the GLCN [15]. The node embeddings obtained from the GCN are fed into the Bi-LSTM and CRF layer for information extraction.

5.4 Comparison with Baseline Methods

We first compare our proposed CGLCN with the baseline methods (Section 5.3) on SROIE dataset in Table 1. To evaluate the superiority of the proposed contrastive learning for graph representation, we replace the graph representation module in GLCN [15], VRD [11] and PICK [12] with the proposed contrastive graph learning module and denote it as cont-GLCN, cont-VRD and cont-PICK respectively. The results are given in Table 2. We report the accuracy in terms of F1 score. As we can see, the contrastive counterpart of baseline methods perform reasonably well compared to their actual method. Also, we report an improvement of 1.8%, 2.4% and 1.6% in terms of F1 score over the baseline methods GLCN, VRD and PICK respectively. We also report the accuracy for our CGLCN model, where we use textual, visual, and positional features as compared to other baseline models. Also, different from these models, we use multimodal feature fusion using block super diagonal tensor decomposition [31] for combining the textual, visual, and positional features. The CGLCN model outperforms all other baseline models.

Method	F1 Score
GLCN [15]	90.8
cont-GLCN	92.4
VRD [11]	85.6
cont-VRD	87.8
PICK [12]	93.7
cont-PICK	95.1
CGLCN (proposed)	**96.5**

Table 1. Comparison with the baseline methods on SROIE dataset in terms of F1 score.

To further evaluate the proposed CGLCN in the presence of perspective distortions and viewpoint changes, we compare with the baseline methods on the Receipt dataset in Table 2. For all the methods, we also report the category wise accuracy and it is given in terms of the F1 score. Similar to the SROIE dataset, we compare the baseline methods with their corresponding contrastive version (cont-GLCN, cont-VRD, cont-PICK). From the results, we can observe that the proposed contrastive graph learning framework significantly improves the performance of all the baseline methods. We report the maximum improvement of 2.5% in the F1 score for the VRD [11], where a fully connected graph is used for graph representation. We report an improvement of 1.8% and 1.7% over the GLCN and PICK models. As we notice from Table 2, CGLCN outperforms all other baseline methods. We also observe the improvement in the performance over all the 12 information categories. Note that the images in the Receipt dataset are not scanned images, and these are captured from different viewpoints and

Method	Comp Nam	Addr	Date	Item Nam	Item Pr	Total	SubTotal	Quant	Ph-Num	Time	Tax	Ind Itm-Pr	Avg
GLCN [15]	76.2	82.0	85.1	87.4	84.3	72.6	69.5	86.4	82.4	86.3	68.6	71.5	79.3
cont-GLCN	78.2	83.5	86.7	88.6	85.7	74.6	73.1	88.2	84.3	88.2	69.8	73.0	81.1
VRD [11]	71.2	75.1	83.6	80.9	85.2	75.2	75.7	84.7	82.7	87.5	66.1	65.3	77.6
cont-VRD	73.6	77.5	85.4	83.9	87.1	77.8	77.3	87.4	84.2	90.2	68.8	69.1	80.1
PICK [12]	75.6	82.7	85.7	84.9	86.1	77.8	76.4	85.9	83.1	91.3	69.4	70.4	80.8
cont-PICK	77.9	85.1	87.5	87.2	88.3	79.5	78.0	87.3	83.7	92.8	71.0	72.4	82.5
CGLCN (proposed)	**79.4**	**85.9**	**88.1**	**90.6**	**88.6**	**81.4**	**79.4**	**90.7**	**84.8**	**95.6**	**71.6**	**73.8**	**84.1**

Table 2. Comparison with the baseline methods on Receipt dataset in terms of F1 score. Here, we compare the accuracy over all the 12 information categories (entities), which includes, company name (Comp Nam), address (Addr), date, item name, item price (Item Pr), total, sub total, quantity (Quant), phone number (Ph-Num), time, tax and individual item price (Ind Itm-Pr).

contain some amount of perspective distortion. This shows the robustness of our model, and is well suited for real-world applications.

We compare CGLCN with the state-of-the-art methods on CORD dataset in Table 3. The proposed CGLCN is comparable with LayoutLMv2 [8], TILT [40], DocFormer [44] and performing well compared to LAMBERT [9]. Note that, the CGLCN is taking fewer number of parameters compared to LayoutLMv2, TILT and DocFormer for achieving comparable performance. Also, CGLCN do not involve any pre-training.

Method	#param(M)	F1 Score
LAMBERT [9]	125	94.41
LayoutLMv2 [8]	426	96.01
TILT [40]	780	96.33
DocFormer [44]	536	**96.99**
CGLCN	308	95.83

Table 3. Comparison with the state-of-the-art methods on CORD dataset.

Method	GT	T-Det	T-Recg	End-to-End
GLCN [15]	79.3	78.4	74.8	73.2
VRD [11]	77.6	76.9	73.0	71.7
PICK [12]	80.8	80.0	76.5	75.1
CGLCN	**84.1**	**83.2**	**81.0**	**79.7**

Table 4. Comparison with the baseline methods, where text bounding boxes and the corresponding text is obtained from the text detection and recognition models.

In real applications, the text bounding boxes and their corresponding text is obtained from the text detection and recognition models. In general, these models may induce some detection and recognition errors. To further evaluate the CGLCN in the presence of these errors, for the test images, we take the text bounding boxes and their corresponding text obtained from the detection [36] and recognition [37] models. The results are given in Table 4. In the Table, the results under GT are obtained using ground truth text bounding boxes and their corresponding text. In T-Det, only the text bounding boxes are obtained from the detection model [36]. In T-Recg, the text in the GT bounding boxes are obtained from the recognition model [37]. In End-to-End setting, both the text bounding boxes and the corresponding text is taken from the text detection and recognition model. The proposed CGLCN outperforms all other baseline methods even if the text bounding boxes and the corresponding text is obtained from the detection and recognition models. For the T-Det, we observe a minimal drop in the performance for all the methods. This is mainly due to the word level annotations for the Receipt dataset, which suggests that it is well suited for evaluating the KIE models. For the T-Recg, we observe 3.1% performance drop in terms of the F1 score. This is mostly due to the recognition error in the numerical entities like date, item price, tax, subtotal, and total, where some digits are misrecognized as characters. Under End-to-End setting, we notice a smaller drop in the performance for CGLCN compared to all the baseline methods. It is more effective in handling text spotting errors than other baseline methods.

5.5 Supervised vs Semi-supervised Contrastive Graph Learning

This section evaluates the performance of proposed semi-supervised and supervised contrastive graph learning techniques. The results are given in Table 5. We show the performance of both semi-supervised and supervised models over a varying number of training images. For the semi-supervised model, the images that are not part of the training set are used as the unlabeled images. We do not use the node (text bounding boxes) labels in those images for learning the graph representation. As we can see, the semi-supervised model performs better compared to supervised model. As we increase the number of training images, the semi-supervised model outperform the supervised model. For the semi-supervised model, using 1300 samples, it is able to achieve the performance of the supervised model on 1500 training samples. The semi-supervised framework is well suited for the problems where we have unlimited unlabeled data and limited labeled data.

# Training Images	700	1000	1300	1500
Semi-Supervised	79.9	82.3	84.0	-
Supervised	79.8	81.6	83.0	84.1

Table 5. Performance of semi-supervised and supervised contrastive graph learning over varying number of training images in terms of F1 score.

5.6 Ablation Studies

To evaluate the various components of proposed CGLCN, we conduct a few ablation studies on the Receipt dataset.

The effect of textual, visual and positional features on CGLCN over Receipt dataset are demonstrated in Table 6. The results show that the textual features play an important role compared to visual and positional features. We notice a 6.7% drop in the F1 score without textual features. The textual features capture the main context of the node compared to visual and positional features. The positional features are contributing minimally to the CGLCN compared to textual and visual features. The positional features assist the graph

Method	F1 Score
without textual features	77.4
without visual features	82.0
without positional features	83.3
without graph learning (GL)	74.3
without adding GL features	83.2
without feature fusion	82.3
CGLCN	**84.1**

Table 6. Effect of textual, visual, positional and graph learning modules of CGLCN on Receipt dataset.

learning module in learning the graph representation. We also show the effect of the graph learning module in Table 6 under 'without graph learning (GL). As we can see, the graph learning module has an essential role in our proposed CGLCN. Compared to all other modules, it results in a significant drop in the F1 score (9.8%). The graph learning module helps significantly in capturing the structural information within nodes neighborhoods. In addition to this, we also show the importance of the features obtained from the graph learning (GL) module for the GCN in Table 6, under 'without adding GL features'. Here, we only use the features obtained from the feature fusion for computing the final node embeddings in GCN and we notice a 0.9% drop in the F1 score. In another ablation study, we examine the significance of feature fusion for combining textual, visual, and positional features. The results are shown under 'without feature fusion' in Table 6. Here, instead of the feature fusion discussed in Section 3.1, we simply concatenate the textual, visual, and positional features. We notice a drop in the F1 score from 84.1% to 82.3% without feature fusion. This suggests the importance of feature fusion techniques for key information extraction.

We also analyze the impact of the number of layers in GCN. We obtain the F1 score of 82.4%, 84.1%, and 83.6% for the number of layers 1, 2, and 3, respectively. We achieve the best performance for 2 layers. It tends to overfit if we set the number of layers to more than 2. In GCN, the probability of overfitting will tend to increase with the increase in the number of layers. The number of layers in GCN is task-specific.

6 Conclusion and Future Work

In this paper, we proposed a graph representation learning framework using contrastive learning. We also present contrastive graph learning convolutional network (CGLCN), where the contrastive graph learning framework and graph convolutional network (GCN) are integrated into an unified network architecture. The proposed contrastive graph learning framework can be used in any graph-based learning problem. In this paper, we validated for the problem of key information extraction. The experimental results on our Receipt dataset demonstrate the effectiveness of contrastive graph learning over other graph representation learning frameworks. In the future, we explore the contrastive graph learning framework for other tasks, such as graph clustering, pose estimation, image classification and image cosegmentation.

References

1. Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shjian Lu, C.V. Jawahar: ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In ICDAR (2019).
2. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer: Neural architectures for named entity recognition. In NAACL (2016).
3. Esser, D. Schuster, D. Muthmann, K, Berger, M, Schill, A: Automatic indexing of scanned documents: a layout-based approach. In DRR (2012).

4. F. Cesarini, E. Francesconi, M. Gori, G. Soda: Analysis and understanding of multi-class invoices. In DAS (2003).
5. A. Simon, J.-C. Pret, A. P. Johnson: A fast algorithm for bottom-up document layout analysis. In PAMI (1997).
6. Thomas N Kipf, Max Welling: Semi-supervised classification with graph convolutional networks. In ICLR (2017).
7. Veličković, Petar, Cucurull, Guillem, Casanova, Arantxa, Romero, Adriana, Liò, Pietro, Bengio, Yoshua: Graph Attention Networks. In ICLR (2017).
8. Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang, L. Zhou: LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. In arXiv (2020).
9. L. Garncarek, R. Powalski, T. Stanisławek, B. Topolski, P. Halama, M. Turski, F Graliński: LAMBERT: Layout-Aware (Language) Modeling for information extraction. In ICDAR (2021).
10. Weihong Lin, Qifang Gao, Lei Sun, Zhuoyao Zhong, Kai Hu, Qin Ren, Qiang Huo: ViBERTgrid: A Jointly Trained Multi-Modal 2D Document Representation for Key Information Extraction from Documents. In ICDAR (2021).
11. Xiaojing Liu, Feiyu Gao, Qiong Zhang, Huasha Zhao : Graph Convolution for Multimodal Information Extraction from Visually Rich Documents. In NAACL (2019).
12. Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, Rong Xiao : PICK: Processing Key Information Extraction from Documents Using Improved Graph Learning-Convolutional Networks. In ICPR (2020).
13. Mikael Henaff, Joan Bruna, and Yann LeCun: Deep convolutional networks on graph-structured data. In arXiv (2015).
14. Feiyun Zhu Junzhou Huang Ruoyu Li, Sheng Wang : Adaptive graph convolutional neural networks. In AAAI (2018).
15. Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang and Bin Luo: Semi-supervised Learning with Graph Learning-Convolutional Networks. In CVPR (2019).
16. Trevor Hastie, Robert Tibshirani, and Jerome Friedman : The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA (2001).
17. Hadsell, R., Chopra, S., and LeCun, Y : Dimensionality reduction by learning an invariant mapping. In CVPR (2006).
18. Prannay Khosla, Piotr Teterwak, ChenWang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu : Supervised Contrastive Learning. In NIPS (2020).
19. Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, Hongsheng Li: Self-paced Contrastive Learning with Hybrid Memory for Domain Adaptive Object Re-ID. In NIPS (2020).
20. Yuhang Zhang, Xiaopeng Zhang, Robert.C.Qiu, Jie Li, Haohang Xu, Qi Tian: Semi-supervised Contrastive Learning with Similarity Co-calibration. In CoRR abs/2105.07387 (2021).
21. Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, ZhangyangWang, Yang Shen: Graph Contrastive Learning with Augmentations. In NIPS (2020).
22. Yuning You, Tianlong Chen, Yang Shen, Zhangyang Wang: Graph Contrastive Learning Automated. In ICML (2021).
23. Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton: A simple framework for contrastive learning of visual representations. In arXiv (2020).
24. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R : Momentum contrast for unsupervised visual representation learning. In CVPR (2020).
25. Chen, T., Kornblith, S., Norouzi, M., Hinton : A simple framework for contrastive learning of visual representations. In ICML (2020).

26. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin : Attention is all you need. In NIPS (2017).
27. A. Graves and J. Schmidhuber : Framewise phoneme classification with bidirectional lstm networks. In IJCNN (2005).
28. J. Lafferty, A. McCallum, F. C. Pereira: Conditional random fields:Probabilistic models for segmenting and labeling sequence data. In ICML (2001).
29. Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou : LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In SIGKDD (2020).
30. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova : BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL (2019).
31. H. Ben-younes, R. Cadene, N. Thome, M. Cord : BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection. In AAAI (2019).
32. De Lathauwer L: Decompositions of a higher-order tensor in block terms part ii: Definitions and uniqueness. In SIMAX (2008).
33. Zhang, Zhengxin, Liu, Qingjie : Road Extraction by Deep Residual U-Net. In GRSL (2017).
34. Diakogiannis, F. I., Waldner, F., Caccetta, P., Wu, C : ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. In ISPRS (2020).
35. Hendrycks, Dan Gimpel, Kevin : Gaussian Error Linear Units (GELUs). In ArXiv (2016).
36. Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, Xiang Bai : Real-time Scene Text Detection with Differentiable Binarization. In AAAI (2020).
37. Baoguang Shi, Xiang Bai, Cong Yao: An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. In PAMI (2017).
38. D. P. Kingma, J. Ba: Adam: A method for stochastic optimization. In arXiv (2014).
39. Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., Lee, H.: CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In Document Intelligence Workshop at NeurIPS (2019).
40. Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, Gabriela Pałka: Going full-tilt boogie on document understanding with textimage-layout transformer. In arXiv (2021).
41. W. Hwang, J. Yim, S. Park, S. Yang, M. Seo: Spatial Dependency Parsing for SemiStructured Document Information Extraction. In ACL-IJCNLP (2021).
42. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL (2019).
43. X. Ma, E. Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNsCRF. In ACL, (2016).
44. Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, R. Manmatha: DocFormer: End-to-End Transformer for Document Understanding. In ICCV (2021).