Multilevel Hypernode Graphs for Effective and Efficient Entity Linking *

David Montero and **Javier Martinez** and **J. Javier Yebes**
NielsenIQ
{david.montero,javier.martinezcebrian,javier.yebes}@nielseniq.com

Abstract

Information extraction on documents still remains a challenge, especially when dealing with unstructured documents with complex and variable layouts. Graph Neural Networks seem to be a promising approach to overcome these difficulties due to their flexible and sparse nature, but they have not been exploited yet. In this work, we present a multi-level graph-based model that performs entity building and linking on unstructured documents, purely based on GNNs, and extremely light (0.3 million parameters). We also propose a novel strategy for an optimal propagation of the information between the graph levels based on hypernodes. The conducted experiments on public and private datasets demonstrate that our model is suitable for solving the tasks, and that the proposed propagation strategy is optimal and outperforms other approaches.

1 Introduction

Information extraction (IE) from documents has become a hot research topic over the last few years (Jaume et al., 2019; Wang et al., 2020; Carbonell et al., 2021; Dang et al., 2021). It is a challenging problem that requires combining Computer Vision (CV) and Natural Language Processing (NLP) models in order to locate and parse the information segments, understand the document layout, and extract semantic relations between the segments.

This problem becomes especially complex when dealing with unstructured documents, such as purchase receipts, where the layout of the documents can highly vary, making it hard for the models to learn how to extract semantic information. At this point, Graph Neural Networks (GNNs) seem to be a promising approach to overcome these difficulties and to solve the semantic information and relation extraction tasks, as they work over flexible graph-based representation capable of adapting to complex layouts, and they provide efficient and effective mechanisms for learning the relations between the segments (Carbonell et al., 2021; Davis et al., 2021; Hwang et al., 2021b; Baumgartner et al., 2021; Papagiannopoulou et al., 2021; Luo et al., 2020; Khalife and Vazirgiannis, 2019).

Nevertheless, semantic IE still remains a challenging task. In fact, due to its complexity, it is usually split into three subtasks:

- Entity Building (EB): refers to the task of connecting text segments together that are related semantically and are spatially close in the document, also known as word grouping.
- Entity Tagging (ET): classify each of the built entities attending to their semantic meaning, e.g., product description, store name, etc.
- Entity linking (EL): connect the semantic entities to form higher level semantic relations, e.g., a product description is connected to a quantity and a price.

Thus, we can distinguish between three levels of information containers:

- Text segment: lowest level information, usually given by an Optical Character Recognition (OCR) engine at word level.
- Entity: intermediate level generated by grouping the text segments during the EB task.
- Entity group: highest level container that groups entities resultant from the EL task.

For a solution based purely on GNNs this leaves two options. One is trying to solve all the tasks using a single graph at segment level (Hwang et al., 2021b). The second option is splitting the problem into two graphs: one graph based on segment nodes for performing EB, and another one composed of

[*] A patent has been applied for that covers the subject matter described in this article.

entity nodes for performing ET and EL tasks (Carbonell et al., 2021). We believe that the second one is more effective for the following reasons:

- The model can work on extracting node-level relations only, which reduces the complexity.
- The information learnt by the segments nodes during the message passing can be used to generate optimal features for the entity nodes.

Nevertheless, the multi-graph approach has more complexity, as it requires designing the way the output segment features and the entity features are related, and it has not yet been studied in depth. Thus, in this work we focus on optimizing this propagation of information between the two stages using a novel approach within the IE field based on hypernodes. These are the main contributions:

- A multi-level GNN-based model that performs EB and EL on unstructured documents. The model is purely based on GNNs, using as inputs for each segment the bounding box and the entity category, and it is extremely light (0.3 million parameters).
- A novel strategy for an optimal propagation of the information from the segment nodes to the entity nodes, where the latter are generated as hypernodes over the base graph and connected to their child segment nodes using relation edges. Then, the subgraph resulting from the relation edges (relation graph) is used to propagate the features with Graph Attention Layers (GATs) (Veličković et al., 2018).
- An ablation study on different feature propagation strategies, evaluating among others the one proposed in (Carbonell et al., 2021), and comparing them with the single graph approach (Hwang et al., 2021b).

The conducted experiments demonstrate the effectiveness of the proposed method over highly unstructured documents in terms accuracy, processing time, and resource consumption.

2 Related Work

The growing interest in IE is patent in the number of recent publications. Attending to the input data, most of the methods rely on the text and bounding boxes of an OCR engine for extracting the input features (Jaume et al., 2019; Carbonell et al., 2021; Hwang et al., 2021b; Prabhu et al., 2021; Zhang et al., 2021; Hong et al., 2022; Wang et al., 2022). Other approaches enrich these OCR predictions with image features (Wang et al., 2020; Dang et al., 2021; Xu et al., 2021; Tang et al., 2021). However, the results reported in public IE benchmarks like FUNSD (Jaume et al., 2019) or CORD (Park et al., 2019) suggest that the image features are not so relevant. Finally, there are also a few models that purely rely on image features (Hwang et al., 2021a; Kim et al., 2021). The model proposed in this work extracts features from the OCR bounding boxes, but does not use the text, as it gathers the necessary information from the entity category input.

Attending to the model architecture, most of the methods are based on Transformers (Vaswani et al., 2017) and Convolutional Neural Networks (CNNs) (Jaume et al., 2019; Wang et al., 2020; Dang et al., 2021; Xu et al., 2021; Hwang et al., 2021a; Li et al., 2021; Prabhu et al., 2021; Zhang et al., 2021; Kim et al., 2021; Villota et al., 2021; Hong et al., 2022; Gu et al., 2022; Wang et al., 2022). Nevertheless, GNNs are gaining importance thanks to their flexibility and capacity of adapting to complex layouts, along with their effective mechanisms for learning relationships between the nodes. In (Carbonell et al., 2021), the authors propose a two-stage GNN model. First, they generate a k-nearest neighbor (KNN) graph to solve EB using text and bounding box features. Then, the entity features are computed by aggregating the output features and processing them with a linear layer, and they are used to solve the ET and EL. In (Hwang et al., 2021b), the authors propose a single-stage GNN model: EB and ET are solved via rel-s edges where each seed entity-type node links to its entity parts in sequence (solving also ET as a consequence), EL links the entities via rel-g edges, finally all mentioned edges are decoded at once. Other GNN approaches solve only the ET and EL tasks, as they rely on the entity regions detected by the OCR engine (Tang et al., 2021; Wan et al., 2021; Zhang et al., 2022), or by a previous CNN model (Davis et al., 2021).

As it can be seen, there are few approaches based on GNN solving both EB and EL. We aim at contributing to this line of research following an approach based on a two-stage GNN model, related to the one presented in (Carbonell et al., 2021), but with important modifications in the feature extraction, edge sampling, feature propagation, GNN architecture, and postprocessing.

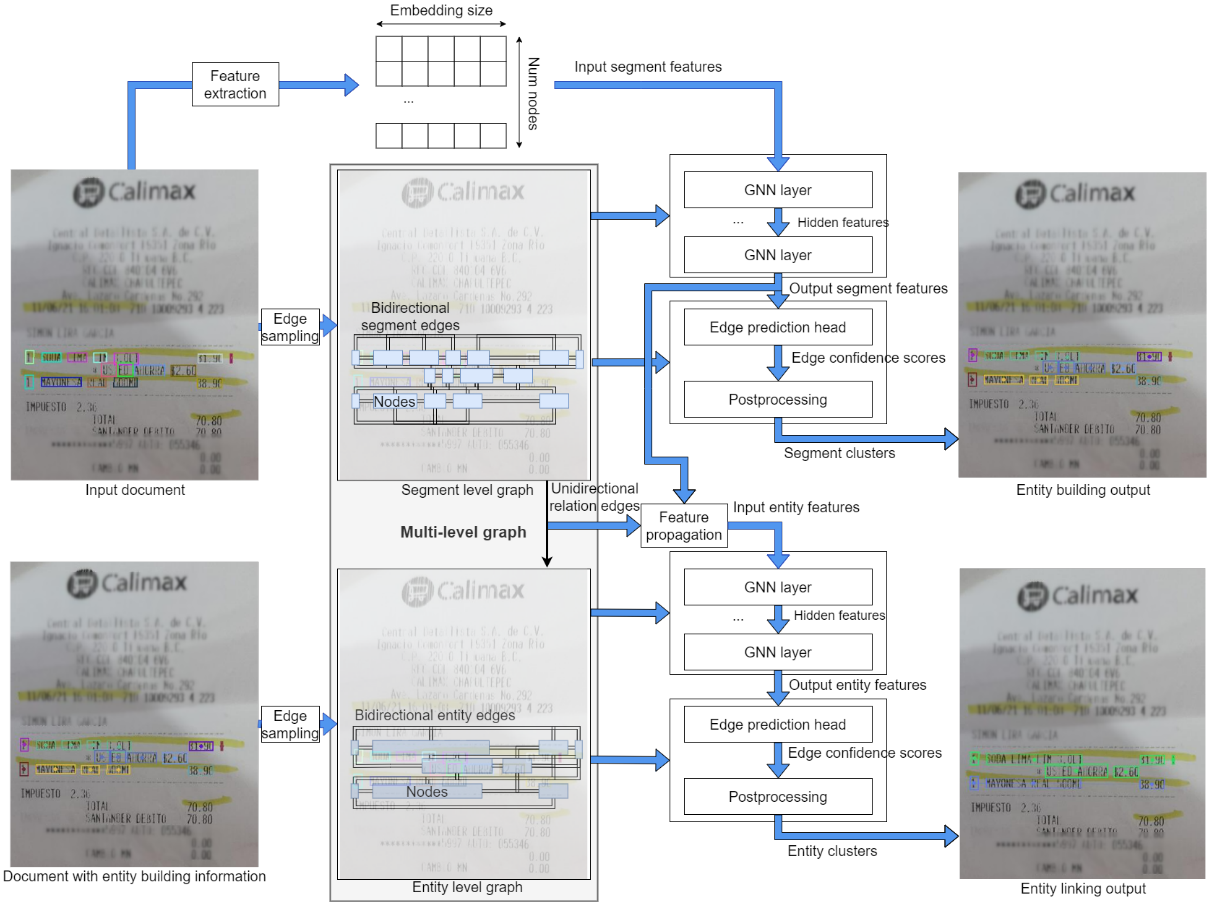

Figure 1: High level diagram of the proposed solution for the EB and EL tasks.

3 Methodology

We aim at solving the entity building (EB) and entity linking (EL) tasks for a given list of documents. Each document is composed of a list of semantic entities, that can be linked together to form entity groups. Each entity can also be divided into smaller text segments. Thus, given a list of text segments from an OCR engine, the goal is to group the text segments by their entity and then link together all the entities that belong to the same entity group. We propose to use GNNs as the best approach:

- Graph-based representations can adapt to complex layouts in unstructured documents.
- EB and EL can be modeled as link prediction tasks between pairs of segments, where GNNs have been demonstrated to be highly effective.
- The number of connections that need to be evaluated can be limited based on the coordinates, limiting the time and resource consumption. GNNs are suitable for this type of highly sparse data structure.

Figure 1 illustrates the proposed solution. From the incoming list of segments, the system performs the edge sampling and generates the base graph level. In parallel, the features for the nodes are extracted. The input features are passed through the segment GNN layers and used to generate the segment clusters (EB output). For each generated cluster, an entity hypernode is created and connected to their child segment nodes using relation edges. Then, feature propagation uses the subgraph of relation edges (relation graph). Finally, these entity features are processed in the same way as in the previous stage to generate the entity clusters (EL output).

3.1 Feature extraction

We consider the three sources of information available: the bounding box, the text string and the entity category. We discard the text, as we have empirically observed that all the necessary information is contained in the entity category. Also, we remove the impact of the OCR text errors.

We select the following features from the bounding box: left and right center coordinates, and the angle in radians $(\frac{-\pi}{2}, \frac{\pi}{2})$. Notice that using the left and right center we are losing the information related to the height of the bounding box. We do this on purpose, as we observed that the model tended to overfit using this feature. We normalize both centers using the width of the document, the most stable dimension, as the height can highly vary. For extracting the information from the entity category we use a one-hot encoder, and then a linear layer to adapt the features and map them to an embedding of length 8. Finally, the category embedding is concatenated with bounding box features to generate the node feature embedding (with 13 float values).

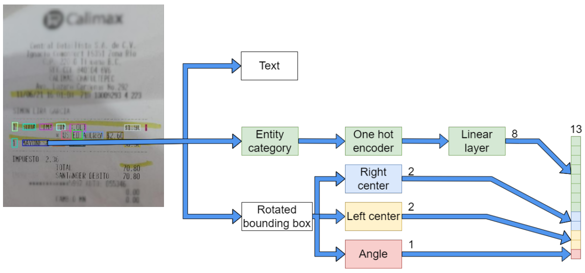

Figure 2: Feature extraction stage.

3.2 Edge sampling

The message passing involve the edges and also they are used by the edge prediction head to generate the final predictions. Hence, it is crucial to select an appropriate sampling function that covers all the possible true positives.

Moreover, we are dealing with highly unstructured documents and we cannot trust the usual sampling functions, such as k-nearest neighbor or beta-skeleton (Carbonell et al., 2021; Wan et al., 2021; Zhang et al., 2022), as they are prone to miss connections between segments that are far away from each other.

Thus, we developed a custom sampling function to ensure that all the segments in the same line are connected: an edge from segment A to segment B is created if the vertical distance between their centers (C) is less than the height (H) of segment A by a constant (K) (see Equation 1). In our experiments we set this constant to two, as we want to generate connections also between the segments of adjacent lines for the case of multi-line entities, and to consider the possible rotation of the document. This sampling function is also used to generate the edges for the entity level graph.

$$edge_{A-B} = |C_A^y - C_B^y| < H_A * K \qquad (1)$$

3.3 GNN

Selecting the most appropriate type of layer is another important step in the model design. Most of the GNN layer implementations require an additional scores vector for performing a weighted message passing, for deciding the contribution of each neighbor node. This implies adding more complexity to the design of the network for computing the weights.

In our case, the information needed for that computation is already embedded in the node features. Taking advantage of this, we select Graph Attention Layers (GAT) (Veličković et al., 2018) as the best suited. In the GAT layers, the weights for the message passing are computed directly inside the layer using the input node features. In addition, they have been widely used and demonstrated their efficiency in document understanding tasks (Carbonell et al., 2021; Zhang et al., 2022). In order to avoid 0-in-degree errors (disconnected nodes) while using the GAT layers, we add a self-loop for each node.

The proposed GNN architectures for the two graph levels are illustrated in Figure 3 and they both use GAT layers. All the layers are followed by SiLU activations (Elfwing et al., 2018) except for the last one. This activation seemed to work better than ReLU and other variants. We also add residual connections in all the layers to accelerate the convergence of the model.

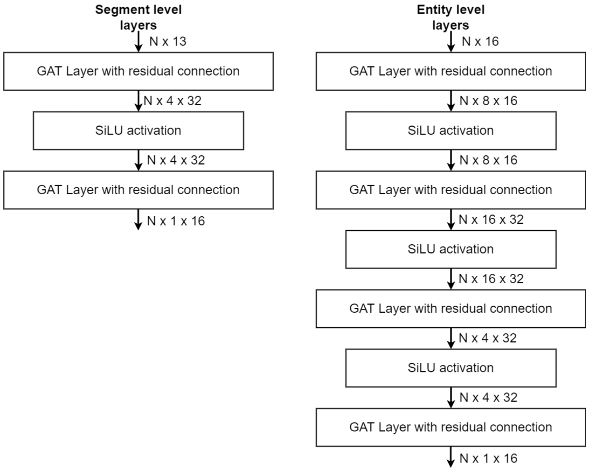

Figure 3: Proposed GNN architectures.

Another introduced enhancement is the use of a global document node, inspired by (Zhang et al., 2022). We use one global node per graph level, and we connect it bidirectionally to the rest of the level nodes. Its feature embedding is initially computed by averaging all the level node embeddings. It has a double function in the network: it provides context information to the nodes, and it acts as a regularization term for the GAT layer weights. These global nodes are only considered during the message passing.

3.4 Feature propagation

The feature propagation strategy is one of the critical parts of the model, as it defines the connection between the two stages and how the entity features are generated.

First, we analyze the strategy followed in (Carbonell et al., 2021), where the features of the nodes belonging to the same entity are added and processed by a linear layer. We believe that this strategy is not optimal for two reasons. First, as the number of nodes of an entity is variable, adding their features will lead to variable magnitude embeddings, which might impact on the stability of the model. This could be mitigated by using a mean aggregation. Second, they assume that all the segment nodes contribute equally to the entity. We believe that this is an erroneous assumption, as there might be key segments (maybe those which are bigger, or which have a strategic position) that should contribute more.

We propose a new approach where the entity nodes are built as hypernodes on top of the segment level graph and connected to their child segment nodes using unidirectional relation edges (from segments to entities). Then, the features propagation is conducted by GAT layers that operates on the subgraph of the relation edges (relation graph). The feature propagation model is composed of 2 GAT layers with a SiLU activation between them. In this case we do not use residual connections, as we want to maximize the information shared by the segment nodes. See below:

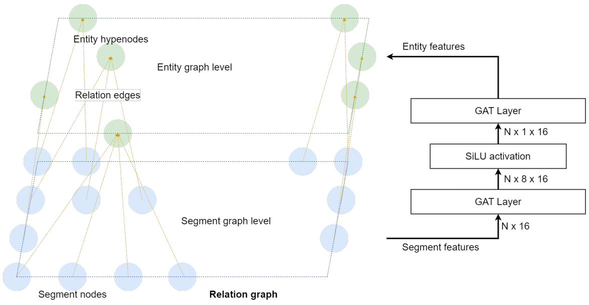

Figure 4: Feature propagation strategy.

3.5 Edge prediction heads

After each GNN level, the node features are used to solve the corresponding task (EB or EL). For each pair of connected segments, we extract the confidence that they belong to the same higher-level container. The strategy we follow is concatenating the output features of the pair of nodes and processing them with an MLP (see Figure 5). After the first layer, we apply another SiLU activation. Finally, we apply a sigmoid function to the output logits to obtain the confidence scores.

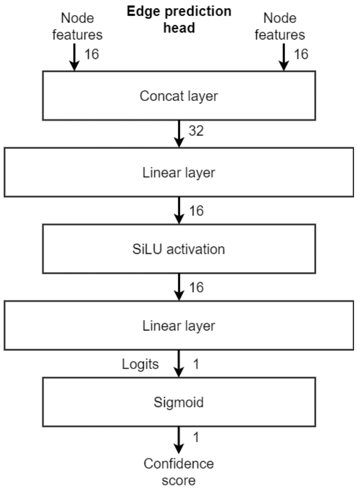

Figure 5: Diagram of the edge prediction heads.

3.6 Postprocessing

Once the confidence scores for a task are computed, we apply a postprocessing function to generate the final clusters. For edge prediction tasks, a commonly used function is Connected Components (CC) (Carbonell et al., 2021). However, due to its simplicity, it highly suffers from any link error, it usually struggles when dealing with complex data distributions, and it depends on a threshold parameter which might be biased to the dataset. For these reasons, we propose to use a different method based on Graph Clustering made of 2 blocks (Equation 2): 1) number of clusters estimator and 2) node grouping. The former, 1), is based on the eigenvalues of the normed graph Laplacian matrix computed from the adjacency matrix (A), by taking first differences (D1) of the sorted eigenvalues and getting the maximum gap + 1. The latter, 2), is based on recursively merging pair of clusters, using the number of clusters estimated (nc) and as the linkage criteria the average of the distances (1 minus the adjacency matrix), being a highly efficient method.

$$\begin{aligned}\lambda &= EigenValues(NormGraphLap(A)) \\ n_c &= argmax(D_1(sort(\lambda))) + 1 \\ c_i &= FeatAgglom(avg(1-A), n_c)\end{aligned} \quad (2)$$

The benefits are: no need to optimize any parameter avoiding concept drift impact, estimating the number of clusters dynamically for each new data distribution, no need of handcrafted heuristics, and efficient and accurate as the CC approach.

3.7 Training details

Only during the training stage, the entities are constructed using the ground truth (GT). This accelerates the convergency of the model, as it reduces the dependency of the EL task and the EB task. The model is trained for 100 epochs using a batch of 4 graphs on each iteration. The selected optimizer is Adam, with an initial learning rate of 0.001, with a reduction factor of 0.1 in epochs 70 and 90. We use binary cross entropy for computing the loss for the two tasks, and then we sum both losses. Finally, we finetune the model using the predicted entities instead of the GT, so the second part of the model adapts to the real data. The benefits of finetuning the models are demonstrated in the experiments section. The model is finetuned for 10 epochs, with an initial learning rate of 0.0002, being reduced to 0.00002 at epoch 7.

4 Experiments

4.1 Datasets

4.1.1 Private dataset

We have built an internal challenging dataset composed of 8729 purchase receipt images from 5 countries: Germany, Italy, France, Mexico, and Brazil. Receipts vary widely in height, density, and image quality. They may contain rotation and all kinds of wrinkles. Each receipt has annotated all the text segments related to purchased products. The available annotated information for each text segment is the rotated bounding box, the text, the entity category, and the product ID.

There are 9 types entity categories: $unit\_type$, $value$, $discount\_value$, $code$, $unit\_price$, tax, $quantity$, $discount\_description$, $description$.

The dataset also contains the receipt region annotation for each receipt, so we have preprocessed the dataset for all the models by cropping the images, filtering the segments that are outside the receipt, and shifting the coordinates of the remaining segments to the cropped pixel space. Finally, we split the dataset in training, validation and test sets using a ratio of 70/10/20.

In Figure 6 we present some examples of the dataset after cropping the receipt region. We also include in the images the GT information for the entity building (bounding boxes) and the entity linking (bounding boxes with the same colors and linked by lines). Note that this dataset is more challenging than other IE datasets, such as FUNSD (Jaume et al., 2019) or CORD (Park et al., 2019), as the number of entities can vary from several to hundreds, layouts are highly diverse, and the quality of the receipts and images has a bigger amount of noise.

4.1.2 CORD

Consolidated Receipt Dataset (CORD) (Park et al., 2019) is composed of 1000 Indonesian receipts which contain images and box/text annotations for OCR, and multi-level semantic labels for semantic parsing and relation extraction tasks. In the ground truth, each segment is associated with the $category$ field (our entity level) and the $group\_id$ field (our group level). It contains more entity categories (30), but with significantly fewer instances. It can be observed that the difficulty level is lower but it is the only public dataset we have found for benchmarking. In this dataset, the receipt region annotations are available only for a subset of receipts, so we are not considering them. The samples are split into 800 for train, 100 for dev(validation), and 100 for test.

4.2 Metrics

4.2.1 Group F1 Score

This metric is very restrictive and aims at evaluating the number of groups that are perfectly formed, highly penalizing the groups that are split or merged with others. We compare the predicted groups with the ones from the ground truth. For each predicted group in a document, we only consider it as a true positive (tp) if it matches exactly the ground truth group. Otherwise, it is considered a false positive (fp). Ground truth groups not found in predictions are considered as false negatives (fn).

4.2.2 ARI

The Adjusted Rand Index (ARI) (Halkidi et al., 2002), is more focused on analyzing the quality of the segment clusters rather than checking if they perfectly match the ground truth ones. First, the Rand Index (RI) computes a similarity measure between two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusters. Then, the raw RI score is "adjusted for chance" into the ARI score.

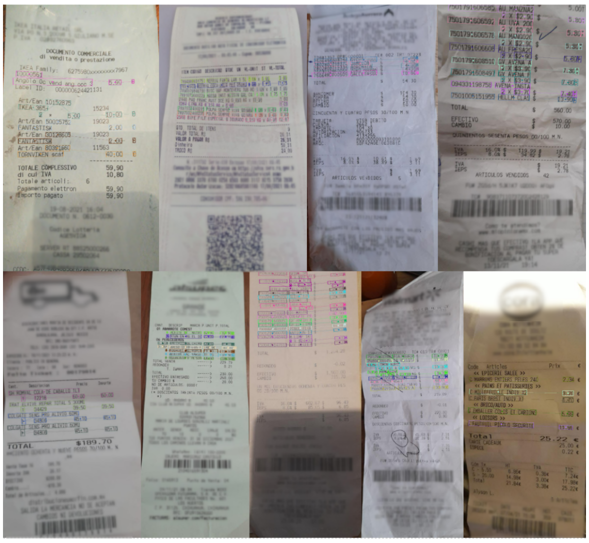

Figure 6: Examples of successful predictions from different countries and retailers. Each box is a predicted entity, and the ones with the same color (and connected by lines) belong to the same group.

4.3 Results

In this subsection, we present and discuss the experimental results with the aim of demonstrating the effectiveness of the proposed method and the contribution of our novel feature propagation. These are the considered approaches:

- Relation graph: described in Section 3.4.
- Without feature propagation: the features of the entities are generated from scratch, in the same way as the text segment features. The entity bounding box is computed using the minimum rotated rectangle and the entity category is computed using the mode.
- Sum aggregation + linear layer: the procedure followed in (Carbonell et al., 2021).

Besides, we include in the comparison the results of a single-stage version of the model, following the approach proposed in (Hwang et al., 2021b). The GNN architecture for this model is the same as for the entity GNN of the proposed model.

Model	EB		EL (E2E)	
	F1	ARI	F1	ARI
ours	0.974	0.966	0.925	0.960
w/o feat prop	0.9756	0.971	0.914	0.955
sum+linear	0.971	0.965	0.915	0.955
Single stage	0.979	0.973	0.913	0.950

Table 1: Results of the proposed model on the purchase receipt dataset and comparison against different feature propagation strategies. We present the results for EB and EL (using the entities predicted in EB).

For all the variants, the model is trained under the same conditions, following the training details specified in Section 3.7. The results of the experiments are gathered in Table 1. It can be observed that the proposed model is achieving impressive results for both tasks (0.974 F1 Score for EB and 0.9252 for EL) considering the challenges of the proposed dataset. Some examples of successful model predictions are shown in Figure 6.

Also the proposed strategy for the entity features generation outperforms the others in the end2end metrics by more than 1%. The strategy without feature propagation achieves slightly better results in EB (less than 0.2%), but we believe this is because in this case the two tasks are more independent from each other, and the model can focus on optimizing better the first task (but at the cost of sacrificing accuracy in the end2end). The same happens with the single stage strategy.

Additionally, we want to measure the impact of the finetuning stage described in Section 3.7, where, instead of using the GT information to construct the entities, we use the predictions from the EB task, and train the model in an end2end manner for 10 epochs. Thus, we compute the end2end metrics for all the model variants before and after the finetuning. The results, presented in Table 3, show that in all the cases both the F1 Score and the ARI metrics are improved. This improvement is less noticeable for our approach, as even if we are using GT information for constructing the entities, the two tasks are still strongly connected by an optimal feature propagation strategy.

Next, we conduct an experiment to test the proposed model under a public benchmark, using the CORD dataset. For this experiment we consider all the annotated segments, using the $category$ field as the entity annotation and the $group\_id$ field as the group annotation. Again, the model is trained following the procedure specified in Section 3.7.

Model	EB Link F1	EL Link F1	EL Group F1	ARI	Params
Rel graph (ours)	0.975	0.988	0.943	0.983	0.3M
Spade(Hwang et al., 2021b)	0.969	0.896	-	-	-
BROS w/o order(Hong et al., 2022)	0.968	0.905	-	-	340M
BROS w order(Hong et al., 2022)	0.966	0.974	-	-	340M

Table 2: Results on the CORD dataset evaluated at link level and at group level.

Model	Before FT		After FT	
	F1	ARI	F1	ARI
Rel graph (ours)	0.917	0.957	0.925	0.960
w/o feat prop	0.903	0.948	0.914	0.955
sum+linear	0.901	0.948	0.915	0.955

Table 3: Impact of the finetuning removing the GT information for the entity generation.

The results are presented in Table 2. To the best of our knowledge, there are no published works that address exclusively the EB and EL tasks, since they are usually combined with the entity tagging task. Consequently, although they are not fully comparable, we decided to include the results of two state-of-the-art end-2-end models that perform ET, EB, and EL, Spade (Hwang et al., 2021b) and BROS (Hong et al., 2022). It can be observed that the proposed model outperforms the others especially on the EL task, while massively reducing the number of parameters if we compare it with BROS. Notice that for BROS we present the results with and without the text order information, as it is dependent on it. We also include the Group F1 Score and the ARI metrics so other future works can fairly compare against our model.

Finally, we also measure the processing time and the resource consumption for our model. The experiment was conducted on a machine with one NVIDIA Tesla V100 GPU, 64 GB of RAM, and 1 Intel(R) Xeon(R) Gold 6142 CPU. For the time calculation, we infer all the dataset samples using batch 1 and compute the average time. We take into account also the preprocessing time since the input files are loaded, including the parsing, feature extraction, and graph generation. The resulting time per image is 0.25 seconds (0.15 for preprocessing and 0.10 for inference and postprocessing), with a low GPU memory consumption of around 1300 megabytes.

5 Conclusions and Future Work

In this work we have addressed the automation of information extraction on unstructured documents, given as inputs the predictions from an OCR engine and an entity tagging model, and focusing on two tasks, entity building and entity linking. We have justified the suitability of GNNs for the considered use case and proposed a model based on this approach. This model tackles the problem in two stages that are strongly connected by using the concept of hypernodes. We have also proposed a novel strategy of propagating the features from the segment nodes to the entity nodes in an optimal way. The results of the conducted experiments demonstrate that the proposed model is suitable for solving the tasks, and that the proposed feature propagation strategy is optimal and outperforms other approaches. In addition, we have compared our model with other state-of-the-art methods that perform the EB and EL tasks using the public benchmark CORD and, although the models are not fully comparable, it can be observed that our model achieves state-of-the-art results with an extremely lower number of parameters.

Future work will focus on expanding the application of the model to address also the ET task. To this end, new types of features could be considered, based on text or image, as we believe that the layout information is not enough to solve ET task. In addition, we will keep enhancing the current capabilities of the model, exploring new ways of propagating the features, improving the postprocessing, and optimizing the GNN architectures.

References

Matthias Baumgartner, Daniele Dell'Aglio, and Abraham Bernstein. 2021. Entity prediction in knowledge graphs with joint embeddings. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 22–31, Mexico City, Mexico. Association for Computational Linguistics.

Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornés, and Josep Lladós. 2021. Named entity recognition and relation extraction with graph neural networks in semi structured documents. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9622–9627.

Tuan Anh Nguyen Dang, Duc Thanh Hoang, Quang Bach Tran, Chih-Wei Pan, and Thanh Dat Nguyen. 2021. End-to-end hierarchical relation extraction for generic form understanding. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5238–5245. IEEE.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, and Curtis Wiginton. 2021. Visual fudge: Form understanding via dynamic graph editing. In *International Conference on Document Analysis and Recognition*, pages 416–431. Springer.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. In *Neural networks: the official journal of the International Neural Network Society 107*, volume 107, pages 3–11. Elsevier.

Zhangxuan Gu, Changhua Meng, Ke Wang, Jun Lan, Weiqiang Wang, Ming Gu, and Liqing Zhang. 2022. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4583–4592.

Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. 2002. Cluster validity methods: part i. *SIGMOD Rec.*, 31:40–45.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775.

Wonseok Hwang, Hyunji Lee, Jinyeong Yim, Geewook Kim, and Minjoon Seo. 2021a. Cost-effective end-to-end information extraction for semi-structured document images. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2021b. Spatial dependency parsing for semi-structured document information extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 330–343.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.

Sammy Khalife and Michalis Vazirgiannis. 2019. Scalable graph-based method for individual named entity identification. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 17–25, Hong Kong. Association for Computational Linguistics.

Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2021. Donut: Document understanding transformer without ocr. In *arXiv preprint arXiv:2111.15664*.

Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. 2021. Structext: Structured text understanding with multi-modal transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1912–1920.

Chuwei Luo, Yongpan Wang, Qi Zheng, Liangchen Li, Feiyu Gao, and Shiyu Zhang. 2020. Merge and recognize: A geometry and 2D context aware graph model for named entity recognition from visual documents. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 24–34, Barcelona, Spain (Online). Association for Computational Linguistics.

Eirini Papagiannopoulou, Grigorios Tsoumakas, and Apostolos Papadopoulos. 2021. Keyword extraction using unsupervised learning on the document's adjacency matrix. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 94–105, Mexico City, Mexico. Association for Computational Linguistics.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

Nishant Prabhu, Hiteshi Jain, and Abhishek Tripathi. 2021. Mtl-foun: A multi-task learning approach to form understanding. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 377–388. Springer.

Guozhi Tang, Lele Xie, Lianwen Jin, Jiapeng Wang, Jingdong Chen, Zhen Xu, Qianying Wang, Yaqiang Wu, and Hui Li. 2021. Matchvie: Exploiting match relevancy between entities for visual information extraction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1039–1045.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, volume 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *arXiv preprint arXiv:1710.10903*.

María Villota, César Domínguez, Jónathan Heras, Eloy Mata, and Vico Pascual. 2021. Text classification models for form entity linking. In *arXiv preprint arXiv:2112.07443*.

Qian Wan, Luona Wei, Xinhai Chen, and Jie Liu. 2021. A region-based hypergraph network for joint entity-relation extraction. In *Knowledge-Based Systems*, volume 228, page 107298. Elsevier.

Jiapeng Wang, Lianwen Jin, and Kai Ding. 2022. Lilt: A simple yet effective language-independent layout transformer for structured document understanding.

Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. 2020. Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. In *Empirical Methods in Natural Language Processing (EMNLP)*, volume abs/2010.11685, pages 898–908.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. In *arXiv preprint arXiv:2104.08836*.

Yue Zhang, Bo Zhang, Rui Wang, Junjie Cao, Chen Li, and Zuyi Bao. 2021. Entity relation extraction as dependency parsing in visually rich documents. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Zhenrong Zhang, Jiefeng Ma, Jun Du, Licheng Wang, and Jianshu Zhang. 2022. Multimodal pre-training based on graph attention network for document understanding. In *arXiv preprint arXiv:2203.13530*.