DOCUMENT STRUCTURE AWARE RELATIONAL GRAPH CONVOLUTIONAL NETWORKS FOR ONTOLOGY POPULATION

Abhay M Shalghar*, Ayush Kumar* & Shobha G
RV College of Engineering
Bengaluru, India
{abhayms.cs18,ayushkumar.cs18,shobhag}@rvce.edu.in

Balaji Ganesan & Aswin Kannan
IBM Research
Bengaluru, India
{bganesa1,aswkanna}@in.ibm.com

Akshay Parekh
IIT Guwahati
Guwahati, India
akshayparakh@iitg.ac.in

ABSTRACT

Ontologies comprising of concepts, their attributes, and relationships are used in many knowledge based AI systems. While there have been efforts towards populating domain specific ontologies, we examine the role of document structure in learning ontological relationships between concepts in any document corpus. Inspired by ideas from hypernym discovery and explainability, our method performs about 15 points more accurate than a stand-alone R-GCN model for this task.

1 INTRODUCTION

Ontology induction (creating an ontology) and ontology population (populating ontology with instances of concepts and relations) are important tasks in knowledge based AI systems. While the focus in recent years has shifted towards automatic knowledge base population and individual tasks like entity recognition, entity classification, relation extraction among other things, there are infrequent advances in ontology related tasks.

Ontologies are usually created manually and tend to be domain specific i.e. meant for a particular industry. For example, there are large standard ontologies like Snomed for healthcare, and FIBO for finance. However there are also requirements for cross domain ontologies for applications in data protection, meta-data management and data discovery in Data Fabric.

However, creating and maintaining ontologies and related knowledge graphs is a laborious process. There have been few efforts in recent years to automate ontology population. Chen et al. (2018) introduced constraints on relations that should be part of ontologies. Guan et al. (2019) et al proposed a method for Link Prediction in n-ary relational data. Shen et al. (2020) used a R-GCN model for the related task of taxonomy expansion. We continue with the recent trend to use R-GCN to predict relation type (link type prediction) between entities.

In this work, we focus on relation extraction using relational graph neural networks (RGCN) for ontology population. These ideas stem from observations of knowledge based systems in industrial applications. Many of these datasets used for this task are usually derived from formatted documents, but the document structure information is often discarded by converting to triples and

*Equal Contribution

other dataset formats. So for the relation extraction task, as shown in Figure 1 instead of starting from plain text sentences, we explore incorporating the *document structure* information to improve accuracy in R-GCN models.

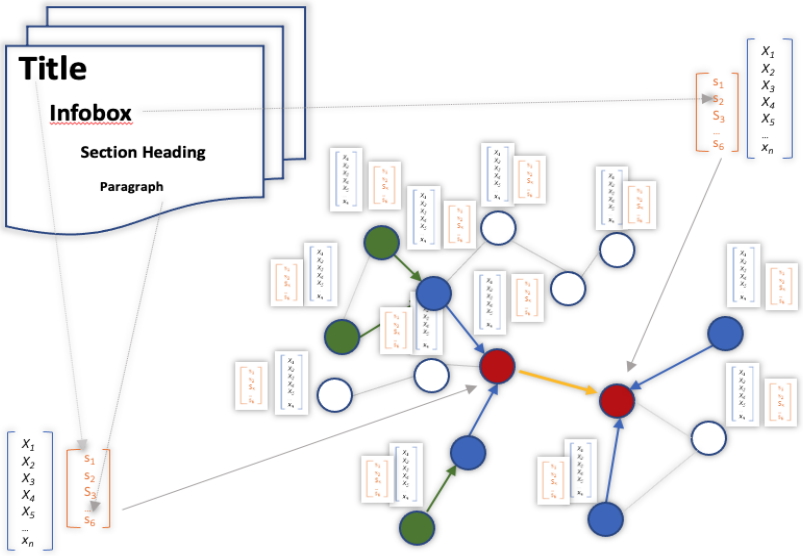


Figure 1: Document Structure can improve relation extraction using RGCN.

We summarize our contributions in this work as follows:

- We propose a document structure measure (DSM) similar to other measures in hypernmy discovery to model ontological relationships between entities in documents.
- We experiment with different methods to incorporate the DSM vectors in state of the art relational graph convolutional networks and share our results.

2 Related Work

Chen et al. (2018) incorporated hierarchy for relation extraction in Ontology Population. Our document structure measures differs from such works in that these don't just capture hierarchy but are very general and encompass document summary, section titles, bulleted lists, highlighted text, infobox etc. We also extend this concept of document structure to personal data (binary).

Nagpal et al. (2022) described a method to use rule based systems along with a neural model to improve fine grained entity classification. Vannur et al. (2020) described a method to augment the training data for better personal knowledge base population. Ganesan et al. (2020) uses graph neural networks to predict missing links between people in a property graph.

Chiticariu et al. (2010) presented a System T method of rule based information extraction. They have also shown a way to determine if the extraction information is relevant to a domain or otherwise Wang et al. (2017). Madaan et al. (2017) has specifically dealt with extracting titles, section and subsection headings, paragraphs, and sentences from large documents. Additionally Agarwal et al. (2017) extracts structure and concepts from html documents in compliance specific applications.

Zhang et al. (2018) proposed an extension of graph convolutional network that is tailored for relation extraction. Their model encodes the dependency structure over the input sentence with efficient graph convolution operations, then extracts entity-centric representations to make robust relation predictions. We derive some of our inspiration from the TaxoExpan work Shen et al. (2020) that also uses RGCN to find hyponyms for taxonym expansion. We note that our work is however significantly more comprehensive in both the types of rules and number of such relationship extractions (not restricted to hypernym discovery).

3 LEARNING DOCUMENT STRUCTURE

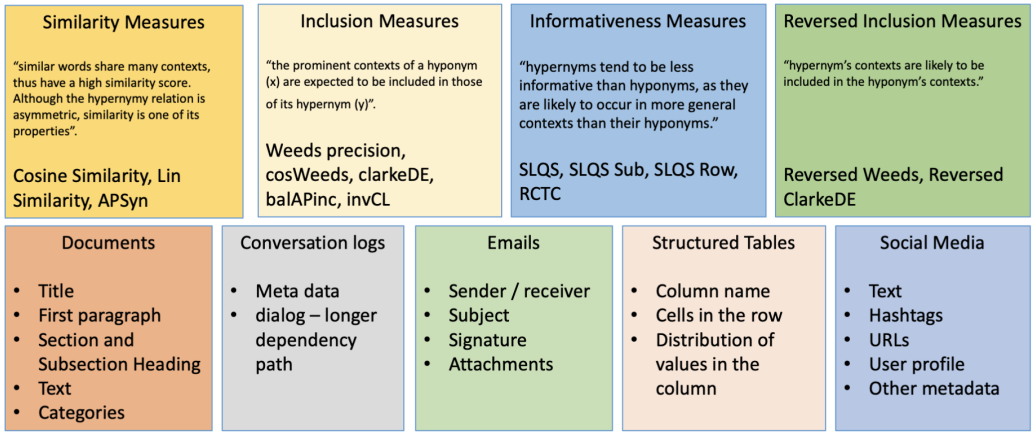


Figure 2: Document Structure in different document formats.

Chen et al. (2018) showed that some relations prove to be rich enough to go into the T-box (relations between Concepts), while other relations need only be present in the A-box. A corollary to the above classification of relations is the use of different measures to solve the hypernym discovery problem. These could be broadly classified as Similarity Measures (Cosine, Lin, ApSyn), Inclusion Measures (ClarkeDE, balAPinc, invCL), Informativeness Measures (SLQS), and Reversed Inclusion Measures (Reversed ClarkeDE).

Inspired by these measures, we introduce a new measure based on document features and structure that helps identifying relationships for knowledge base population. We broadly term these under the umbrella of Document Structure Measures (DSM) and further deploy a Relational Graph Convolutional Network (RGCN) to learn the attributions of the same towards our larger goal. To validate the effectiveness of using DSM, we perform experiments on Wikipedia pages of entities in our datasets. Our focus on unstructured text and personal data in Wikipedia pages is primarily due to restrictions on using emails or social media posts for research. However, the approach is general to any type of document with a template. Some easy examples include corporate documents, emails, tweets, logs, and chat transcripts.

3.1 DOCUMENT STRUCTURE MEASURE (DSM)

To motivate the notational aspects, even before defining our model, we shed light on the broad term called "document features". These can correspond to positioning of words, their hierarchy in a document, and even their formatting. We note that some of the features from embedding can also be cast into this framework, but we precisely mention that our structural features are generic and not tied to the aspects of any specific language. As an example, text within bullets can represent important data such as sensitive data, hypernyms, or even a short excerpt. As another isometric viewpoint, personal data like anonymized or pseudonymized credit card numbers can be presented in brackets, found in footnotes, and even given by fine worded text (small text) to mask readability.

In our scope of work corresponding to ontologies, we choose "personal information" as our running example. These can say point to names, addresses, biographical details etc. and covers a very broad range of labels. As the most crucial aspect, we define the metric called importance probability, ρ. This measure signifies the factor by which a specific occurrence of text corresponds to a classification label in the space defined by document structure measure. Say, in our running example, if the text "xxxx–56" is found within brackets, we attribute a higher probability score ρ to its classification as personal data. As another example in a slightly different context, say if the word "red" appears in bullets, and the word "color" appears in the main text, we attribute a higher probability score to the hypernym-hyponym relationship between the two words.

Next, we define vectors, $v_a(x)$, $v_b(x)$, and $v_c(x)$ to define the document structure measure corresponding to a word x. Here, $v_a(x)$ and $v_b(x)$ refer to relational measures and $v_c(x)$ refers to an absolute measure. More importantly, v_a refers to a higher hierarchy and v_b refers to a lower hierarchy. Say, in the example of "red/color" relationship above, let "x" denote "red". Then, $v_a(x)$ takes a lower value (say 0) and v_b takes a higher value (say 1). Without loss of generality (this can take non-integral values also), we assume that the values taken by v_a, v_b, and v_c are 0/1 (similar to an indicator function). Next, we further specify the fine details on the quantification of these vectors based on the logical document structure rules. Noting space considerations, we discuss in detail in about relational DSM vectors in the next subsection. Details regarding absolute DSM vectors can be found in the supplementary material.

3.2 Relational DSM Vectors

We start with an example of bulleted text. Without loss of generality, we split the entire document into multiple paragraphs ensuring that each paragraph at the most contains only one set of bulleted text. Bulleted text are more probable to be hyponyms (Lists / Enumerations also included) as with the the instance below.

"X contains the following:

- *X1*
- *X2"*

Here, X1 and X2 are hyponyms of X. Note that hypernym-hyponym exactly corresponds to one relationship where v_a denotes the former and v_b denotes the latter. This can be easily generalized to other relationships and document structure rules. Given two words x and y, their probability of their hierarchical (hypernym-hyponym in this case) relationship specific to a document feature i (in this case bulleted list) can be stated as follows:

$$\rho^k(x,y) = \frac{\sum_{j=1}^{m^k} v_a^{j,k}(x) \cap v_b^{j,k}(y)}{\sum_{j=1}^{m^k} v_a^{j,k}(x)}.$$

Here, j refers to an entity mention, which in this case is the presence of the word x in paragraph j that contains a document feature, for example, a bulleted list. Note that the suffix k corresponds to the index of the document structure measure. Say for the three measures bulleted text, footnotes, and title, the indexes are respectively k = 1, 2, and 3. The notation for the vectors v_a, v_b, and v_c follow similar suite. In case a paragraph j does not contain the document feature, the corresponding entries are 0 for both the numerator and denominator sub-portions. In some cases a document feature may contain more than one occurrence of a hypernym / word. We merely consider the above expression to have an indicator function and do not pursue on the track of multiple occurrences. Usually, the vectors v_a and v_b are specified in an opposite sense, where say $v_a^{i,j}(x) = 1$, $v_b^{i,j} = 0$ and vice-versa. Say, if a word occurs at the text preceding the bullets, they are directed towards the indicator function in v_a and if they occur within the sub-bullets, those are accounted towards the indicator function in v_b. However, it can also be true that some words can be present in both the preceding text or sub-bullets, leading to both v_a and v_b taking the value of 1. Generalizing the above to all possible document features, we have the following.

$$\rho(x,y) = \sum_{k=1}^{Kr} w^k f^k(n_x, n^{k,x}) \rho^k(x,y),$$

where w^k refers to the weight assigned to each document feature (pre-set by the user depending on the application) and $f^k(.)$ denotes an importance function corresponding to the occurrence of the entities both in the presence of the context and overall (presence and absence included). More specifically, $n^{k,x}$ refers to the number of times the word x has occurred in text preceding the document feature in the document and n_x denotes the overall number of times the word x has appeared in the document. For a complete set of rules and document features, please refer to the supplementary material.Alternatively, instead of a summation, ρ can be kept as a vector and given by $\rho(x,y) = \left\{\rho^k(x,y)\right\}_{k=1}^{Kr}$, where Kr denotes the total number of relative features.

3.3 GENERATING DSM VECTORS

This in section we'll describe our information retrieval based approach to generate document structure measure vectors for each pair of entities in the Wikipeople and TACRED datasets. We use Wikipedia pages to produce the document structure, but any corpus of documents with a templated format like electronic health records, customer invoices, legislation, regulatory documents could be used to generate these DSM vectors. Additionally, as discussed in Section 3, data sources like tweets, chat conversations, emails also display templated document structure and hence can be processed for generating DSM vectors using the method described in this section.

As discussed in Section 4.1 and shown on Table 1, Wikipeople and TACRED are relational datasets and unstructured datasets respectively, which we have converted into graph formats for this work. Hence in the case of Wikipeople, we're only concerned with Person to Person relations and in TACRED, only on relationships between people and organizations.

In order to generate document structure measure for the relationship between any of the above pairs, we begin by computing the frequency of the occurrence of pairs of entities in the corresponding Wikipedia pages of these entities. We index different parts of the Wikipedia pages in a search index and retrieve frequencies of occurrences using the surface forms of the entities (names) as queries.

We observe that an indexing system that leverages document structure and ontological information to enrich the raw documents, has higher performance than regular document indexing. We treat the search index as a black box (which is usually the case in many cloud based implementations) and make all our improvements to the documents being indexed. We leave further improvements to the search index as future work.

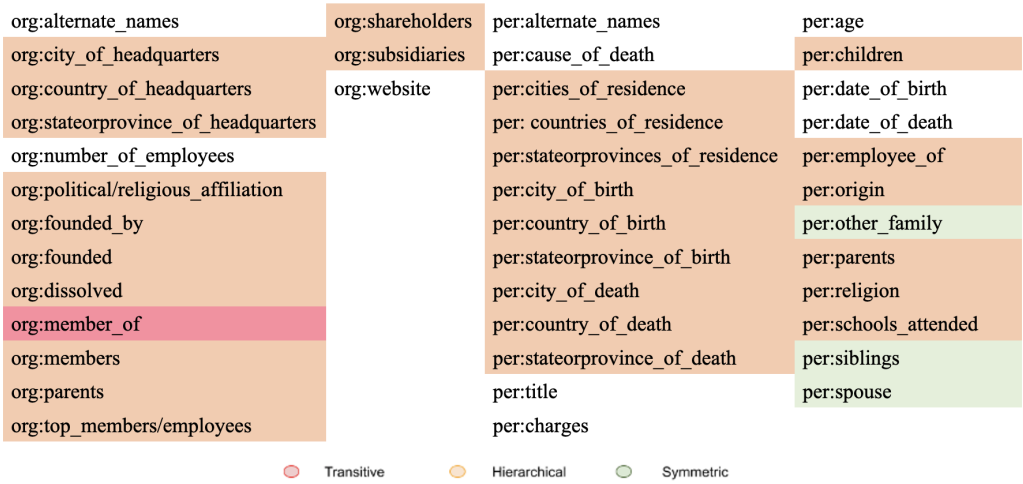


Figure 3: Relations in Tacred

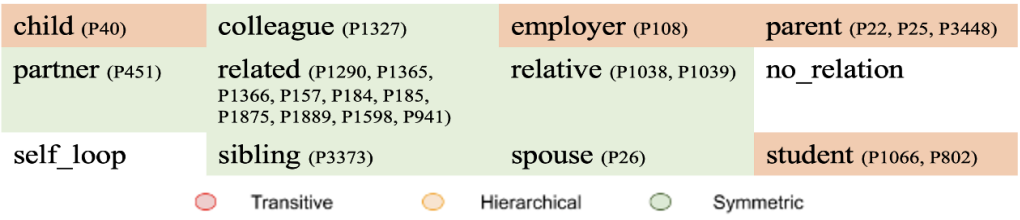


Figure 4: Relations in Wikipeople (with Wikidata property ids)

4 EXPERIMENTS

4.1 DATASETS

We have created our datasets from the Wikipedia pages for the entities in Wikipeople and TACRED datasets. The original relations in the datasets are as shown in Figures 4 and 3.

We first used NLTKLoper & Bird (2002) to split the text into sentences and tokenize the sentences. We used the method described in Chiticariu et al. (2010) to identify the span of entity mentions in Wikipedia pages. We call these scripts as Personal Data Annotators. This method requires creation of dictionaries each named after the entity type, and populated with entity mentions. We ran the Personal Data Annotators on these sentences, providing the bulk of the annotations that are reported in Table 1.

	Wikipeople	**Tacred**
Documents	16845	4142
Nodes	9760	11161
Node Types	1	2
Relation Types	13	22

Table 1: Statistics on datasets adopted for relation type prediction in graphs

We consider the following document structures in this experiment. Title, introduction section, infobox, section (and subsection) headings and text. We could include other structures like category, in and outlinks in the case of Wikipedia. We leave that for future work. To calculate the document structure measure for a relation, {subject, relation, object}, we use a weighted average of the frequency of their occurrences in the corpus. The weight of each document structure is manually assigned by us, but it could be learned from the corpus as well. We leave this also to future work.

This approach does not take the context of the entity mentions while assigning labels and hence the data is somewhat noisy. However, labels for name, email address, location, website do not suffer much from the lack of context and hence were annotated using this tool.

4.2 EXPERIMENTAL SETUP

We experiment on the R-GCN models proposed by Schlichtkrull et al. (2018); Shen et al. (2020). These have been used in the cases of link prediction and hyponym based taxonomy expansion respectively. For training R-GCN and our improvements, we use a single V100 GPU with 16GB memory. We use the RGCN implementation in DGL Wang et al. (2019) and the elastic search instance from a cloud provider.

4.3 LINK TYPE PREDICTION WITH DSM

Assuming that the network is defined appropriately, the hidden representation for each node i at (l+1)th layer can be written as follows.

$$h_i^{l+1} = \sigma\left(W_0{}^{(\mathrm{l})} h_{\mathrm{i}}{}^{(\mathrm{l})} + \sum_{r \in R} \sum_{j \in N_{\mathrm{i}}^{\mathrm{r}}} \frac{W_{\mathrm{r}}{}^{(\mathrm{l})} h_{\mathrm{j}}{}^{(\mathrm{l})}}{C_{\mathrm{i,\,r}}}\right). \tag{1}$$

BASELINE

We use the heterograph version of R-GCN from DGL. One of the limitations of using the heterograph version is the requirement to add a self-loop for all the singleton nodes in the graph. As discussed in Section 4.1, we train all our R-GCN models by consuming the datasets as attributed graphs. In the case of Wikipeople, the relations are of the form (Person, relation_type, Person), and in the case of TACRED, we have relations of the form (Person, relation_type, Person), (Person, relation_type, Org), (Org, relation_type, Person) and (Org, relation_type, Org). The baseline RGCN results seem comparable to results on attributed multiplex heterogenous networks in Cen et al. (2019), Vannur et al. (2020).

REGULARIZATION

We began by incorporating the DSM scores for each pair of relationship in the cross entropy loss function similar to any regularization parameter. As shown in Table 2, this did not yield any improvement in performance.

HIDDEN LAYER

We then tried incorporating our DSM vectors by adding another hidden layer to the network. We pass the output of the first hidden layer in RGCN to this new layer and update the representation of entities in the (l+1)th layer with the DSM vectors. Recall that the hidden representation of entities in (l+1)th layer in R-GCN can be formulated as shown in Equation 1.

We updated the first hidden layer output for an entity my multiplying with the corresponding aggregate DSM score of all the edges incident on the entity. This method gave marginal improvement in some relationship types, but hurts the performance in general. We observe that using the aggregate scores from DSM vectors for all edges does not help to capture the correlation between the document structure and the edge. Hence we tried incorporating the DSM vectors in the message passing layer, which we describe in the next section.

EDGE WEIGHTS

Here, we describe a way to incorporate document structure measure in the node representation. We add a key embodiment to Equation 1 in the form of ρ, corresponding to document structure measures. Note that we pre-compute ρ based on samples from the training data. For the expression below, we use the general vector version of ρ, with individual ρ^k's contributing towards the activation θ. This can be replaced by the scalar ρ^k, where the activation can be changed accordingly. Note that the dimension of inputs and outputs corresponding to the layer can be re-defined accordingly with respect to the dimensions of x and y.

$$h_{\mathrm{i}}^{\mathrm{l+1}} = \sigma\left(W_0^{(\mathrm{l})} h_{\mathrm{i}}^{(\mathrm{l})} + \sum_{r \in R} \sum_{j \in N_{\mathrm{i}}^{\mathrm{r}}} \frac{W_{\mathrm{r}}^{(\mathrm{l})} h_{\mathrm{j}}^{(\mathrm{l})}}{C_{\mathrm{i,\,r}}}\right) + \theta(\rho^i h_{\mathrm{i}}^{(\mathrm{l})}). \tag{2}$$

We implemented the above by updating the edge weights in the forward pass of RGCN. A similar approach had been taken in Qin et al. (2021) but the initial edge weights there were based on the node type and were updated with self-adversarial training. We instead incorporate the DSM vector during message passing by adding DSM scores to each corresponding edge weight. As discussed in the next section, this improves RGCN accuracy in both the datasets, with the Wikipeople increasing by 15 points.

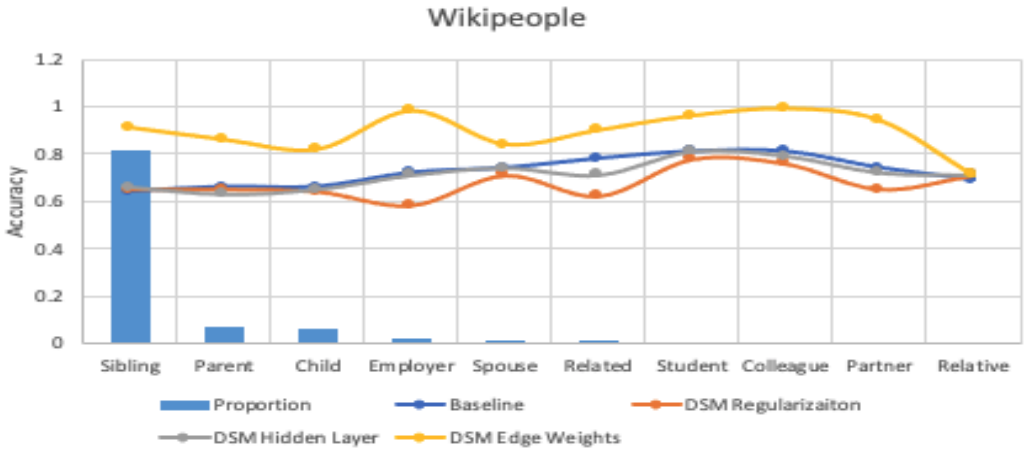
4.4 RESULTS

We report the performance of RGCN models as shown in Table 2. Our RGCN model with DSM edge weights performs about 15% better than the baseline model. We also observe that using the DSM vectors for regularization and as a hidden layer do not seem to help and those models performed worse than the baseline.

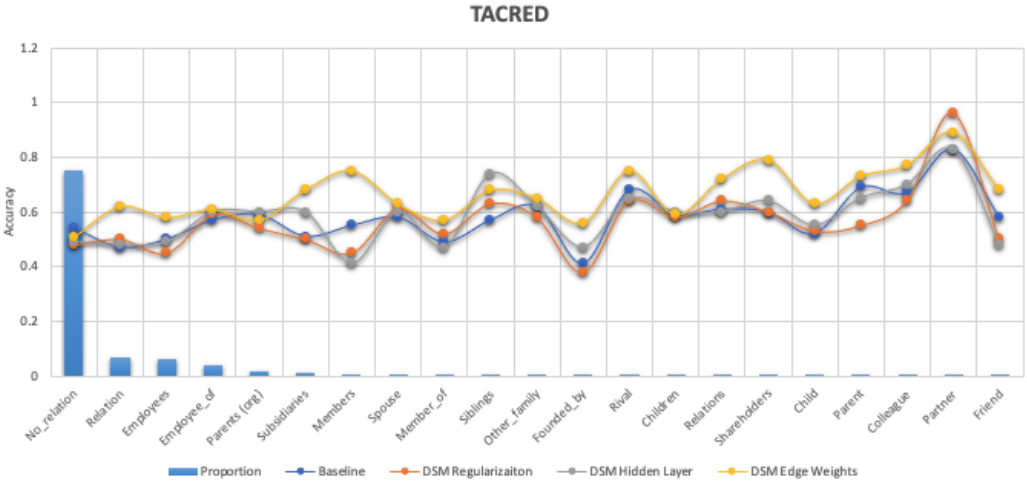
Model	**Wikipeople**	**TACRED**
RGCN	0.73	0.58
RGCN+DSM regularization	0.71	0.57
RGCN+DSM hidden layer	0.67	0.59
RGCN+DSM edge weights	**0.88**	**0.67**

Table 2: Accuracy of RGCN models on the WikiPeople and TACRED Datasets

In Figures 5a and 5b, we overlay the accuracy of the models on the class distribution. Sibling relationship type dominates the Wikipeople dataset and the use of document structure helps this type too. We believe that sibling relationship (which are symmetric) benefits from document structure



(a) Wikipeople class distribution overlaid on accuracy of models



(b) TACRED dataset distribution overlaid on accuracy of models

Figure 5: Class wise performance overlaid on accuracy of models

because it gets the boost from the way we index sentences and associate the document and section title with the sentences. We also observe that no_relation class which might rarely occur other than as text (and not in the infobox of a wikipedia page for example), does not get much improvement in accuracy.

We observe similar distribution in accuracy improvement for tacred, though the improvement is relatively less compared to Wikipeople. This seems to be because of the number of no_relation examples in this dataset.

CONCLUSION

In this work, we proposed using the document structure to improve ontological relation extraction from unstructured documents. In particular, we described a document structure measure (DSM) vector that can be incorporated while training a relational graph convolutional network (RGCN). We theoretically explained the need for such a measure in ontology population, and conducted experiments on different ways to incorporate the document structure measure in RGCNs. Our experiments show good improvement in RGCN performance while using our approach on the Wikipeople and TACRED datasets.

REFERENCES

Arvind Agarwal, Balaji Ganesan, Ankush Gupta, Nitisha Jain, Hima P Karanam, Arun Kumar, Nishtha Madaan, Vitobha Munigala, and Srikanth G Tamilselvam. Cognitive compliance for financial regulations. *IT Professional*, 19(4):28–35, 2017.

Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul 2019.

Muhao Chen, Yingtao Tian, Xuelu Chen, Zijun Xue, and Carlo Zaniolo. On2vec: Embedding-based relation prediction for ontology population. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 315–323. SIAM, 2018.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick R Reiss, and Shivakumar Vaithyanathan. Systemt: an algebraic approach to declarative information extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

Balaji Ganesan, Gayatri Mishra, Srinivas Parkala, Neeraj R Singh, Hima Patel, and Somashekar Naganna. Link prediction using graph neural networks for master data management. *arXiv preprint arXiv:2003.04732*, 2020.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. Link prediction on n-ary relational data. In *The World Wide Web Conference*, pp. 583–593, 2019.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.

Nishtha Madaan, Hima Karanam, Ankush Gupta, Nitisha Jain, Arun Kumar, and Srikanth Tamilselvam. Visual exploration of unstructured regulatory documents. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion*, pp. 129–132, 2017.

Abhinav Nagpal, Riddhiman Dasgupta, and Balaji Ganesan. Fine grained classification of personal data entities with language models. In *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pp. 130–134, 2022.

Xiao Qin, Nasrullah Sheikh, Berthold Reinwald, and Lingfei Wu. Relation-aware graph attention model with adaptive self-adversarial training, 2021.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*, pp. 486–497, 2020.

Lingraj S Vannur, Lokesh Nagalapatti, Balaji Ganesan, and Hima Patel. Data augmentation for personal knowledge graph population. *arXiv preprint arXiv:2002.10943*, 2020.

Chenguang Wang, Doug Burdick, Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, and Huaiyu Zhu. Towards re-defining relation understanding in financial domain. In *Proceedings of the 3rd International Workshop on Data Science for Macro–Modeling with Financial and Economic Datasets*, pp. 1–6, 2017.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.