Large-Scale Relation Extraction Dataset using Wikidata

Paka Hema Chandra Kumar

IIT Guwahati

20-05-2019

Relation Extraction (RE)

- Extraction of the semantic relation between various entities in the text
- e.g., *Narendra Modi* is the current Prime Minister of *India*.

Limitations for Large-Scale RE

- need annotated data
- expensive to produce
- hard to extend to new relations, in the sense that detecting new relation types requires new training data

Why Distant Supervision?

- uses Knowledge base(KB) to obtain training data
- reduces human effort
- can be extended to new relations
- availability of large-scale knowledge bases such as DBPedia, Wikidata etc.

Problem in Existing Distant Supervision Approaches

- NYT dataset by Riedel et al.,[1] is used to train and evaluate the RE systems
- consists of only 53 relations
- more than half of the relations have less than 100 sentences for training
- testing data consists of sentences for 32 relations only

Our Goal

- To provide a dataset with more number of relations
- To know the effect of increasing relations and relations with less training data on existing models

Preliminaries

- Knowledge bases typically store data as statements for an entity which can be extracted into triples
- triples are of the form (head entity*(e1)*, relation*(r)*, tail entity*(e2)*)
- e.g.,(Narendra Modi, Prime Minister, India)
- **Distant supervision assumption:** If there exists a relation between two entities according to the knowledge base, then all the sentences which contain these two entities express the relation.
- e.g., *Narendra Modi* is the current Prime Minister of *India*.

- **Bag of sentences(bag):** All the sentences with same head and tail entities comprise a bag and the relation between the entities is the relation label of that bag.

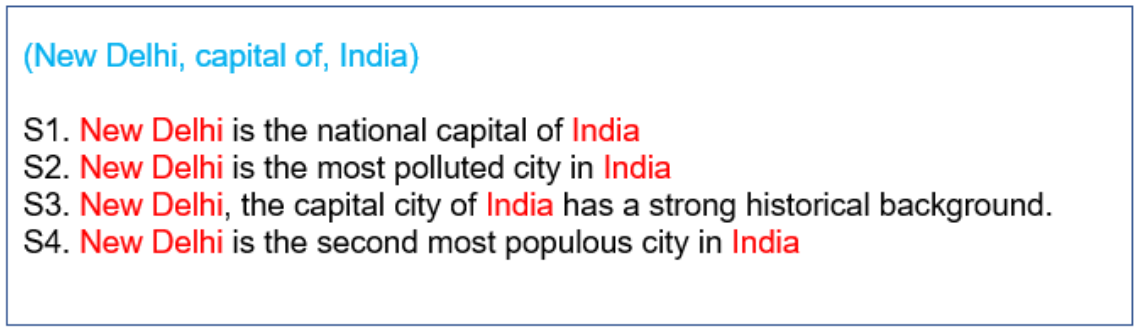

Figure 1: Example for Bag of sentences

RE Task Formulation

- multi-class classification problem
- given an entity pair with a bag of sentences, the trained classifier needs to predict the relation label of the bag(bag level evaluation
- Consider N bags $M = \{M_1, M_2 \ldots, M_N\}$ and $y = \{y_1, y_2 \ldots, y_N\}$ are corresponding bag labels.
- $M_j = \{m_1^j, m_2^j \ldots, m_n^j\}$ are the bag of sentences.
- predict y_j as the relation label for bag M_j

Construction of New Dataset

- knowledge base : Wikidata
- Wikipedia corpus to generate sentences

Extraction of Triples from Wikidata

- Triples are of the form : (item, property, value)
- e.g., (Narendra Modi, instance of , human)
- e.g., (Narendra Modi, place of birth , Vadnagar)

Figure 2: Wikidata page of Narendra Modi

Hierarchy of classes with *subclass of* (P279) property

- sub-classes of Person: Creator, Worker, Artist, ...
- sub-classes of Organization: Company, International Organization, Business, ...

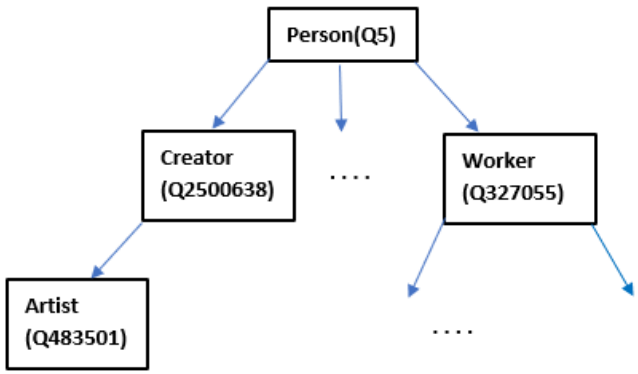

Figure 3: Person(human(Q5))

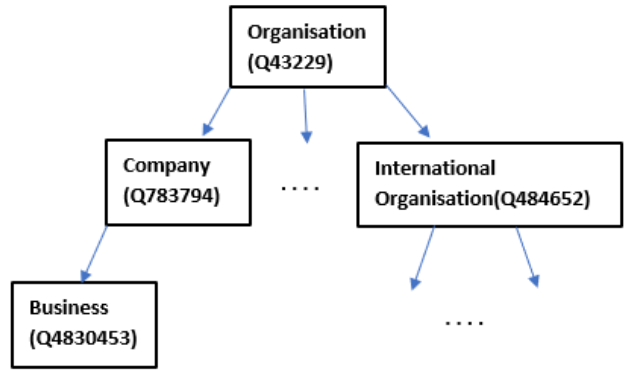

Figure 4: Organization(Q43229)

Identify Items of Type Person, Organization and Location

- extracted 643 sub-classes of Person, 18417 sub-classes of Organization and 11114 sub-classes of Location
- *"instance of"*(P31) any of the class or sub-classes of Person, Organization and Location
- e.g., Narendra Modi is an *instance of* human, a sub-class of Person, so the entity is of type Person

Statistics of Wikidata

- Wikidata has more than 38 million items and 5147 properties
- extracted more than 15 million triples and there are 466 different properties(relations) in them

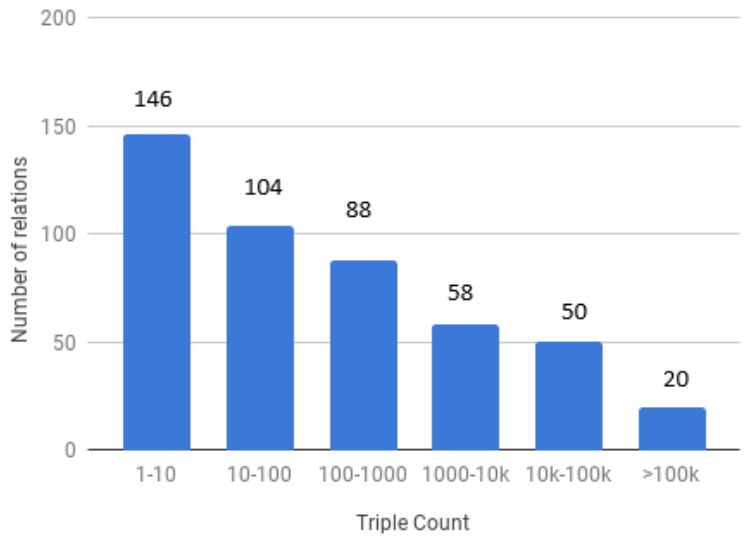

Figure 5: Number of relations in different ranges of triple count (Triples obtained from Wikidata)

Extraction of Sentences from Wikipedia

- Wikipedia contains 94,635,369 sentences
- 22,818,790 sentences contain more than two entities
- extracted sentences for 804,680 bags(triples) and they contain 324 relations in them
- no bag of sentences for 122 relations

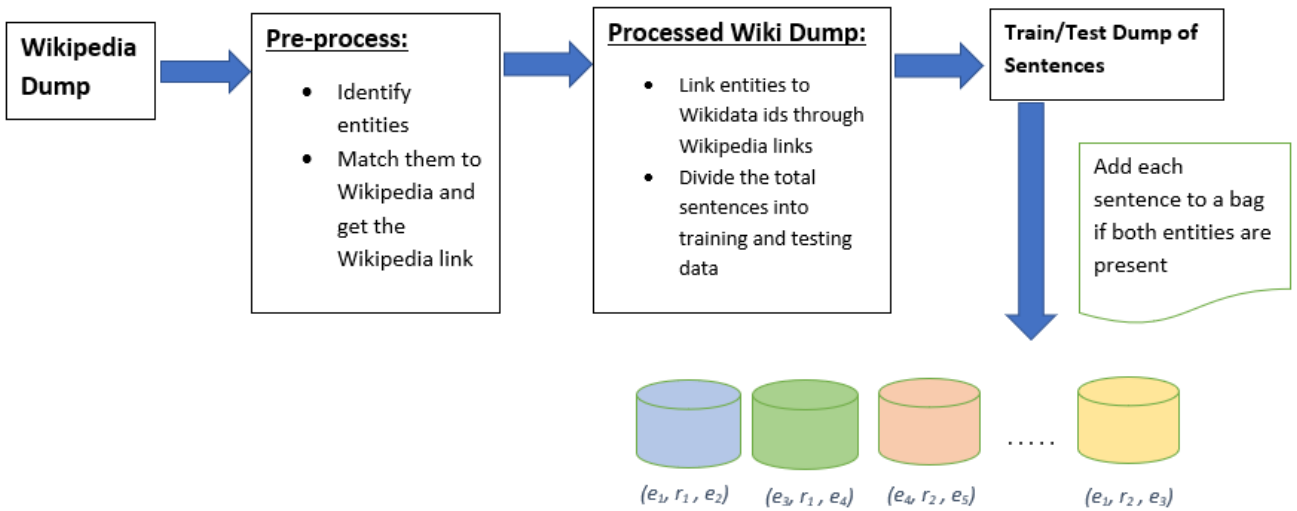

Figure 6: Pipeline for creation of dataset

Negative Training/Testing Data

- sentences in which the entities does not have any relation according the knowledge base
- 700,000 entity pairs in training data and 100,000 in testing data
- assigned “NA” relation

NYT Dataset

- developed by Riedel et al.[1]
- New York Times articles of year 2005-06 for training and 2007 for testing
- knowledge base used: Freebase
- 52 relations and "NA" relation in training
- only 32 relations including "NA" have sentences in testing
- Training data: 522,611 sentences (385664 for "NA") and 18,252 relation facts (triples)
- Testing data:172,448 sentences (94917 for "NA") and 1950 relation facts (triples)

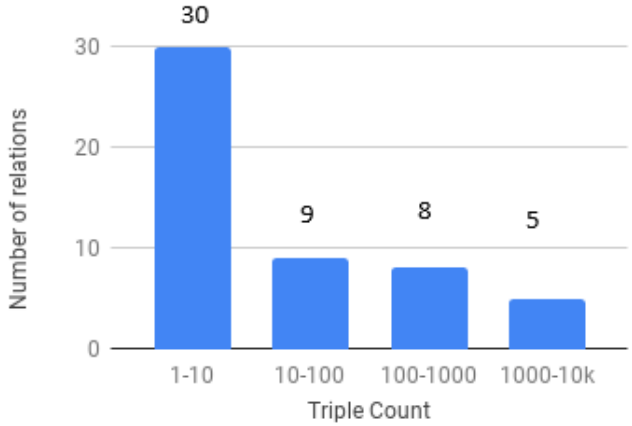

Figure 7: Relations vs Triple count(NYT)

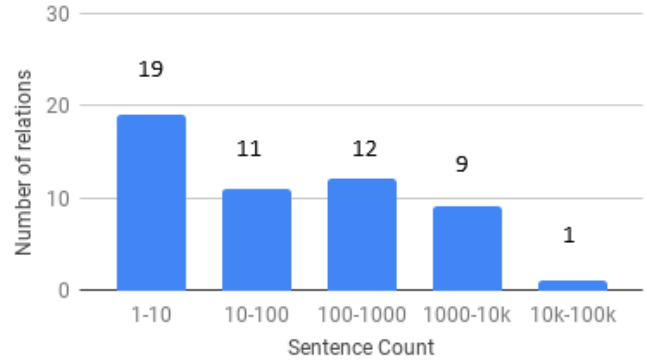

Figure 8: Relations vs Sentence count(NYT)

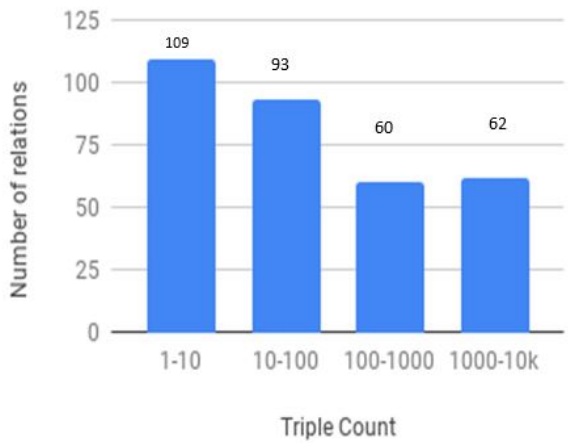

Figure 9: Relations vs Triple count

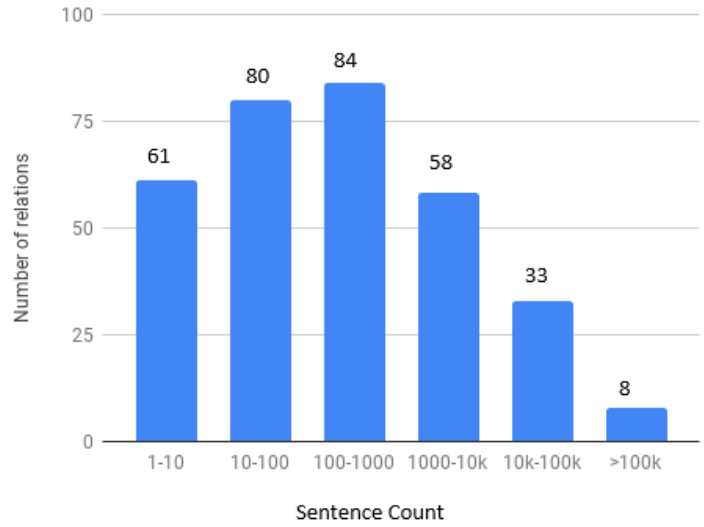

Figure 10: Relations vs Sentence count

Construction of Datasets

- four different datasets based on the number of bags(triples) that are present for a relation in training dataset
- In each dataset, a subset of relations whose bags(triple) count is greater than a threshold value
- For each relation, maximum of 100 bags(triples) are considered in the test dataset
- avoided overlapping of bags(triples) in train and test datasets

Dataset	**Minimum bags**	**# relations**	**Test bags**
dataset1	1000	60	6000
dataset2	100	122	11814
dataset3	50	149	13013
dataset4	10	215	14045

Table 1: Threshold values (based on training data), number of relations and number of test bags in each dataset

Overview of Deep Learning Models

- CNNs are used to capture the features in text
- Basic architecture consists of 4 parts:
  - Vector representation
  - Convolution
  - Max-pooling or Piece-wise max-pooling
  - Softmax output

Vector Representation

- Word embeddings
  - each word is represented as d_k dimensional vector
  - Skip-gram model is used to train word embeddings
- Position embeddings
  - relative distance between a given word and both the entities
  - two position embedding matrices are randomly initialized
  - relative distances are transformed into real valued vectors by lookup

Figure 11: Example of relative distances used in position embeddings

- vector representation of a word is the concatenation of word embedding and position embedding
- In Figure 12, d_k =5 $d_p = 2$
- dimension of each word $d = d_k + 2 \times d_p$

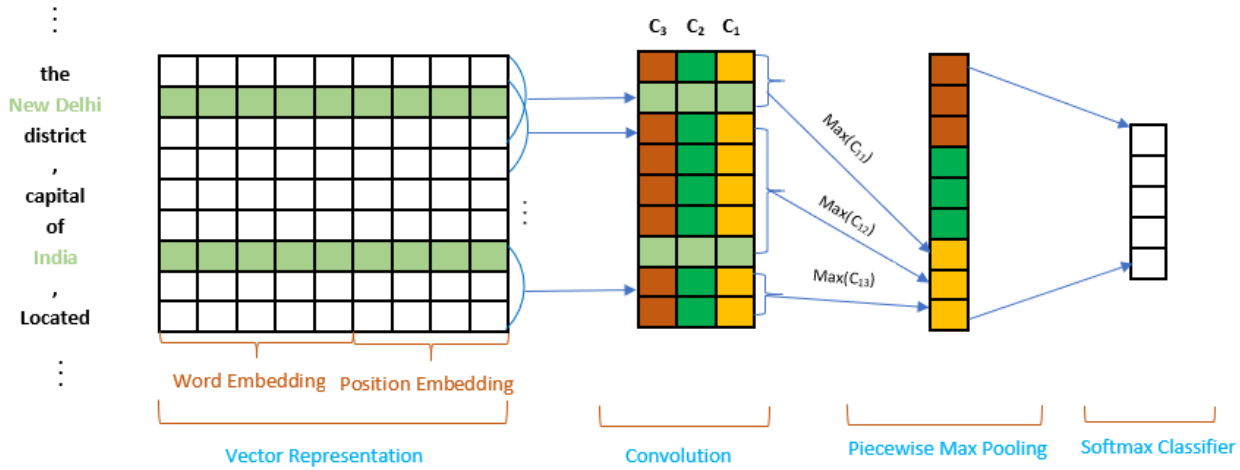

Figure 12: Architecture of PCNN model (adapted from Zeng et al.[2])

Convolution

- Convolution layer consists of n filters $\mathbf{W} = \{w_1, w_2 \ldots, w_n\}$
- Each filter $\mathbf{w} \in \mathbb{R}^{w \times d}$ where w is the length of each filter
- In Figure 12, w=3 and d =9
- The convolution output $\mathbf{C} = \{c_1, c_2, \ldots, c_n\}$ and $\mathbf{C} \in \mathbb{R}^{n \times (m+w-1)}$
- each element $c_{ij} = w_i q_{j-w+1:j}$ where $q_{j-w+1:j}$ is the w-gram of the input sentence

Piece-wise Max-pooling

- Single max-pooling
  - $p_i = max(c_i)$, $p_i \in \mathbb{R}^n$
  - insufficient to capture structural information of entities
- Piece-wise max-pooling
  - maximum value before e1, between e1 and e2 and after e2
  - $p_{ij} = max(c_{ij})$, $1 \leq j \leq 3$
  - $p_i = \{p_{i1}, p_{i2}, p_{i3}\}$, $p_i \in \mathbb{R}^{3n}$

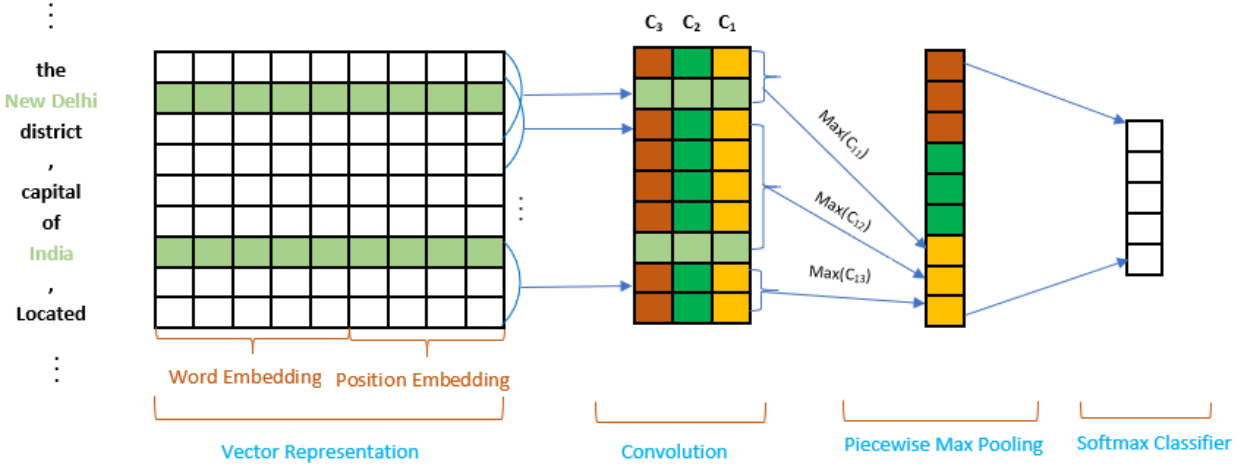

Figure 13: Architecture of PCNN model (adapted from Zeng et al.[2])

Softmax Output

- non-linear function such as tanh is applied on output $g = tanh(p)$
- softmax classifier is then applied to the vector to obtain the confidence score of each relation
- $o = W_1 g + b$ where $W_1 \in \mathbb{R}^{n_1 x 3n}$ and $o \in \mathbb{R}^{n_1}$

Update Procedure using Selectors

- Given T training bags of the form (M_i, y_i) and each sentence in a bag m_i^j
- ONE
  - sentence with maximum softmax output score
  - objective function: $J(\theta) = \sum_{i=1}^{T} log\ p(y_i|m_i^{j^*}; \theta)$ where $j^* = \arg\max_j p(y_i|m_i^j; \theta)$

- x_i denotes the output after encoding the sentence
- **s** is the combined representation of all sentences in a bag
- $\mathbf{s} = \sum_i \alpha_i x_i$
- objective function : $J(\theta) = \sum_{i=1}^{T} log\ p(y_i|s_i;\theta)$
- AVE
  - $\alpha_i = 1/|M|$
- ATT
  - weight of the sentence depends on some function e_i, which scores compatibility between the sentence and relation
  - $\alpha_i = \frac{exp(e_i)}{\sum_k exp(e_k)}$ where $e_i = x_i \mathbf{Ar}$
  - **A** is weighted diagonal matrix and **r** is the representation of relation r

Evaluation Metric: Precision-Recall Curve

- evaluation at bag(entity pair) level
- predicted relation between entities of a bag is compared against the relation in triples
- P-R curve is obtained by ranking the confidence scores of relations of all bags combined and traversing from high score to low score to measure the precision and recall at each position
- Precision P $= \frac{Number\ of\ correctly\ extracted\ relations}{Total\ number\ of\ extracted\ relations}$
- Recall R $= \frac{Number\ of\ correctly\ extracted\ relations}{Total\ number\ of\ bags}$
- F-measure F1 $= \frac{2PR}{P+R}$

Effect of Number of Bags for a Relation

- scores are low when maximum bags of 100 are considered due to very less training data for each relation
- In case of maximum bags of 10000, the model performs better only for a few relations

Max bags	**100**		**1000**		**10000**	
Dataset	**train**	**AUC**	**train**	**AUC**	**train**	**AUC**
dataset1(60)	6000	0.2036	60000	0.3494	284822	0.2745
dataset2(122)	12200	0.2570	74333	0.3456	301654	0.2416
dataset3(149)	13247	0.2647	75352	0.3381	302602	0.2438
dataset4(215)	14028	0.2527	76087	0.3144	303263	0.2253

Table 2: PCNN_ATT model: AUC scores of datasets with different upper bounds on number of bags for a relation

Comparison of Deep Learning Models

- *tail relation* refers to a relation with less than 100 bags of sentences.
- We present results and analysis on dataset2 (*non-tail dataset*) and dataset4 (*tail dataset*)
- *non-tail dataset:* 122 relations, minimum of 100 bags, maximum of 1000 bags for each relation
- *tail dataset:* 215 relations, minimum of 10 bags, maximum of 1000 bags for each relation

Dataset	**non-tail(122)**		**tail(215)**		**NYT**	
Enc/Sel	CNN	PCNN	CNN	PCNN	CNN	PCNN
ONE	0.3317	0.3434	0.2901	0.3033	0.3101	0.3198
AVE	0.3135	0.3354	0.2841	0.3027	0.3044	0.3190
ATT	0.3412	0.3456	0.2960	0.3147	0.3277	0.3408

Table 3: Comparison of AUC scores of datasets with different encoders and selectors

Effect of using Encoders with Piece-wise Max-pooling

- irrespective of the selector, model with PCNN as encoder performs better than CNN for NYT and tail dataset
- In case of *non-tail dataset*, CNN_ATT and PCNN_ATT perform almost similarly because of the ATT selector
- PCNN is useful when the available training data is less
- PCNN does not achieve much improvement in performance when sufficient training data is available and the problem due to noisy data is alleviated

Precision-Recall curve of CNN_ATT and PCNN_ATT

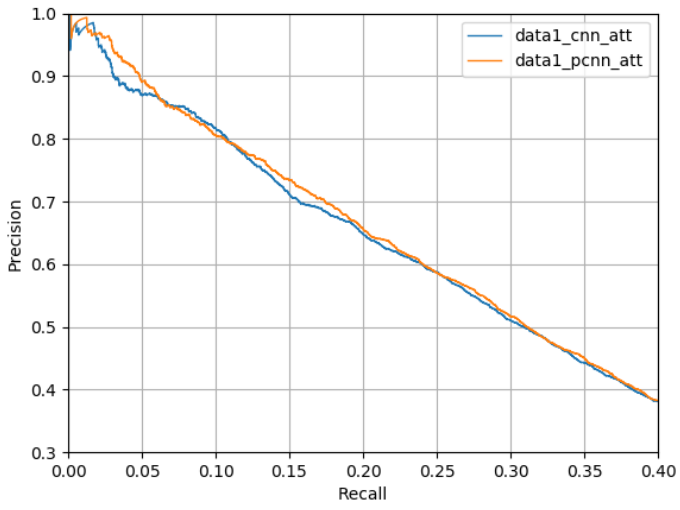

Figure 14: *non-tail dataset*

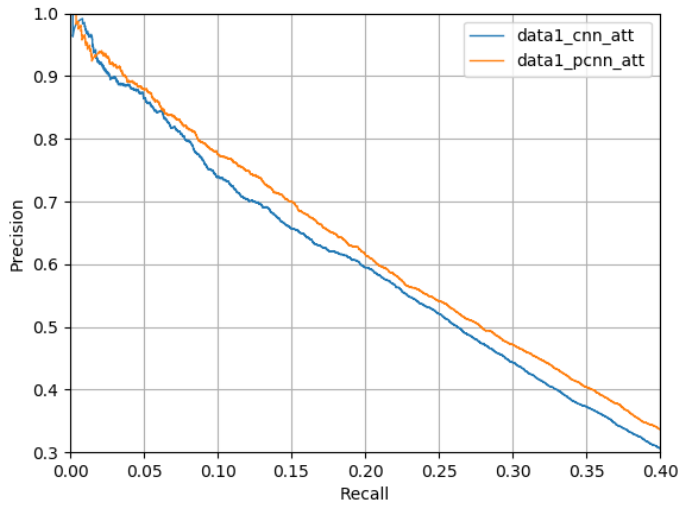

Figure 15: *tail dataset*

Effect of Selectors

- Lin et al. [3] presented the following two conclusions:
  - *Conclusion 1:* ATT selector achieves higher precision over entire range of recall compared to other selectors
  - *Conclusion 2:* AVE selector performs similar to ONE

Contradiction 1

- *tail dataset*
  - *conclusion 1* is true
  - ONE suffers from lack of data
  - ATT uses multiple sentences of a bag to derive features
  - performance is better using ATT selector in case of presence of tail relations
- *non-tail dataset*
  - *conclusion 1* does not hold true
  - ONE and ATT perform equally for non-tail dataset
  - Although only one sentence of a bag is used, there is sufficient data for each relation due to the presence of more number of bags for each relation

Precision-Recall Curve of PCNN_ONE and PCNN_ATT

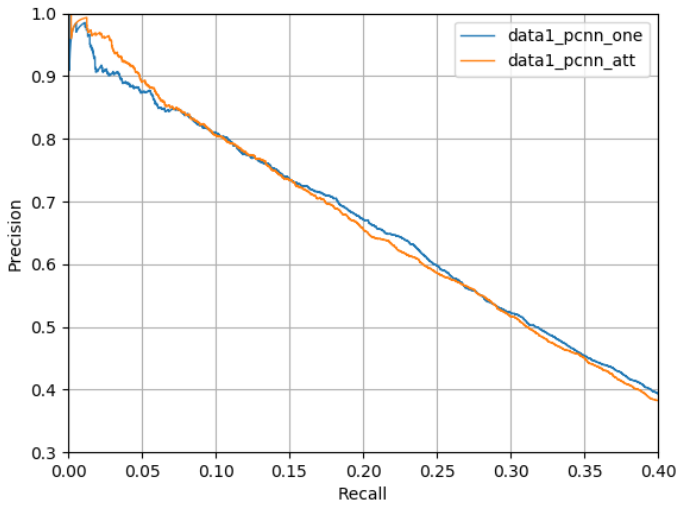

Figure 16: *non-tail dataset*

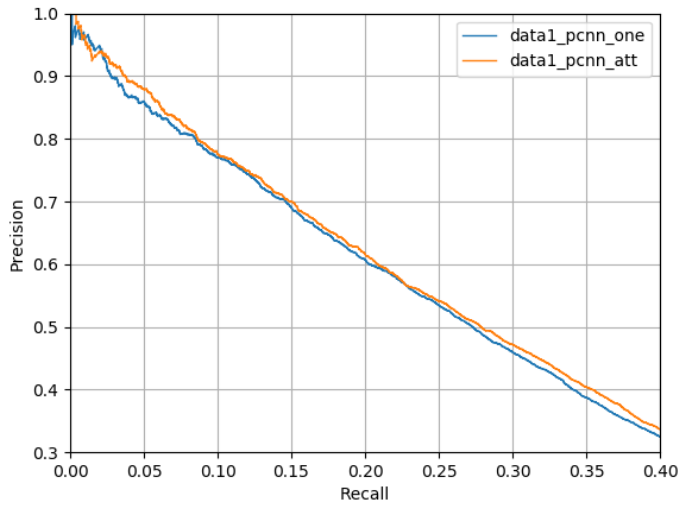

Figure 17: *tail dataset*

Contradiction 2

- ONE performs better when more number of bags are available but not in the case of tail relations
- AVE brings massive noise but has advantage of extracting features from all sentences in case of tail relations
- *tail dataset*
  - *conclusion 2* is true
- *non-tail dataset*
  - *conclusion 2* does not hold true
  - ONE performs better than AVE for non-tail dataset

Precision-Recall Curve of CNN_ONE and CNN_AVE

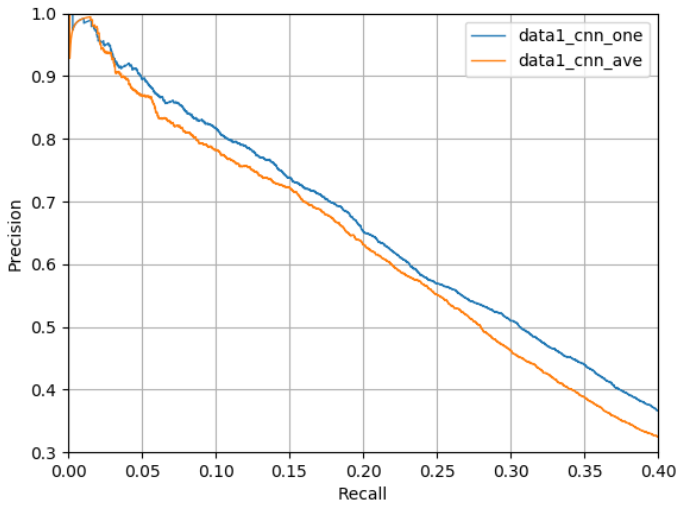

Figure 18: *non-tail dataset*

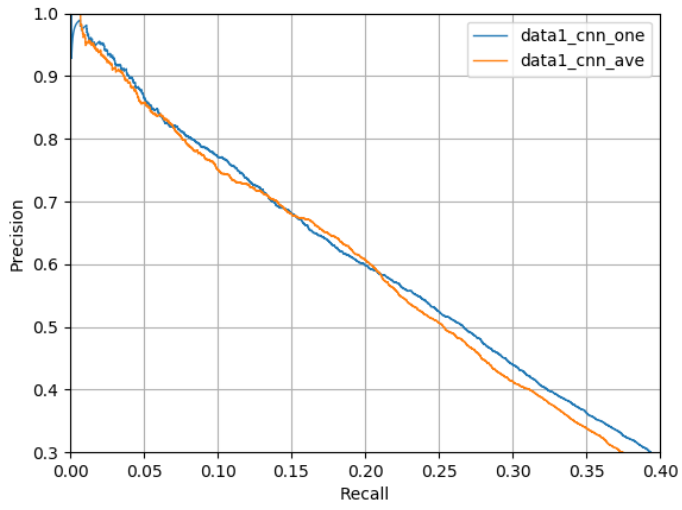

Figure 19: *tail dataset*

Precision-Recall curve of PCNN_ONE and PCNN_AVE

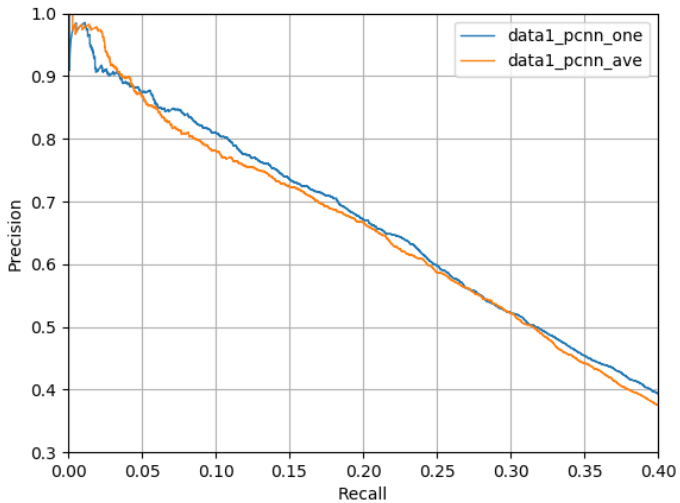

Figure 20: *non-tail dataset*

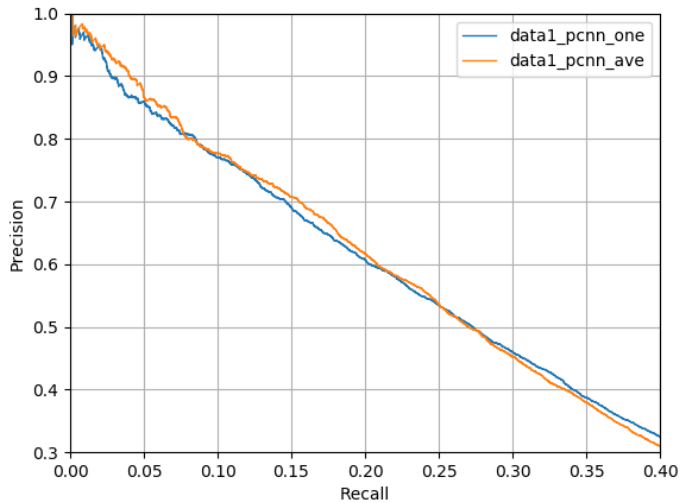

Figure 21: *tail dataset*

Conclusion

- improvements presented in earlier works hold true when tail relations are present
- NYT dataset and *tail dataset* give similar results as both the datasets contain tail relations
- performance of all the models decrease with the presence of tail relations
- models with AVE or ATT selectors give better results for tail relations than ONE
- selectors play a major role than encoders in achieving good performance
- Our dataset can be used for Large-scale RE

References

S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1753–1762.

Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 2124–2133.

Thank you