Large-Scale Relation Extraction Dataset using Wikidata

A Thesis
Submitted in Partial Fulfillment of the Requirements
for the Degree
of

MASTER OF TECHNOLOGY

by

Paka Hema Chandra Kumar
(174101015)

under the guidance of

Dr. Ashish Anand

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

DECLARATION

*This is to declare that the work in this thesis entitled "***Large-Scale Relation Extraction Dataset using Wikidata***", submitted by me to the Indian Institute of Technology Guwahati for the award of the degree of Master of Technology is a bonafide work carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under the supervision of Dr. Ashish Anand. The contents in this thesis, in whole or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.*

Paka Hema Chandra Kumar

May, 2021

Guwahati.

Department of Computer Science & Engineering,
Indian Institute of Technology Guwahati,
Assam-781039, India.

CERTIFICATE

*This is to certify that the work in this thesis entitled "**Large-Scale Relation Extraction Dataset using Wikidata**" is a bonafide work of **Paka Hema Chandra Kumar (Roll No. 174101015)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Dr. Ashish Anand**
Associate Professor,
Department of Computer Science & Engineering,
Indian Institute of Technology Guwahati, Assam.

May, 2021
Guwahati.

Acknowledgement

I would like to take this opportunity to express my gratitude towards my supervisor Dr. Ashish Anand for his continuous support and motivation throughout this work. His immense knowledge and valuable insights has always made me to think and work better. I would like to extend my thanks to each and every member of AMAL research group for supporting me whenever needed.

I am thankful to each and every faculty member of the department of Computer Science and Engineering for their efforts in designing and teaching the courses which helped me to explore and tackle new research problems. I am also thankful to the entire staff of the department of Computer Science and Engineering for helping me to work seamlessly.

I would like to express my gratitude to my batch mates for their valuable suggestions and unmatchable support.

I would like to thank my parents for believing in me and supporting me throughout this journey.

I am extremely grateful to God for providing me this opportunity and the strength to fulfill my dreams.

Paka Hema Chandra Kumar
Indian Institute of Technology Guwahati
May, 2019

Contents

List of Figures

List of Tables

Abstract

Relation extraction is identifying the semantic relation between entities in the text. Supervised methods for relation extraction are hard to extend to large scale relation extraction systems as obtaining training data requires a lot of human effort. Distant supervision methods can be used to build large scale relation extraction systems as it is easier to generate training data. The evaluation of existing distant supervision methods is performed on NYT dataset [2] which has very less number of relations and more than half of the relations have less than 100 training instances. So, we have constructed different datasets with more number of relations using Wikidata as knowledge base and English Wikipedia as text corpus. We examined the performance of distant supervision models proposed by Zeng et al., [1] and Lin et al. [3] on the new datasets. Our experimental results on existing models show that the the improvements presented in their works can be achieved when a dataset with *tail relations*(relations with lesser training instances) is used whereas the same conclusions do not hold for a dataset with adequate training data.

Chapter 1

Introduction

Recent advancements of internet and technology have paved way for the abundance of data in the form of text from various sources like blogs, articles, news, publications, social media etc. Most of the text data available is unstructured. Extracting structured data which can be machine-readable or processable from text would benefit many applications of NLP. Relation extraction (RE) is one of the main sub task in this context which extracts semantic relation between entities in the text. One of the limitations of supervised approaches for relation extraction is the unavailability of annotated or labelled data. Obtaining annotated data is expensive and requires a lot of human effort. This makes extension of supervised methods to large scale RE systems difficult. In recent years, there is a huge development of Knowledge bases (KB) like Wikidata, DBpedia [4] and various private KB's by companies like Google, Facebook etc., due to their applications in various NLP tasks like question answering, web search, knowledge base completion etc. This lead to the introduction of a new paradigm for relation extraction, namely distant supervision (DS). Distant supervision uses knowledge base to generate training data for relation extraction which makes these methods easier to extend to large scale RE systems.

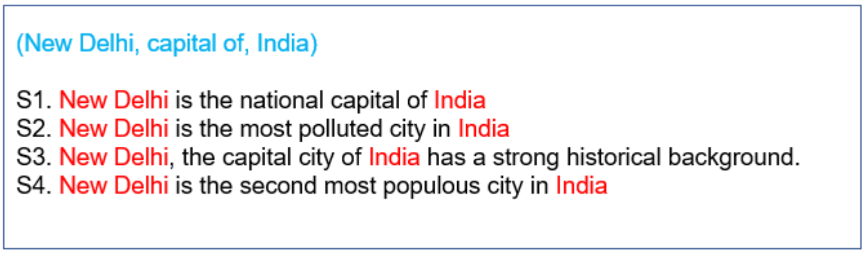
Most of the previous works on distant supervision used NYT dataset by Riedel et al., [2] to train and evaluate their relation extraction systems. NYT dataset consists of 53 relations and more than half of the relations have less than 100 instances. So, we have built a new dataset using an active KB Wikidata and English Wikipedia corpus. Statements(facts) from Wikidata are extracted into triples of the form (head entity, relation, tail entity). We considered 466 relations that exist between head and tail entities of type Person, Organization or Location in the triples. Sentences are labeled with a relation, if both the entities of the triple are present in the sentence. All the sentences which have same entity pair form a bag of sentences. We were able to generate sentences for 324 relations and considered the top 215 relations such that each relation has 10 or more bags of sentences(triples). We constructed different datasets using the top 60, 122, 149 and 215 relations and considered a maximum of 100, 1000 and 10000 bags of sentences in each of the datasets. We have conducted experiments on state of the art RE model by Lin et al., [3] using the above datasets and studied the effect of increasing number of relations and number of bags for a relation.

In our datasets, a *tail relation* refers to a relation with less than 100 bags of sentences. To study the effect of *tail relations* on the existing models, we constructed two datasets with 122 and 215 relations. The former dataset consists of adequate training data for each relation whereas the latter dataset includes the tail relations as well. Our experimental results on existing models show that the the improvements presented in their works can be achieved when a dataset with *tail relations* is used whereas the same conclusions do not hold for a dataset with adequate training data.

1.1 Preliminaries

Distant supervision heuristically aligns the entities in text to knowledge base to obtain annotated data. Knowledge bases typically store data as statements for an entity which can be extracted into triples. The triples are of the form (head entity(e_1), relation(r), tail entity(e_2)) where both the entities e_1 and e_2 are connected though the relation r(e.g.,(*Narendra Modi, Prime minister, India*)). The head and tail entities can interchange depending on the relation between them. The idea of generating training data automatically using knowledge base was initially used in bio-medical domain by Craven and Kumlien [5]. Mintz et al. [6] generalized the idea by proposing the following assumption:

Def. 1 ***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."*



(New Delhi, capital of, India)

S1. New Delhi is the national capital of India
S2. New Delhi is the most polluted city in India
S3. New Delhi, the capital city of India has a strong historical background.
S4. New Delhi is the second most populous city in India

Fig. 1.1 Example for Bag of sentences

The automatic labelling of data according to DS assumption is as follows: for any triple (e_1, r, e_2) that exists in KB, all the sentences in text corpus mentioning the entities e_1 and e_2 are labelled with relation the r. In Figure 1.1, sentences S1 to S4 contain the entities *New Delhi* and *India*, thus they are labelled with the relation *capital of*. However, labelling data with above assumption brings a lot of noise in the data. For example, in Figure 1.1, although the sentences S2 and S4 contain both the entities *New Delhi* and *India*, they do not express the relation *capital of*. The sentences S2 and S4 are examples of noise in the data due to false positive instances. Mutli-instance learning was adopted by Riedel et al. [2], Hoffmann et al. [7] and surdeanu et al. [8] to tackle the noise due to false positive instances. All these earlier works were feature based methods. Zeng et al. [9] [1] and Lin et al. [3] adopted neural network based approaches to automatically obtain features for their relation extraction models.

Def. 2 ***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 1.1 shows a bag of sentences for the triple (*New Delhi, capital of, India*). The terms *bag* and *triples* are used interchangeably to denote a *bag of sentences*. The relation extraction system extracts features from one or more sentences in a bag to train a classifier. The task of relation extraction can be formulated as a multi-class classification problem where for a given entity pair with a bag of sentences, the trained classifier needs to predict the relation label of the bag(bag level evaluation).

1.2 Layout of Report

The structure of rest of the report is as follows: Chapter 2 presents literature survey of Relation Extraction methods. Existing dataset and construction of new datasets are discussed in chapter 3. Chapter 4 and chapter 5 present the experimental results and the conclusion respectively.

Chapter 2

Literature Survey

Relation extraction is one of the extensively studied topic in the field of NLP. The major approaches for the task of RE can be classified as follows:

2.1 Supervised Methods

The task of RE using supervised learning methods is a classification task where a mapping function predicts true if there exist a relation r between entities e1 and e2. The classification can be binary or multi-class classification based on the number of relations that are being considered. The mapping functions which classify the relations can be a perceptron, voted perceptron or SVM etc. Labelled sentences along with set of features like POS tags, dependency parse trees are provided as input to train the models.

Labelled sentences with positive and negative instances for each relation is provided as input to the model. Syntactic features like the entities types, sequence of words between the entities, count of number of words and parse tree of entities and semantic features like dependency parse are extracted from these labelled sentences. Feature vector which consists of these combined features is presented to train the classifier. From these features, relevant features which achieve high performance of the classifier are selected heuristically.

Kambhatla ,2004 [10] used syntax tree features like part of speech tags of both entities and chunk type to train a log linear model. Zhao & Grishman, 2005 [11] used polynomial kernel and GuoDong et al., 2002 [12] used linear kernel with these features to train SVM. Reichartz et al., 2009 [13] used dependency tree features like the distance between the entities and part of speech tags of words present between the entities in the dependency tree etc., as features to train the classifier.

The main disadvantage of supervised methods is their applicability to a specific domain. They are difficult to extend to extract higher order and new relations. They require labelled data which is costly. The methods heavily depend on preprocessed data in the form of parse trees which can have errors and can be propagated, thus hindering the performance of the classifier. Also, supervised methods are not scalable to the input data and computational complexity increases with the size of input data.

2.2 Semi-supervised Methods

These methods follow bootstrapping approach which start with a initial set of seed tuples and iteratively generate more tuples to extract relational tuples. These methods are desirable because of the sparsity of labeled data and creation of large amounts of labelled data for RE is expensive.

2.2.1 DIPRE

DIPRE by Brin, 1998 [14] was one of the first bootstrapping approach which uses yarowskys algorithm [15] to extract relations. It starts with an initial set of seed examples with which the pattern matching classifier is trained to extract new seed examples. The classifier in turn recursively trains with the new seed examples.

2.2.2 Snowball

Snowball by Agichtein & Gravano, 2000 [16] also starts with a seed set of relations similar to DIPRE but with a confidence of 1 for the relations. Unlike DIPRE, Snowball doesn't use a pattern matching system but uses a similarity function which groups tuples that are similar based on features obtained from prefix, middle and suffix parts of entities in the tuples.

2.2.3 KnowItAll

KnowItAll by Etzioni et al., 2005 [17] is a web IE system which uses a set of domain independent extraction patterns to label its training examples and generate relation specific extraction rules. These rules are applied to web pages obtained through search engines to extract relations.

The main disadvantage of DIPRE, Snowball and KnowItAll is they all are relation specific systems. The set of relations which are being considered has to be decided in beforehand. DIPRE uses a pattern matching system and its precision decreases as the number of iterations increases. Snowball on the other hand requires a large amounts of labelled data and it depends on named entity recognizer to identify the entities in tuples.

2.3 Open Information Extraction(Open IE) Methods

Most of the RE systems requires to specify the relations of interest in advance for extraction which works fine for small homogeneous datasets. Traditional RE methods are not scalable to large web scale corpora. Also, the simple linguistic methods proposed fails when applied to the massive corpora which is heterogeneous and contains vast number of relations.

Banko et al. [18] first proposed Textrunner , a Open IE system which discovers the possible set of relations from the text without providing any relations in advance. Naive Bayes classifier is trained based on heuristically obtained training examples and a probability is assigned to the extracted tuples indicating the correctness of the tuple. They also observed a better performance upon using CRF instead of NB classifier. Improvement to the Textrunner was proposed by Wu and Weld [19] in Wikipedia based Open Extractor system which provides training examples using Wikipedia info-boxes instead of heuristic methods. Weld et.al. [20] also used Wikipedia info-boxes in kylin Open IE system. Fader et al. [21] proposed ReVerb, an advanced Open IE system which improves over TextRunner . Statsnowball by Zhu et al. [22], a bootstrapping approach and R2A2 by Etzioni et al. [23] are some of the most notable works.

2.4 Distant Supervision using Feature Based Models

Mintz et al., 2009 [6] first proposed the idea of DS in domain independent setting using Freebase. The assumption which they followed was *"if there exist a relation between two entities, any sentence that contains both the candidate entities, should express the same relation"*. Training data for each relation is generated by finding all the sentences which contain the candidate entities that express that relation in Freebase. Lexical features like part of speech(POS) tags, sequence of words between entities etc., and syntactic features like dependency path between entities in dependency parse tree together form a feature vector which is used to train a multi-class logistic regression classifier. They neglected the noise (sentences that do not express the relation) in the data and trained the classifier on invalid sentences.

Multi Intsance Learning (MIL) by Riedel et al., 2010 [2] argues that the assumption followed in Mintz et al., 2009 [6] is too strong and relaxes the assumption to *"if there exist a relation between two entities, at least one sentence that contains both the candidate entities, should express the same relation"(atleast one).* They used an undirected graphical model which selects valid sentences for training and predicts relations. *MultiR* by Hoffmann et al., 2011 [7] uses similar graphical model of MIL but addresses the drawback of MIL which doesnt allow overlapping relations i.e r(e1,e2) & q(e1,e2) cannot exist where r, q are relations and e1,e2 are candidate entities. Experimental results showed a better precision and recall for MultiR than MIL.

Multi-Instance Multi-Label (MIML) by Surdeanu et al., 2012 [8] proposed a similar model like MultiR which used Expectation Maximization (EM) for trainng. *MIML-semi* by Min et al., 2013 [24] addresses the problem of false negatives generated due to the incompleteness of KBs.

The major disadvantage of these conventional feature based methods is that features such as Part of speech tags etc., are obtained using NLP tools. The performance of these methods heavily depend on the quality of features obtained. The errors that are generated using preexisting tools propagate in the feature based models. Thus deep learning methods which extract relevant features automatically are used to address the limitations of feature based models.

2.5 Distant Supervision using Deep Learning Models

Deep Learning Models uses Convolutional Neural Networks to capture the semantics of the sentence. Liu et al., 2013 [25] first used CNN's to automatically extract features at sentence level. The word embedding used in this model failed as they were assigned randomly to each class.

CNN model with max-pooling by Zeng et al., 2014 [9] also used CNN, but exploited a better way of word embedding which was pre-trained on large unlabelled corpus. This paper first used positional embedding which was adapted in all the subsequent deep learning models. They have used max-pooling layer at the output which would make the output independent of size of sentence.

CNN with multi-sized window kernels by Nguyen and Grishman, 2015 [26] uses the same approach as Zeng et al., 2014 [9] uses a convolutional kernels of different window sizes to capture n-gram features. They used word embeddings which was pretrained using *word2vec* Mikolov et al., 2013 [27].

Piecewise Convolutional Neural Networks by Zeng et al., 2015 [1] explored multi-instance learning with similar model as Zeng et al., 2014 [9] and Nguyen and Grishman, 2015 [26]. They claimed that the max-pooling layer at the output reduces the hidden layers and not sufficient to capture the structural information in the sentences. So, they used max-pooling on different segments of sentence. The major drawback was that the model is trained assuming *atleast one assumption* (atleast one sentence expresses the relation) Riedel et al., 2010 [2] and uses the most probable sentence due to which it looses large amount of useful data from other sentences.

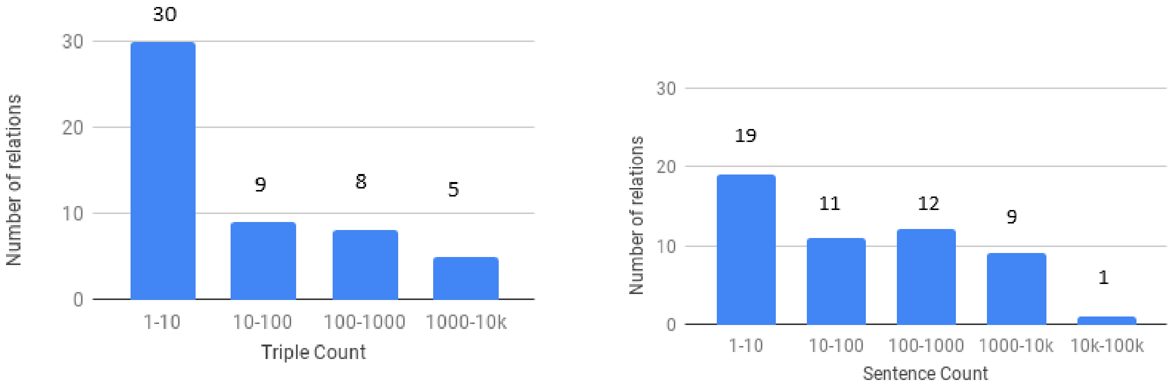
This drawback was addresses by Lin et al., 2016 [3] by *Selective Attention over Instances.* They tackled the problem by assigning weights to all the candidate sentences. The model identifies the candidate sentences which expresses the relation and gives a larger weight and lesser weight to noisy sentences. This paper also addresses the problem of multi-label (same entity pairs with multiple relations) of RE.

Chapter 3

Datasets

3.1 NYT Dataset

Most of the previous works used NYT dataset developed by Riedel et al. [2] for evaluation. The dataset was generated using New York Times corpus. Freebase [28] was used as the knowledge base. Articles from 2005-06 corpus are used for generating training data and 2007 corpus is used for generating testing data. NYT contains 52 relations and a special "NA" relation for negative training and testing sentences. The training data contains 522,611 sentences (385664 sentences for "NA" relation) and 18,252 relation facts (triples) and the testing data contains 172,448 sentences (94917 sentences for "NA" relation) and 1950 relation facts (triples) [3].



(a) Relations vs Triple count

(b) Relations vs Sentence count

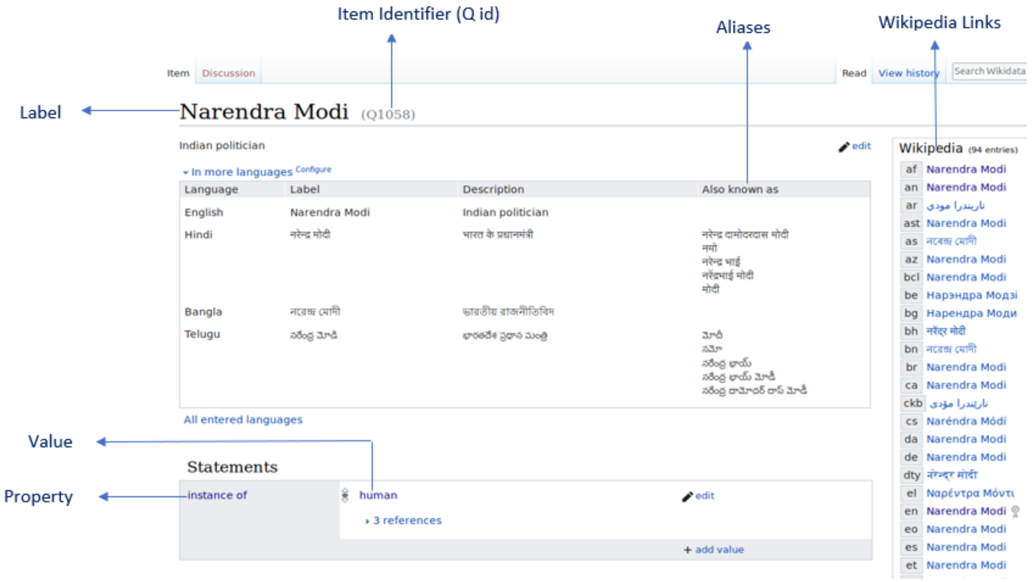
Fig. 3.1 Number of relations in different ranges of triple count and sentence count in the NYT training dataset

Figure 3.1(a) shows the distribution of relations in different ranges of triple count (triples which have sentences) in the NYT training dataset. Figure 3.1(b) shows the distribution of relations in different ranges of sentence count in the NYT training dataset. The NYT testing dataset consists of sentences for only 32 relations.

3.2 Construction of New Dataset

NYT dataset was constructed using Freebase as Knowledge base which was discontinued in 2016. So, we use Wikidata as Knowledge base which is an active KB and also available under public domain license to obtain more number of relations. We use Wikipedia dump in English language as text corpus to generate sentences for our dataset.

3.2.1 Extraction of Triples from Wikidata



Item Discussion Read View history Search Wikidata

Narendra Modi (Q1058)

Indian politician ✎ edit

▾ In more languages Configure

Language	Label	Description	Also known as
English	Narendra Modi	Indian politician	
Hindi	नरेन्द्र मोदी	भारत के प्रधानमंत्री	नरेन्द्र दामोदरदास मोदी नमो नरेन्द्र भाई नरेंद्रभाई मोदी मोदी
Bangla	নরেন্দ্র মোদী	ভারতীয় রাজনীতিবিদ	
Telugu	నరేంద్ర మోడీ	భారతదేశ ప్రధాన మంత్రి	మోడీ నమో నరేంద్ర భాయ్ నరేంద్ర భాయ్ మోడీ నరేంద్ర దామోదర్ దాస్ మోడీ

All entered languages

Statements

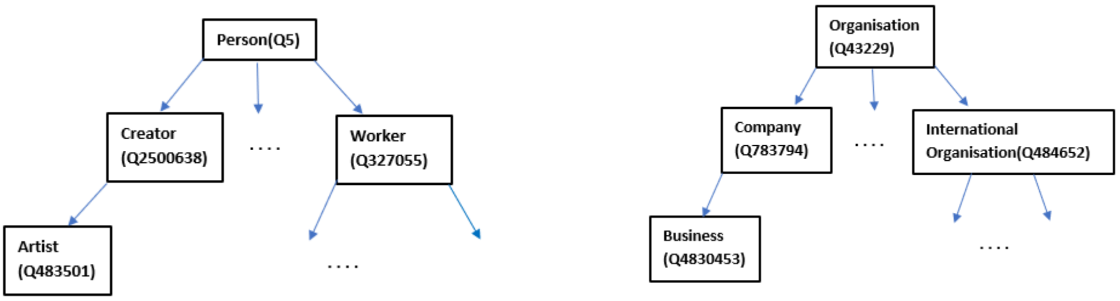
instance of | human ✎ edit
▸ 3 references
+ add value

Wikipedia (94 entries)

af	Narendra Modi
an	Narendra Modi
ar	ناريندرا مودي
ast	Narendra Modi
as	নৰেন্দ্ৰ মোদী
az	Narendra Modi
bcl	Narendra Modi
be	Нарэндра Модзі
bg	Нарендра Моди
bh	नरेंद्र मोदी
bn	নরেন্দ্র মোদী
br	Narendra Modi
ca	Narendra Modi
ckb	نارێندرا مۆدی
cs	Naréndra Módí
da	Narendra Modi
de	Narendra Modi
dty	नरेन्द्र मोदी
el	Ναρέντρα Μόντι
en	Narendra Modi
eo	Narendra Modi
es	Narendra Modi
et	Narendra Modi

Fig. 3.2 Wikidata page of Narendra Modi

Wikidata contains information about **items** which can be topics, objects and various other things. Each item in Wikidata is represented with a unique id starting with Q called **Qid**. For e.g., in Figure 3.2 *Narendra Modi* is an item in Wikidata and *Q1058* is its Qid. An item in Wikidata contains labels and descriptions in different languages. Information about an item is presented as **statements**. Each statement is a key-value pair where key is a **property**(starts with P and referred as Pid) in Wikidata. For e.g., a sentence “New Delhi is the capital of India” is stored as statement of item *New Delhi (Q987)* where the key is *capital of (P36)* and the value is *India (Q668).* In Figure 3.2 , a statement with key *instance of* and value *human* about item *Narendra Modi* is shown.



(a) Person(human(Q5)) (b) Organization(Q43229)

Fig. 3.3 Hierarchy of classes with *subclass of(P279)* property

Identify Items of Type Person, Organization and Location

As most of the relations are expressed among entities of type Person, Organization and Location, we restrict our work to relations that exist between them. Items in Wikidata are hierarchically arranged with *“sub-class of” (P279)* property as shown in Figure 3.3. Sub-classes of Person(human(Q5)), Organization(Q43229) and Location(Q17334923) are obtained by traversing them along the hierarchy. We observed that there are 643 sub-classes of Person, 18417 sub-classes of Organization and 11114 sub-classes of Location. The obtained sub-classes are also Wikidata items. Items are classified into one of the above three classes if they are an *“instance of”(P31)* any of the class or sub-classes of Person, Organization and Location. For e.g., in Figure 3.2 *Narendra Modi* is an *instance of human*, a sub-class of Person, so the entity is of type Person.

We extract triples of the form (item, property, value) from Wikidata where item and value are of type Person, Organization or Location. We consider triples in which value is also an item in Wikidata. We refer item and value as entities and property as the relation between them.

Statistics of Wikidata

Wikidata has more than 38 million items and 5147 properties. We extracted more than 15 million triples and there are 466 different properties(relations) in them. Table 3.1 shows the top 5 relations with more triple count. Figure 3.4 shows the distribution of relations in different ranges of triple count. One-third of relations has less than 10 tuples.

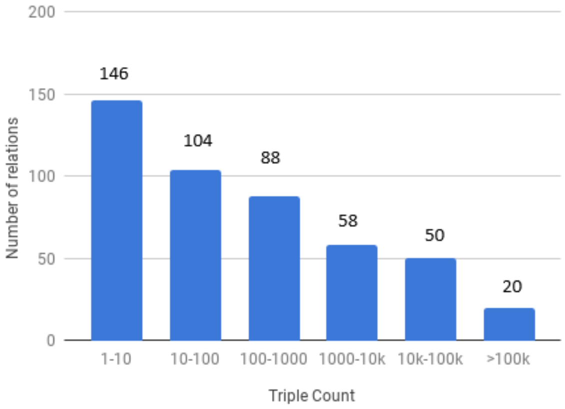


Fig. 3.4 Number of relations in different ranges of triple count (Triples obtained from Wikidata)

Relation	**Triple Count**
country of citizenship	2806203
country	2409792
place of birth	1819921
located in the administrative territorial entity	1372697
member of sports team	1196525

Table 3.1 Top five relations(based on number of triples) in Wikidata and their triple count

3.2.2 Extraction of Sentences from Wikipedia

We considered English language Wikipedia dump in our experiments. It contains 94,635,369 sentences. Figure 3.5 shows the pipeline for extraction of sentences through distant supervision paradigm. Entities in each sentence are identified through the hyperlinks present in them. Items in Wikidata contain Wikipedia link, if there exists a Wikipedia page for that item. Figure 3.2 shows links of Wikipedia pages in different languages. Wikidata id (Qid) of each entity in the

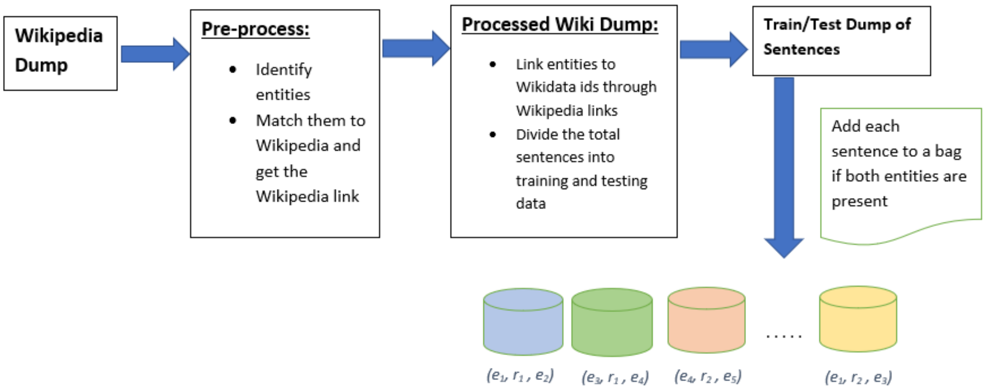
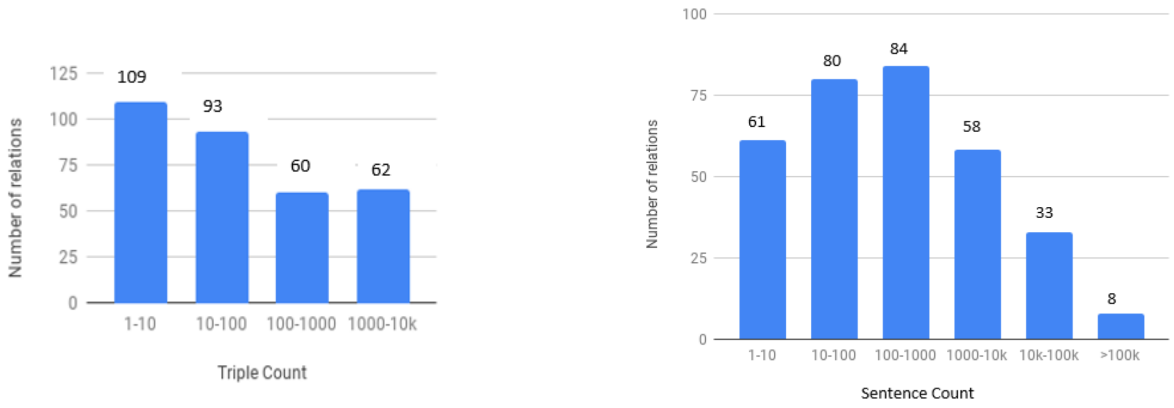

Wikipedia Dump
Pre-process:
Identify entities
Match them to Wikipedia and get the Wikipedia link
Processed Wiki Dump:
Link entities to Wikidata ids through Wikipedia links
Divide the total sentences into training and testing data
Train/Test Dump of Sentences
Add each sentence to a bag if both entities are present
(e_1, r_1, e_2)
(e_3, r_1, e_4)
(e_4, r_2, e_5)
.....
(e_1, r_2, e_3)

Fig. 3.5 Pipeline for creation of dataset

sentence are obtained using Wikipedia links. There are 22,818,790 sentences with more than two entities present in them. We considered these sentences for generating training and testing datasets. We randomly divided two-third sentences into training and the remaining into testing. According to distant supervision assumption, a sentence expresses a relation if both the head and tail entities are present in it. We consider every possible pair of entities in the sentence and check if there exists any relation between them according to Wikidata(using triples generated from Wikidata). Sentences are labelled with head entity, tail entity and relation. We were able to generate sentences for 804,680 bags(triples) i.e, 5% of total triples generated from Wikidata and there are 324 different relations in them .There are no bag of sentences for 122 relations. Figure 3.6(a) shows the distribution of relations in different ranges of triple count (triples which have sentences). Figure 3.6(b) shows the distribution of relations in different ranges of sentence count.

3.2.3 Construction of Datasets

We have divided the sentences into training and testing sentences. We have created four different datasets based on the number of bags(triples) that are present for a relation in training dataset. In each dataset we have considered a subset of relations whose bags(triple) count is greater than a threshold value. Table 3.2 gives the threshold values and the corresponding number of relations for each dataset in training. For each dataset, a corresponding test dataset



(a) Relations vs Triple count

(b) Relations vs Sentence count

Fig. 3.6 Number of relations in different ranges of triple count and sentence count

is constructed with the same number of relations using test sentences. For each relation, maximum of 100 bags(triples) are considered in the test dataset. We also avoided overlapping of bags(triples) in train and test datasets (triples considered in test dataset are not considered in train dataset).

Dataset	**Minimum bags(threshold)**	**Number of relations**	**Test bags**
dataset1	1000	60	6000
dataset2	100	122	11814
dataset3	50	149	13013
dataset4	10	215	14045

Table 3.2 Threshold values (based on training data), number of relations and number of test bags in each dataset

Negative training/testing data

Similar to NYT dataset, we consider sentences in which the entities does not have any relation according the knowledge base as negative sentences and assign them a special relation “NA”. We considered 700,000 different entity pairs which does not have any relation according to Wikidata and added their bag of sentences in training data. Similarly, we considered 100,000 entity pairs in testing data. These bags are assigned “NA” relation.

Chapter 4

Experiments and Results

In our experiments, we observed the performance of existing deep learning methods ([1] [3]) on our dataset. In this chapter, we discuss the existing deep learning methods for distant supervision followed by the results and analysis.

4.1 Framework of Deep Learning Models

4.1.1 Vector Representation of Input

Word Embeddings

Word Embeddings are the representations of words in a d_k dimensional vector(real valued) which can capture information about words. Word embeddings are learned on an unlabeled text according to the co-occurrence of words in that text. Consider a sentence s containing m words, $s = \mathrm{s}_1,\mathrm{s}_2\ldots,\mathrm{s}_m$ where each word s_i is a k dimensional column vector. We use Skip-gram model by Mikolov et al. [27] for training word embeddings on Wikipedia dataset. Word tokens in the input sentence are transformed into vectors by looking up word embeddings.

Position Embeddings

Position Embeddings or position features were first used by Zeng et al.(2014) [9] which represents relative distance of a word from head and tail entities. For e.g., Figure 4.2 shows relative

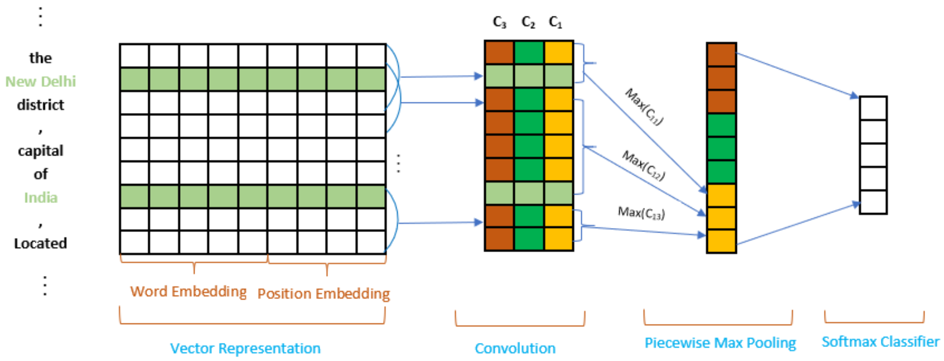


Fig. 4.1 Architecture of PCNN model (adapted from Zeng et al. [1])

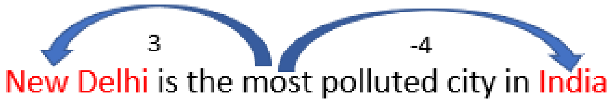


Fig. 4.2 Example of relative distances used in position embeddings

distance from *most* to *New Delhi* (head entity) and *India* (tail entity). Position embedding matrices ($\mathbf{PF}_1$ and $\mathbf{PF}_2$) are randomly initialized and learned during training [9]. Real valued vectors for relative distances are obtained by looking up the position embedding matrices.

The combined vector representation of a sentence is a matrix $\mathbf{S} \in \mathbb{R}^{m \times d}$, where m is the number of words in sentence and each word $s_i \in \mathbb{R}^d$ $(d = d_k + 2 \times d_p)$. In Figure 4.1 the size of word embedding d_k is 5 and the size of position embedding d_p is 2.

4.1.2 Encoders

Convolution

Convolution Neural Networks are known for their usage in the computer vision domain. They capture the local features from images by applying filters in a sliding window fashion. The obtained local features are merged using max-pooling operation to obtain a fixed representation of features of the input image. Similarly, CNNs are used to extract features or n-grams from sentences by applying filters over the vector representation of sentences.

Multiple filters are used to capture different features in a sentence. Consider a convolution layer $\mathbf{W}$ consisting of n filters (vector of weights), $\mathbf{W} = \{\mathrm{w}_1, \mathrm{w}_2 \ldots, \mathrm{w}_n\}$. Each filter $\mathbf{w} \in \mathbb{R}^{w \times d}$ where w is the length of each filter and d is the length of combined input representation of word and postion embeddings for each word. In Figure 4.1, w =3 and d=9. The convolution operation produces output matrix $\mathbf{C} = \{c_1, c_2, \ldots, c_n\}$ and $\mathbf{C} \in \mathbb{R}^{n \times (m+w-1)}$ where n is total number of filters, m is length of sentence and w is length of filter. Each element c_{ij} of C is obtained by convolution operation of w-gram of sentence and filter w_i:

$$c_{ij} = w_i q_{j-w+1:j}$$

where i ranges on number of filters and j ranges from 1 to *m+w-1* and $q_{j-w+1:j}$ is the w-gram of the input sentence. In Figure 4.1, three filters are considered. So, we have c_1, c_2, c_3 as output after convolution.

Piece-wise Max-pooling

Max-pooling technique is used to reduce the convolution output of sentence and make it independent of sentence length to apply subsequent layers. However, single max-pooling reduces the size of hidden layers rapidly and not sufficient to capture features for Relation Extraction [3]. Also, single max-pooling is insufficient to capture structural information of entities in the sentence [3]. So, max-pooling is applied in three segments: before head entity, between head and tail entity and after tail entity of the sentence called piece-wise max-pooling. The output after piece-wise max-pooling is:

$$p_{ij} = max(c_{ij})$$

where i ranges on number of filters and $1 \leq \mathrm{j} \leq 3$. The piece-wise max-pooling outputs are concatenated into $p_i = \{p_{i1}, p_{i2}, p_{i3}\}$ where i ranges on number of filters and $p_i \in \mathbb{R}^{3n}$. A non-linear function such as hyperbolic tangent is then applied on the piece-wise max-pooling output. The confidence scores for each relation are obtained by applying softmax classifier. We denote encoder with single max-pooling as CNN and piece-wise max-pooling as PCNN.

4.1.3 Selectors

Selectors are used to alleviate the wrong labelling problem. The following methods were used to select the sentences from bags:

ONE

The objective function used by Zeng et al. [1] to train the model by selecting a single instance at the bag level is defined using cross-entropy as follows:

$$J(\theta) = \sum_{i=1}^{T} logp(y_i|m_i^{j^*}; \theta)$$

where $j^* = \arg\max_j p(y_i|m_i^j; \theta)$, i ranges on T training bags of form (M_i, y_i), y_i is the relation label of bag, m_i^j denotes a sentence in bag, θ is the set of model parameters which are word and position embedding matrices, Convolution matrix (filters) and transformation matrix for the softmax classifier. Stochastic Gradient Descent (SGD) is used to maximize the objective function.

AVE

Let $\mathbf{x}_i$ denotes the output after encoding the sentence. A vector $\mathbf{s}$ which is the combined representation of all sentences in a bag is the weighted sum of individual sentences.

$$\mathbf{s} = \sum_i \alpha_i x_i$$

α measures the relatedness of each sentence in a bag and the relation of the bag. In this method, each sentence is given equal weight and α_i is considered as $1/|M|$ where $|M|$ is size of the bag.

ATT

Selective attention on instances proposed by Lin et al., [3] uses a scoring function e_i to get the relatedness of sentence and the relation of the bag.

$$\alpha_i = \frac{exp(e_i)}{\sum_k exp(e_k)}$$

where $e_i = x_i \mathbf{A} \mathbf{r}_i$, $\mathbf{A}$ is a diagnol matrix and $\mathbf{r}_i$ is relation embedding.

In our further discussion, we represent a model with the encoder and selector used in it. For e.g, PCNN_ATT denotes a model with PCNN as encoder and ATT as selector.

4.2 Evaluation Metric

In our experiments, we evaluate at bag(entity pair) level i.e., the model should predict the relation label of a bag rather than each sentence in a bag. The predicted relation between entities of a bag is compared against the relation in triples obtained from Wikidata. We compare the precision-recall curve for each model. The precision-recall 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 $= \frac{Number\ of\ correctly\ extracted\ relations}{Actual\ number\ of\ relations}$
- F-measure F1 $= \frac{2PR}{P+R}$

4.3 Effect of Number of Bags for a Relation

The datasets created are based on minimum number of bags that are present for a particular relation. The difference in number of bags between relations is very high. So, different upper bounds (100,1000,10000) are put on the number of bags that are considered for a relation in the training data. The same test data corresponding to each dataset as described in section 3.2.3 is used in all settings to check the performance of state of the art model PCNN_ATT by Lin et al., using AUC score of precision-recall curve.

Maximum bags	**100**		**1000**		**10000**	
Dataset	**train bags**	**AUC**	**train bags**	**AUC**	**train bags**	**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 4.1 PCNN_ATT model: AUC scores of datasets with different upper bounds on number of bags for a relation

Table 4.1 shows that the scores of all the datasets are high when a maximum of 1000 bags are considered for each relation compared to 100 or 10000 bags. The scores are low when maximum bags of 100 are considered due to very less training data for each relation. The AUC score of *dataset1* with maximum 100 bags is 0.2036 due to less training data. In case of maximum bags of 10000, the model performs better only for a few relations as it is skewed towards the relations which has more number of bags. The AUC score of *dataset1* with maximum of 10000 bags is less compared to 1000 bags.

The AUC score decreases with the increase of relations as the the newly added relations has less training bags compared to the existing relations. The decrease in score depends on the number of bags available for the newly added relations. However, in the case of maximum bags of 1000 for a relation, the decrease in AUC score is less compared to other cases. In our further experiments we consider a maximum of 1000 bags for each relation in all the datasets.

4.4 Comparison of Deep Learning Models

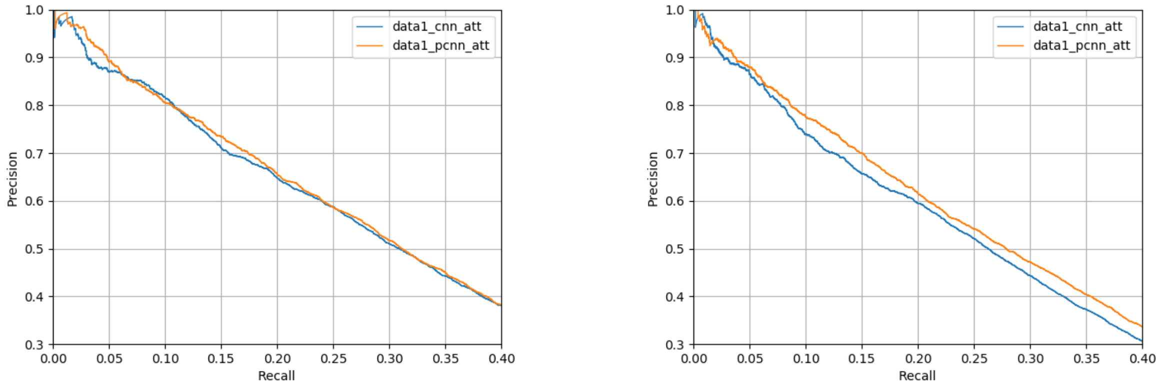
We present our results and analysis on *dataset2* and *dataset4.* Each relation in *dataset2* contains a minimum of 100 bags and a maximum of 1000 bags. Similarly, *dataset4* consists of a minimum of 10 bags and a maximum of 1000 bags. We refer *dataset2* as *non-tail dataset* and *dataset4* as *tail dataset.* Table 4.2 shows the AUC scores for *non-tail dataset*, *tail dataset* and NYT dataset in different settings using CNN and PCNN as encoders and ONE, AVE and ATT methods as selectors.

Dataset	**non-tail(122)**		**tail(215)**		**NYT**	
Encoder/Selector	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 4.2 Comparison of AUC scores of datasets with different encoders and selectors

4.4.1 Effect of using Encoders with Piece-wise Max-pooling

Table 4.2 shows that 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. ATT selector gives more weight to sentences that are more likely to express the relation and thereby alleviates the problem of noisy data. Structural information of the sentence obtained using PCNN as encoder does not achieve much improvement in performance when sufficient training data is available for each relation and the problem due to noisy data is alleviated. Figure 4.3(b) shows that PCNN_ATT performs better than CNN_ATT for *tail dataset*. Structural information obtained using piecewise max-pooling is particularly useful when the available training data is less.



(a) *non-tail dataset*

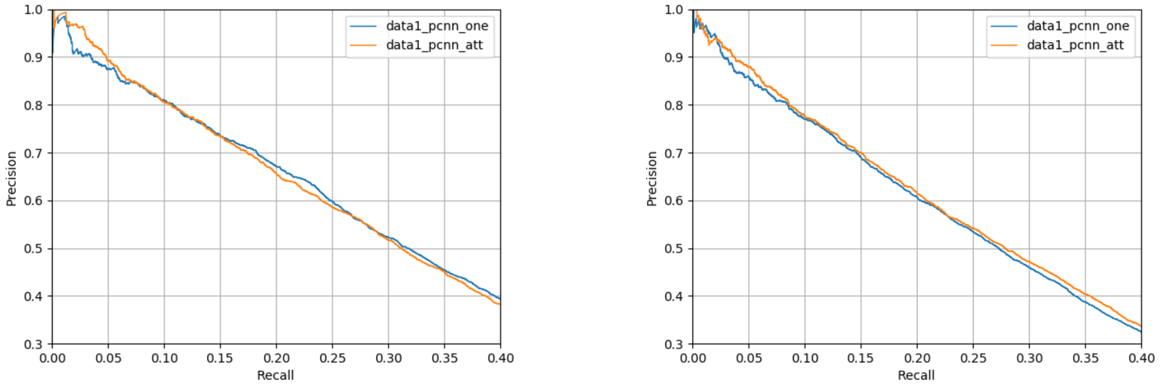
(b) *tail dataset*

Fig. 4.3 Precision-Recall curve of CNN_ATT and PCNN_ATT

4.4.2 Effect of Selectors

Lin et al. [3] presented the following two conclusions from their experiments on NYT dataset:
(1) ATT selector achieves higher precision over entire range of recall compared to other selectors.
(2) AVE selector performs similar to ONE.
Experimental results on our datasets contradicts the above two conclusions.

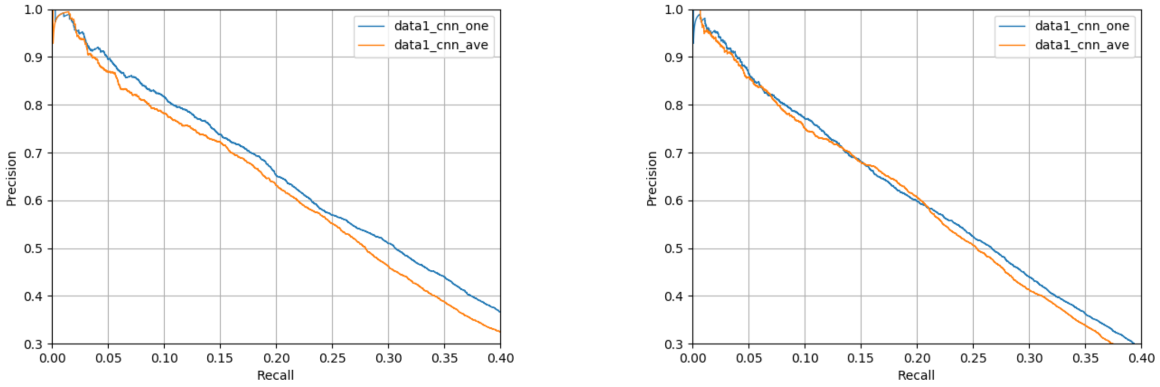
Contradicting the first conclusion, ONE and ATT perform equally for *non-tail dataset* as shown in Figure 4.4(a). Although only one sentence of a bag is used in training in case of ONE selector, there is sufficient data for each relation in *non-tail dataset* as more number of bags are present for each relation. Thus, ONE achieves similar performance as ATT. However, their conclusion holds true for *tail dataset.* From Figure 4.4(b), we can see that the performance of ATT is slightly better than ONE over entire range of recall for *tail dataset* as the available data



(a) *non-tail dataset* (b) *tail dataset*

Fig. 4.4 Precision-Recall curve of PCNN_ONE and PCNN_ATT

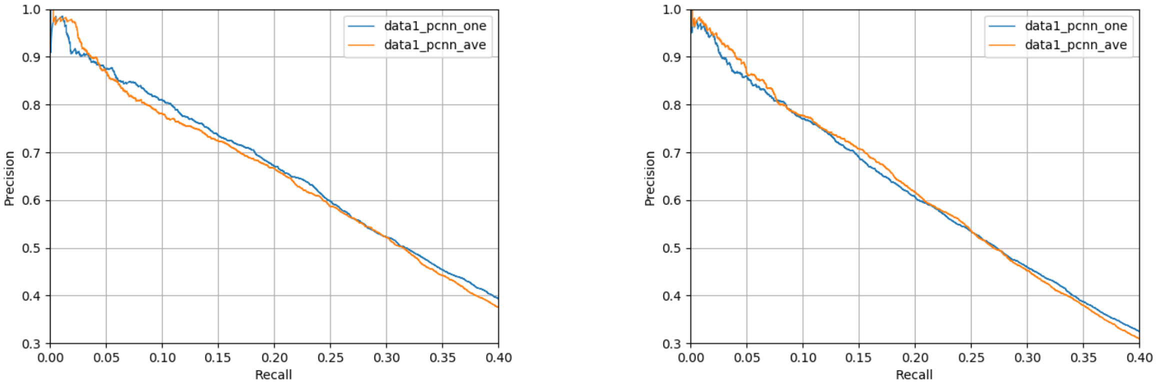
for each relation decreases due to the tail relations in *tail dataset*. The performance is better using ATT selector in case of presence of tail relations as it uses multiple sentences of a bag to derive features.



(a) *non-tail dataset* (b) *tail dataset*

Fig. 4.5 Precision-Recall curve of CNN_ONE and CNN_AVE

Contradicting the second conclusion, ONE performs better than AVE for *non-tail dataset* as shown in Figures 4.5(a) and 4.6(a). The performance of AVE decreases as it brings massive noise by assigning equal weight to all the sentences of a bag. However, Figures 4.5(b) and 4.6(b) show that their performance is similar for *tail dataset*. ONE performs better when more number of bags are available but not in case of tail relations. AVE does not perform well with the former case due to noise but performs better than ONE in case of tail relations. As *tail* *dataset* contains tail relations along with relations with more number of bags, the performance of ONE and AVE is similar.



(a) *non-tail dataset*

(b) *tail dataset*

Fig. 4.6 Precision-Recall curve of PCNN_ONE and PCNN_AVE

4.4.3 Conclusion

- Experiments on NYT dataset and *tail dataset* give similar results as both the datasets contain tail relations.
- The performance of all the models decreases with the presence of *tail relations* in the dataset.
- The models with AVE or ATT selectors give better results for tail relations than the models with selector ONE.
- Using better selectors to reduce noise achieves a better performance than knowing the structural information of the sentence using piece-wise max-pooling i.e., selectors play a major role than encoders in achieving good performance.

Chapter 5

Conclusion

Distant supervision can be used to build large-scale relation extraction systems as obtaining training data is easier. The evaluation of existing distant supervision methods was performed on NYT dataset [2] which has less number of relations and more than half of them have less than 100 training instances. So, we created a new large scale relation extraction dataset using Wikidata as knowledge base and Wikipedia as text corpus. We have reexamined the performance of the existing deep learning models using our large-scale RE dataset. Our experimental results show that conclusions presented in earlier works are true for a dataset which contains relations with lesser training data but does not hold for a dataset with adequate training data for each relation.

References

[1] 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.

[2] 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.

[3] 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.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web.* Springer, 2007, pp. 722–735.

[5] M. Craven, J. Kumlien *et al.*, "Constructing biological knowledge bases by extracting information from text sources." in *ISMB*, vol. 1999, 1999, pp. 77–86.

[6] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2.* Association for Computational Linguistics, 2009, pp. 1003–1011.

[7] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 2011, pp. 541–550.

[8] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning.* Association for Computational Linguistics, 2012, pp. 455–465.

[9] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 2335–2344.

[10] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions.* Association for Computational Linguistics, 2004, p. 22.

[11] S. Zhao and R. Grishman, “Extracting relations with integrated information using kernel methods,” in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 419–426.

[12] Z. GuoDong, S. Jian, Z. Jie, and Z. Min, “Exploring various knowledge in relation extraction,” in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 427–434.

[13] F. Reichartz, H. Korte, and G. Paass, “Dependency tree kernels for relation extraction from natural language text,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 270–285.

[14] S. Brin, “Extracting patterns and relations from the world wide web,” in *International Workshop on The World Wide Web and Databases*. Springer, 1998, pp. 172–183.

[15] S. Abney, “Understanding the yarowsky algorithm,” *Computational Linguistics*, vol. 30, no. 3, pp. 365–395, 2004.

[16] E. Agichtein and L. Gravano, “Snowball: Extracting relations from large plain-text collections,” in *Proceedings of the fifth ACM conference on Digital libraries*. ACM, 2000, pp. 85–94.

[17] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Unsupervised named-entity extraction from the web: An experimental study,” *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[18] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, “Open information extraction from the web.” in *IJCAI*, vol. 7, 2007, pp. 2670–2676.

[19] F. Wu and D. S. Weld, “Open information extraction using wikipedia,” in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 118–127.

[20] D. S. Weld, R. Hoffmann, and F. Wu, “Using wikipedia to bootstrap open information extraction,” *SIGMOD record*, vol. 37, no. 4, pp. 62–68, 2008.

[21] A. Fader, S. Soderland, and O. Etzioni, “Identifying relations for open information extraction,” in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

[22] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, “Statsnowball: a statistical approach to extracting entity relationships,” in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 101–110.

[23] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, “Open information extraction: The second generation.” in *IJCAI*, vol. 11, 2011, pp. 3–10.

[24] B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek, “Distant supervision for relation extraction with an incomplete knowledge base,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 777–782.

[25] C. Liu, W. Sun, W. Chao, and W. Che, “Convolution neural network for relation extraction,” in *International Conference on Advanced Data Mining and Applications.* Springer, 2013, pp. 231–242.

[26] T. H. Nguyen and R. Grishman, “Relation extraction: Perspective from convolutional neural networks,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 39–48.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[28] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data.* AcM, 2008, pp. 1247–1250.