Hybrid Deep Learning Graphical Model for Relation Extraction

Zachariah Zhang
New York University
zz1409@nyu.edu

Lizi Chen
New York University
lc3397@nyu.edu

***Abstract*— The relation extraction task is to predict the relationship of two entities give a text document. This task is very important for a variety of natural language processing tasks such as information retrieval or question answering. Deep learning has seen a huge level of interest in this area due to its ability to model very complex relations in language. Most of these models focus on predicting a relationship given two entities. However, in practice we may have many entities and want to predict multiple relationships. We introduce a framework for using a graphical model in combination with deep learning to learn a model that is able to leverage the power of deep learning as well as model the joint distribution over relationships in a document. We evaluate our model on a dataset of Wikipedia biographies and show that it outperforms models just using deep learning.**

I. Introduction

Relation extraction has historically been a very important task in NLP. Extracting the relationships between different entities in a body of text is a very important subtask for machines to understand natural language. This has important applications in question answering, searching, unstructured data labeling and many other areas. However, this has historically been a difficult task as natural language data is typically unstructured and has many ambiguities.

Rule-based system can work with high precision in narrow applications but tend to have low recall and are generally inflexible. More recently, supervised learning has become a popular approach to this problem. Using this approach we learn a model that is able to predict a probability distribution over relations given two entities and their context.

Deep learning has seen an explosion in popularity for many different NLP tasks because of its ability to leverage large amounts of data to build very complex and robust models for natural language. It has had a similar impact on the relation extraction task, setting many state of the art baselines. Most of these deep learning models only consider the task of predicting a single relationship from two entities. However, in practice we would often like to predict the relationship of the subject with many entities. Predicting each of these relations independently fails to capture the global context of a document. For example if a biography is about a musician and we know that the "genre" relationship is present we would expect other relationships such as "instrument", or "debut-date" relationships as well.

One way in which we can model these constraints is to use a graphical model in conjunction with a deep learning model. Graphical models can efficiently represent distributions over many random variables. We experiment with several variants of models in this paper to try and model the joint distribution of relationships. This approach can be very effective when training on documents that have strong dependencies between relationships. For this reason we chose the wikipedia biography dataset. The dataset comprises shortened wikipedia articles along with the wikipedia info box that lists different entities and relationships such as spouse, date-of-birth, political party, ect.

In this paper we show that this approach outperforms deep learning along on this task. In the following sections we will provide background of the relation extraction task, introduce our model, as well as provide analysis of the learned model.

II. Related Work

A. Markov Random Fields

Graphical models have been used extensively in natural language processing for tasks such as part of speech tagging and name entity recognition. The purpose of using graphical models is to efficiently encode the conditional indecencies of a group of random variables. Markov random fields define a probability distribution with the following equations.

$$P(x) = \frac{1}{Z(\theta)} \prod_{c \in C} \phi_c(x_c) \tag{1}$$

$$Z(\theta) = \int_{x \in X} \prod_{c \in C} \phi_c(x_c) \tag{2}$$

B. Deep Learning

Traditionally, before the recent upsurge in Deep Learning, the state-of-art method for relation classification are fundamentally based on statistical machine learning. To confront such task directly from a statistical point of view results in propagation of errors from previously derived existing NLP systems [4]. Therefore, in this paper, we think of the task of relation classification via neural network as an intermediate step, starting by exploiting the benefit from deep neural network, which provides lexical and sentence level features extraction.

One of the most representative approaches in relation extraction to use the supervised paradigm in neural networks, which has feature-based methods and kernel-based methods. The feature-based method uses selected sets of features in the form of vectors; for example, Part-of-Speech (POS) tagging feature, and/or syntactic parsing. Deep Learning works as an intermediate step in this paper, itself is an end-to-end

process. To identify the relations between pairs of entities in a diverse text data, we need to combine fundamental features and sentence-level information. The output is a pair of identified relations with a labeled relation. For example, in a sentence " $[aaron \quad hohlbein]_{e_1}$ born august 16 , 1985 in middleton , wisconsin is an american $[soccer \quad player]_{e_2}$ who is currently without a club .", to recognize the $position$ relationship between e_1 : $aaron \quad hohlbein$ and e_2 : $soccer \quad player$, we look up vector forms of all words in the sentence in the form of vectors, then find the target entities in the sentence by the given training entity pair, in addition to all the informations of the sentence learned by the neural network. These vectors are then work as a whole (i.e., concatenation) feed to a softmax classifier to to predict the according marked relationship label.

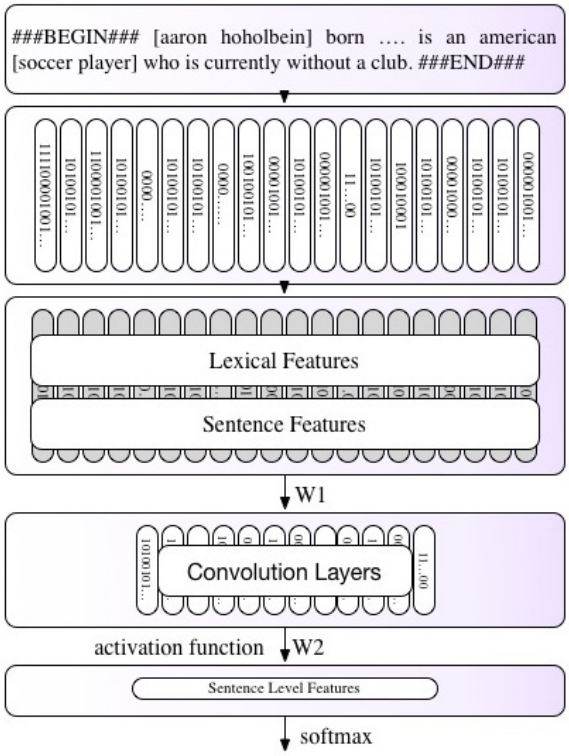


Fig. 1: Network Architecture Overview

1) Feature Extraction: Word embedding has put many features regarding statistical NLP into consideration such as Part-Of-Speech Tagging, word segmentation, and dependency parsing. Lexical and sentence level features; however, can be devised as a cue for making the relation recognition decision. In our application, we have the lexical features shown in blue colored cells (in the Lexical Feature Positions Figure).

a) Sentence level features:
Sentence level features are presented via a max-pooled neural network. Words in a sentence are represented as word features and position features.

i). *Word features* includes vector representation of the word and the vector representation of the word in the

	H1		H2				
###BEGIN###	aaron	hohlbein	born	august			
Left to Entity_1	Entity_1		Right to Entity_1				
			H3		H4		
16 ,	1986	in	middleton	,	wisconsin	is	an
H5			H6				
american			soccer	player	who		is
Left to Entity_2			Entity_2				
			H7				
currently	without	a	club	.	###END###		
			Right to Entity_2				

Fig. 2: Lexical Feature Positions (in blue) *Entity 1 and Entity 2* are the pair of input entities, both the ones on the left and right of them are also parsed. A list of hypernyms of nouns in the sentences from WordNet is also parsed for additional lexical features.

sentence. For the example sentence given previously, all the words in the sentence can be represented as a list of vectors $(x_s, x_1, x_2, ..., x_e)$ (The last index 25 is the ending tag $x_e = ###END###$.) We use context size of w to enrich the word context feature. For example; when $w = 3$, we have the following vector representation for the whole sentence:

$$\{[x_s, x_1, x_2], [x_1, x_2, x_3], [x_2, x_3, x_4], ... [x_{23}, x_{24}, x_e]\}$$

ii). w can only be a very limited amount for a richer word features because it includes all w words sequentially indexed in a sentence. The introduction of *position features* walks around the complicated structure in a sentence by focusing on just the entities. Each token in a sentence has two vector numbers corresponding to the entities. For example, the word {wisconsin} in the example has distance vectors d_1 and d_2 with respect to the relative distance of entity words $[aaron \quad hohlbein]_{e_1}$ and $[soccer \quad player]_{e_2}$, $[d_1, d_2] = [-8, 3]$

b) Adapt Distance Supervision:
Mintz et al., 2009 [5] has an assumption such that if two entities participate in a relation, any sentence that contain those two entities might express that relation, while in reality may result to non-trivial noises. The proposed distant supervison method generates training data automatically via aligning documents with known knowledge base. Features that can be extracted successfully includes sequence of words, POS tags, syntactic (dependency parser) features and performed named entity tagging by Stanford four-class named entity tagger. This model is reported to be able to extract 10k instances of 102 relations at a precision of 67.6%. In addition, Riedel et al., 2010 [6] alleviates noises via deciding whether two entities are related and also mentioned in a given sentence, and then applying "constraint-driven semi-supervision" to train.

2) Neural Networks:: We combine the vectors from feature extraction to have the final vector representations feed to a neural networks.

a) Convolution:

With all the features extraction in vector form as mentioned above, we need to utilize all these local features to predict a relation between a pair of entities globally. In neural networks, deep learning merges all the features as we expected.

i. Linear Transformation:

$$Z_t = W_1 \times X$$

$X \in R^{(n_0 \times t)}$

X is the concatenation of all the feature vectors.

$n_0 = w \times n$, where n is a hyper-parameter corresponding to the dimension of feature vector.

t is the number of nouns in a sentence, including the entity pair.

$W_1 \in R^{n_1 \times t}$ where n_1 refers to the number of input nodes of the first layer in the network.

$$Z_{max}{}^{i} = argmax(Z, i)$$

For i represents the $i-th$ row of the transformed matrix, We select the feature that has the greatest value in each row for the first layer input. This results in a uniformed length of vector, n_1

ii. Non-Linear Layer:

$$G = [ReLU, Tanh, Sigmoid]\,(W_2 \cdot m)$$

W_2 is just another linear transformation, $W_2 \in R^{n_2 \times n_1}$, where n_2 is the number of input nodes of another layer.

$[ReLU, Tanh, Sigmoid]$ We experiment with many non-linear functions and results in the fact that hyperbolic tanh works better as an activation function due to the fact that:

$$\frac{d}{dx} tanh(x) = 1 - tanh^2 x$$

b) Attention on the Entities:

Yankai Lin et al 2016 [2] improved the Distant supervised relation extraction method with additional attention mechanism over its previous work. The improved version has CNN for semantic feature extraction and a sentence-level attention for the purpose of reducing wrong labeling weights.

c) Selective Attention over Instances:

For each previous layer, we add attention attribute:

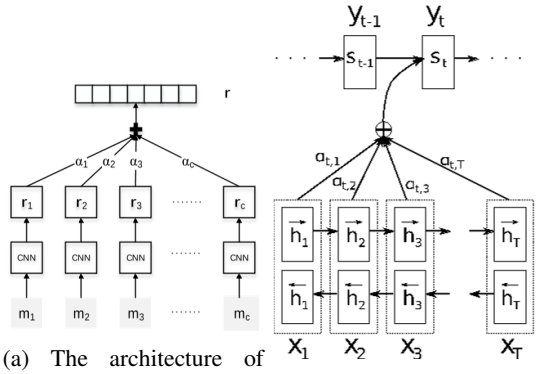
$$a_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$$

where $e_i = x_i \cdot A \cdot r$

A is a matrix, r represents the relation in matrix. Therefore, e_i means the matching between the sentence and entities.

The final result conditional probability can be represented as the following through a softmax layer:

$$p(r|S, \theta) = \frac{\exp(o_r)}{\sum_{k=1}^{n_r} \exp(o_k)}$$



(a) The architecture of sentence-level attention-based CNN. Used in Lin 2016 [2]

(b) Classical attention mechanism used by Dzmitry 2014. [7]

Fig. 3: Comparison between two attentions.

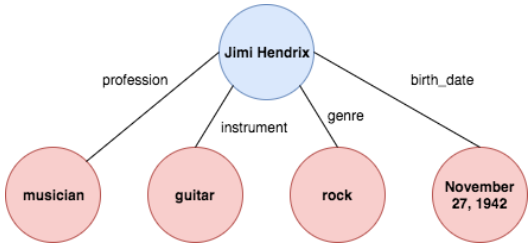


Fig. 4: F

n_r = total number of relations

o = the final output of the neural network, which corresponds to the scores associated to all relation types.

θ = the parameter of this model

$$o = Ms + d$$

Therefore, the objective function using cross-entropy for the training is:

$$J(\theta) = \sum_{i=1}^{s} \log(p(r_i|S_i, \theta))$$

III. Problem Definition and Algorithm

A. Task

Let s_p be the subject, or principal entity, of a document and let $s_0, ..., s_n$ be secondary entities within document , X. In our experiments s_p is the subject of a biographical wikipedia article and s_i's are entities such as dates, names, place, ect. Let R be the set of possible relationship between two entities. The goal is to find the most likely relation assignment for each entity, $r_0, ...r_n = argmax_R P(r_0, ...r_n|s_p, s_0, ..., s_n, X)$.

B. Algorithm

Our algorithm combines the deep learning model with a graphical to describe the pairwise potential between relations. We parameterize our model using an exponential

family where the unary potentials are provided as the output of the deep learning model and pairwise potentials capture the interaction. We experiment with two different models a fully connected model and a sequentially connected model.

1) Linear Chain MRF: In our initial experiments we use a linear chain MRF to model the data. Using this model we are able to recover sequential structure in the data. For example, the model learns that is it much more likely that an entity with the "birth day" relation is much more likely to precede the "death day" relation. We define the following transition parameters to capture this interaction.

$$\phi(r_i, r_j) = \theta_{ij} \tag{3}$$

We compine thise pairwise parameters with unary potentials, denoted $\Phi(r)$, for each assignment that are provided by the deep learning model. And our fully model is given by

$$P(r_0, ...r_n) = \frac{1}{Z(\theta)} \Pi_{T-1} \phi(r_i, r_{i+1}) \Pi_T \Phi(r_i) \tag{4}$$

One advantage of this model is that the induced graph is a tree and we can perform efficient exact inference. However one drawback of this model is we are unable to capture more global dependencies between relations. We are able to improve upon this using a fully connected pairwise MRF.

2) Fully Connected MRF: In order to model global constraints between labels we add connections to make the model fully connected. We parameterize the pairwise potentials the same way as the linear chain model.

$$P(r_0, ...r_n) = \frac{1}{Z(\theta)} \Pi_M \Pi_{N<i} \phi(r_i, r_j) \Pi_M \Phi(r_i) \tag{5}$$

While this model is more expressive, it becomes more computationally complex. Therefore we need approximate methods for learning as well as for inference. For parameter learning we experimented with both MLE as well as pseudo likelihood. For MLE we use the fact the gradient of the loss is given by the following equation.

$$\nabla L(\theta) = \bar{f} - E_\theta\{f\} \tag{6}$$

We estimate $E_\theta\{f\}$ using Gibbs sampling. However, we found that we required many samples in order to get a good approximation due to the number of samples which slowed training. As an alternative we found maximizing the pseudo likelihood to be a much more efficient training procedure.

$$L(r_0, ...r_n) = \sum_M \sum_N log(P(r_n | r_0, ...r_N)) \tag{7}$$

The parameters can then be learned using gradient descent. We found that this training procedure produced between empirical results than Gibbs sampling.

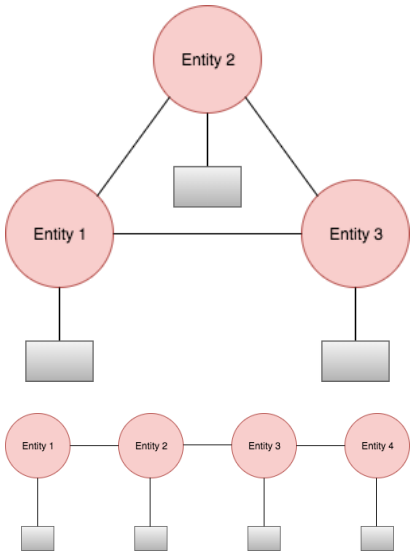


Fig. 5: Number of neurons which each word activate in the max pooling layer

Example Biography:
Ava Acres is an American **actress** . She is best known for playing the role of Erik in "Happy Feet Two". Acres' older sister, Isabella, is also an actress.

relation expressed occupation - actress

Fig. 6: Example biography from wikipedia dataset

IV. Experiments

A. Data

To evaluate our model, we use a dataset of Wikipedia Biographies [3]. This dataset contains 728,321 biographies from wikipedia covering a variety of domains. each biography has the data from the wikipedia infobox included. This contains different relations of entities to the subject of the document such as day of birth, genre(for musicians), what team they played for , ect. We restrict our data to the 100 most frequent relationships expressed in the data and train the model to predict these relations.

We think that this task is especially well suited for our model because, while there are different ambiguities in the text, there are very strong dependencies between distributions that the graphical model can capture. In figure 2 we show the first 2 principal components of the relationship data for biographies that contain the "goals", "office", and "genre" relation. We see very clear separation as the relationships expressed in each of these tend to be very different from one another.

B. Methodology

For cross validation we use the provided train, validation, test, split provided.

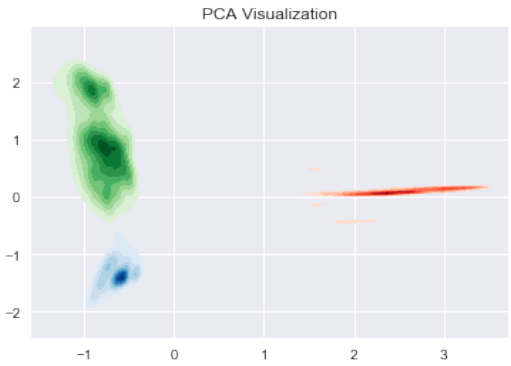


Fig. 7: First 2 principal components for biographies containing ”goals”(red), ”genre”(blue), or ”office”(green)

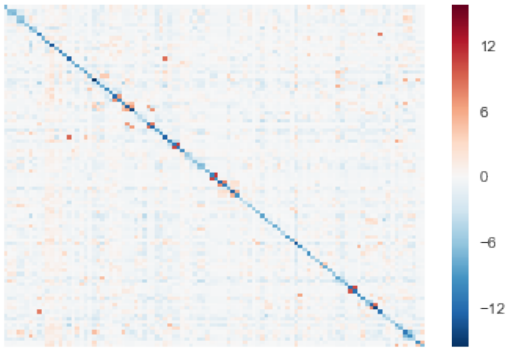


Fig. 8

The parameters of the graphical model are learned using mini-batch gradient descent with learning rate of .01 and batch size 128. We minimize pseudo likelihood of the data to avoid computing the partition function. We also use a Gaussian prior for regularization.

C. Results

For each model, we report macro precision, recall, and F-score. We compare both deep learning architectures and found that the bidirectional GRU with attention outperforms that PCNN with attention. We then applied both the linearly connected MRF and fully connect MRF to the bidirectional GRU. We found that both approaches are able to increase performance, however the fully connected model is able to increase performance by a larger margin.

Method	Precision	Recall	F score
PCNN+ATT	0.294	0.310	0.301
BGRU+ATT	0.352	0.326	0.339
BGRU+ATT Lin CRF	0.359	0.336	0.347
BGRU+ATT FC CRF	**0.379**	**0.347**	**0.3622**

1) Feature Analysis: In order to better understand the structure in the data that the model is learning we analyze the parameters of the graphical model. For each relationship we look at which other relationships have a high potential and low potential with. We found that the model is able to identify several different types of biography such as athlete, politician, or musician.

Relation	Highest Pairwise Potential
goals	clubnumber , nationalgoals, position
genre	instrument, label
country	office, branch, yearsactive

TABLE I: Results. Ours is better.

V. Conclusion

In this work we have looked at an approach to the relation extraction task by creating a joint model of all relations expressed in a document. We have done this by augmenting state of the art deep learning models with graphical models to model the pairwise relationships between relations. Using both a linearly connected and fully connected model we have shown this approach gives improved performance on the Wikipedia biography dataset. Analysis of the parameters of our learned model show that is is able to successfully learn which relationships have high probability of occurring together. Future work in this area may include an end to end deep learning approach using graph convolutional networks.

VI. Bibliography

[1] Chris Q., Hoifung P. 2017. *Distant Supervision for Relation Extraction beyond the Sentence Boundary.* arXiv:1609.04873v3.

[2] Yankai L., Shiqi S., Zhiyuan L., Huanbo L., and Maosong S. 2016. *Neural Relation Extraction with Selective Attention over Instances.*. In *Proceedings of ACL.*

[3] Rmi Lebret, David Grangier and Michael Auli, EMNLP 2016. *Neural Text Generation from Structured Data with Application to the Biography Domain.* arXiv:1603.07771 [cs.CL]

[4] Nguyen Bach, Sameer Badaskar, 2007 *A Review of Relation Extraction. Literature review for Language and Statistics II.*

[5] Mintz, M., Bills, S., Snow, R. and Jurafsky, D. (2009). *Distant Supervision for Relation Extraction Without Labeled Data*. 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 (p./pp. 1003–1011), Stroudsburg, PA, USA: Association for Computational Linguistics. ISBN: 978-1-932432-46-6

[6] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. *Modeling relations and their mentions without labeled text.* In Proceedings of ECML-PKDD, pages 148163.

[7] Dzmitry Bahdanau, Kunghyun Cho, Yoshua Bengio. 2014. *Neural Machine Translation by Jointly Learning to Align and Translate*