Literature Survey on Relation Extraction and Relational Learning

Kush Goyal
Indian Institute of Technology, Bombay
kushgoyal@cse.iitb.ac.in

Pushpak Bhattacharyya
Indian Institute of Technology, Bombay
pb@cse.iitb.ac.in

Abstract

Semantic relation extraction between entities plays key role in many applications in natural language processing and understanding, information retrieval, text summarizing, etc. These application require an understanding of the semantic relations between entities. We present a comprehensive review of various aspects of the entity relation extraction task. We also present a review of relation extraction techniques in medical domain. Relation extraction techniques are used in relation extraction from unstructured text which can be used for further processing. These extracted relations are useful to construct a knowledge base. There are various relational learning methods which are used to learn new relations with the help of existing relations. Here we present a review of recent research on relational learning techniques as link prediction techniques.

1 Introduction

Semantic relationships between entities have an important role in natural language understanding and information retrieval applications. In this paper, we discuss a comprehensive review of few important techniques in relation extraction field. We also discuss about the performance of different approaches and comparative study of existing techniques. We also discuss about recent research on link prediction algorithms and their importance in relation extraction task.

We discuss three major class of approaches in relation extraction field. We begin with simple rule-based approach in Section 2 to more complex semi-supervised approaches in Section 3. We have a short discussion about semi-supervised methods in Section 4. After that we discuss about existing relation extraction methods in medical domain in Section 5. We also focus on link prediction techniques in Section 6 which are helpful to predict the missing links from a knowledge base. In this section we discuss about current research on link prediction algorithms.

2 Rule-Based Systems

Rule-Based systems follows multiple hand built rules to perform information extraction task. Rules are defined after a detailed analysis of multiple examples and then adopted by the system.

For example, New *fluoroquinolones* such as *ofloxacin* would advantageous in the treatment of *chronic obstructive airways disease (COPD) aggravation* and requires mechanical ventilation.

While reading this sentence, we can predict a hyponym relation between *fluoroquinolones* and *ofloxacin* due to presence of connecting word 'such as' between entities. Some rules are specially designed for domain specific information extraction tasks. For example, in the given sentence relation **'TREATS'** exists between entities as

Ofloxacin **TREATS** *Chronic obstructive airways disease exacerbated.*

Relation **'TREATS'** is defined between the given pair of entities according to the rule

Pharmacologic Substance **Treats** Disorders *Ofloxacin* is a 'Pharmacologic Substance' and *Chronic obstructive airways disease exacerbated* is a 'Disorder', so the relation 'TREATS' holds between given pair of entities. SemRep is a relation prediction tool in medical domain which follows many of such rules for relation extraction between medical entities (Rindflesch and Fiszman, 2003). Precision of SemRep is 0.69 with 0.88 recall value (Rosemblat et al., 2013).

3 Supervised Methods

Supervised methods are important and very common in natural language processing tasks. The supervised system learns from various examples with predefined feature set. Features for supervise systems are designed very carefully by experts and are domain specific. In relation extraction task, supervised learning is used for classification (Binary or multi-class) problem. There are many existing machine learning techniques which can be useful to train classifiers for relation extraction task.

For example, consider a simple binary relation classifier for a relation R between given pair of entities e_1 and e_2. Mapping function f (.) can be defined as

$$f_R(T(S)) = \begin{cases} \text{True}, & \text{if}\ {}_{e_1}R_{e_2}, \\ \text{False}, & \text{otherwise.} \end{cases} \quad (1)$$

Where T(S) is the set of features extracted for entity pair e_1 and e_2. These features can be linguistic features from the sentence where these entities are mentioned or path features from a knowledge graph. The mapping function f (.) defines the existence of relation R between entities. Support Vector Machine (SVM) (Byun and Lee, 2002) is an example of classifier which can be used to train as a binary relation classifier. Features for such classifier for relation prediction can be linguistic features (POS tags, dependency tree features, entity features, etc.) or path features (in a knowledge graph) in a specific domain. These features are carefully designed by experts and this is a very difficult task to do. In convolutional neural network based approach, we use word embeddings, position vectors, etc for our input data and system identify appropriate features for classification task.

3.1 Feature Based Methods

In this approach, a set of relevant features are designed by domain experts for a classification problem. Later this set of features are given to classifier for training and classification purpose. For relation extraction task, sentences with predefined entities are used to construct feature vector through feature extraction process (Kambhatla, 2004; Hong, 2005; Minard et al., 2011). Commonly used feature for relation extraction task are described below.

- **Lexical Features:** In this feature set, lexical features such as position of mentioned pair of entities, number of words between mentioned pair, word before or after mentioned pair, etc. are used to capture context of the sentence. With this, bag-of-words (Hasegawa et al., 2004) approach can be useful to represent represent sentence and words as a feature in our feature vector.

- **Syntax Tree Features:** In this feature set, grammatical structure of the sentence and mentioned pair are used for feature creation. For example, part of speech tags for each mentioned pair, chunk head, etc., can be used as a feature for relation extraction (Kambhatla, 2004).

- **Dependency Tree Features:** Dependency tree provides us the words on which mentioned pair is dependent and we can use such words and their part-of-speech tags in our feature set. With this we can also use dependency tree path path between mentioned pair, path labels, distance between mentioned pair in dependency tree, etc., in our feature set (Reichartz et al., 2009).

- **Entity Features:** A relation can exist between certain type of entities, for example treatmentForMedicalProblem can exist between a treatment entity and problem entity. So, type of mentioned pair of entities are also important feature values for classification purpose. Entity features also includes presence of other medical entities between mentioned pair.

- **Word Embedding Features:** Although lexical features represent structure of sentence with mentioned pair, we can use word embeddings to represent our mentioned pair (Mikolov et al., 2013). Word embedding features have an important role in Named Entity Recognition, Chunking, Dependency parsing, semantic role labeling, and relation extraction.

3.2 Convolutional Neural Network Based Approach

Features generation for relation extraction use natural language processing modules extensively till recent past. These features are not always error free and errors in these features propagate to the next level and results in error in relation prediction task. In this section, we will discuss convolutional neural network (CNN) for relation extraction which does not rely on complicated feature engineering. CNN automatically learns features from sentences and minimizes the dependency on external modules and resources(Nguyen and Grishman, 2015).

Input to the convolutional neural network can be words represented by word embedding and positional features based on the relative distance from the mentioned entities. So there will not be any dependency to other Natural language processing modules. The convolutional layers provide a local correlation between features at the lower layers and learn long distance features in the higher layers. Convolutional operations are carried out in each layer which takes care of the local convolution of the input. A max pooling layer cuts the input dimensions without losing the dominated features. A nonlinear layer at the end transforms input to a linearly separable space. Convolution neural network shows promising results in the relation extraction tasks.

Baseline structure of CNN approach consists of three layers, including convolution, max pooling, and non linear layer. There are lots of hyper parameters which are needed to tune for the best performance of CNN systems, makes this approach a little challenging. Word embedding dimensions, number of units in hidden layer, number of hidden layers are example of such parameters. Convolution filter size is also needed to tune for the best performance of the CNN systems.

4 Semi Supervised Methods

Supervised Methods are useful when number of training examples are sufficiently large. In case of example or training data deficiency, semi supervised methods provide a way to train the model which follows bootstrapping techniques. In this approach, available examples or training data is used as seed instances. First, classifier learns from these examples and then tested on test data. After testing, classifier adds valid test cases in its training set. Thus, the training set grow up to a sufficient amount.

NELL (Never-Ending Learning) (Mitchell et al., 2015) is a system that follows the semi supervised learning method to learn the relations between concept entities. NELL is continually learning from the web and using semi supervised methods to train its classifier with growing set of test and train data. Performance of NELL system varies across predicates: the precision for categories like ”river,” ”body part,” and ”physiological condition” is above 0.95 and confidence scores for such categories are also in the top. On the other hand, for ”machine learning author” and ”city capital of country(x,y)” precision are quite low i.e., below 0.5 (Mitchell et al., 2015).

5 Relation Extraction in Medical Domain

There are various supervised and semi supervised techniques exist in the field of relation extraction. In medical domain the impact of supervised approach is very significant. SVM based relation extraction systems show state of the art relation extraction system using various Syntactic, dependency, lexical and domain knowledge from the existing systems like UMLS (Unified Medical Language System). On the other hand, CNN based relation extraction systems are also used in medical domain in recent years.

5.1 Feature based methods

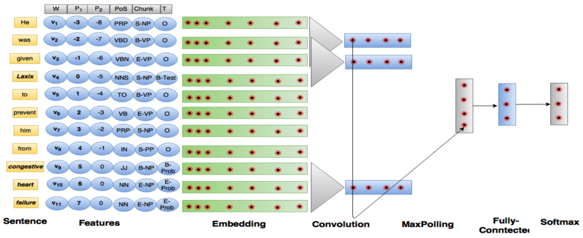
SVMs are most common approaches among the most effective relation extraction systems. Since medical data consist of a large number of concept pairs which are not related, some relation extraction system do this task in two steps. In first step, all the related pairs are separated from the non-related pairs. And in second step, these system identifies nature of these relationships. Many of these systems used lists of n-gram with specific semantics and hand-built linguistic rules. The state of the art system in relation extraction system, Rink et al. (2011) has shown F-score of 73.7 for the i2b2 data set using SVM classifier with a rich feature set.

5.2 CNN for medical relation extraction

Sahu et al. (2016) proposed a CNN based relation extraction method for medical data as shown in the Figure 1[1]. Input to this model is a complete sentence with pre-annotated medical entities and output from the model is a vector of probabilities corresponding to all existing relation types. Each feature is represented by a vector which is initialized randomly except word embeddings. Word

[1] Source: https://arxiv.org/pdf/1606.09370.pdf

embedding vectors are directly taken from the pre-trained word embeddings from medical text. Then this complete feature vector is given to neural network's initial layers as shown in Figure 1. Subsequent layers of this architecture are described below.



Sentence	W	P_1	P_2	PoS	Chunk	T
He	v_1	-3	-8	PRP	S-NP	O
was	v_2	-2	-7	VBD	B-VP	O
given	v_3	-1	-6	VBN	E-VP	O
Laxix	v_4	0	-5	NNS	S-NP	B-Test
to	v_5	1	-4	TO	B-VP	O
prevent	v_6	2	-3	VB	E-VP	O
him	v_7	3	-2	PRP	S-NP	O
from	v_8	4	-1	IN	S-PP	O
congestive	v_9	5	0	JJ	B-NP	B-Prob
heart	v_{10}	6	0	NN	E-NP	E-Prob
failure	v_{11}	7	0	NN	E-NP	E-Prob

Figure 1: Convolutional neural network architecture for relation extraction

- **Feature Layer:** This layer includes feature values for input to CNN system. Each word in the sentence is represented with a vector of 6 different features. This feature set includes word itself, distance of the word from the first entity (P1) in number of words, distance of the word from the second entity (P2) in number of words, part-of-speech (PoS) tag for the word, chunk head tag of the word and type of the entity if word is a part of mentioned pair.

- **Embedding layer :** This layer maps each feature value with it's corresponding feature vector and then concatenate all the embeddings. To generate word embeddings for each feature value we can train embeddings on the text corpus or pre-trained word embeddings can be used. Word embeddings trained on medical data are available on various web-resources [2].

- **Convolution Layer:** This layer extracts local features from each part of the sentence.

- **Max Pooling Layer:** Outcome of convolution layer is passed to max polling layer. This layer extracts global features for the complete sentence.

- **Fully Connected Layer:** The outcome from previous layer is a sequence p which comes with different filters. This sequence now is a global feature for a sentence because it came by taking max over complete sentence. Later this max value is passed to the next layer for classification of relation.

- **Softmax Layer:** In final layer, softmax classifier is used over global feature vector to calculate correct class of relation.

6 Link Prediction Techniques

In natural language processing, we enhance computer system's ability such that it can understand and process the text written in natural language like English, Hindi, etc. There are many existing techniques to process the text which is written in natural language. With the existing techniques, use of available knowledge resources may helpful to improve accuracy. For example, if we are processing text for relation extraction then with relation extraction techniques we can refer available knowledge resources to verify or get the information about existing relations between mentioned pairs. Since, knowledge resources are always incomplete i.e. all the information is not present in the knowledge base, link prediction techniques allowed us to use available information to draw new information. Path rank algorithm and subgraph feature extraction are recent research in the field of link prediction techniques.

Path Rank Algorithm (PRA) is one of the efficient relational learning approaches proposed by Lao et al. (2011). In the subsequent sections we will discuss about basics of PRA and improvement over PRA algorithm.

6.1 Introduction to PRA

Relational learning considers both machine learning and knowledge representation. Machine learning approaches are widely used to improve behavior of the system over time with experience. And a more expressive knowledge representation describes knowledge as entities and relationship between entities. These representations are known as *relational* or *logical representations*, if they are derived from first-order logic. PRA works with these representation of knowledge to perform relational learning.

Consider a simple knowledge base consisting of a set C of concepts and a set R of labels. Each label r denotes some binary relation in knowledge base. This represents a concrete knowledge base as a directed, edge-labeled graph K = (C,

[2] http://bio.nlplab.org/

R, T) where T ⊆ C × R × C is the set of labeled edges (c, r, c′). Each triple in knowledge base represents an instance r(c, c′) of the relation r ∈ R. Each concept corresponds to an entity in the knowledge domain such as *ChetanBhagat*, or an abstract notation *Writer*. Each edge represents an existing relation between two entities such as *Wrote(ChetanBhagat, HalfGirlfriend)*, or the category of an entity *IsA(ChetanBhagt, Human)*. Inverse of a relation r is denoted as r^{-1} such that $r^{-1}(c', c) \Leftrightarrow r(c, c')$. For instance *People_With_Profession*$^{-1}$ is equivalent to *Profession_Has_Instance*. The knowledge base may be incomplete, that is, r(c, c′) exist in fact but (c, r, c′) ∉ T.

PRA performs generic relational learning task called *link prediction*: given a directed edge-labeled graph represents background knowledge, a source node s and a relation r, find the set of nodes G, such that r(s,t) for each t in G.

6.2 Horn Clause and Random Walk Inference

Lao et al. (2011) introduced *random walk inference* to formulate relational learning as statistical classification problem as, given a query node pair *(s, t)*, generate random walk features that summarize their relational neighborhood, and then train classifier based on these features. These random walk features are denoted as $P(s \rightarrow t; \pi)$, the probability of reaching from node s to node t through a particular path type π and following a particular random walk process. For example, consider the question:

Whether *Amitabh* has *Actor* as his *Profession*? Random walk features (or path features) for the above query can be represented as

- $P(Amitabh \rightarrow Actor; \langle HasSon,\ Profession \rangle)$
- $P(Amitabh \rightarrow Actor; \langle Mention, Mention^{-1},\ Profession \rangle)$

Each such path π_i is considered as a feature in random walk feature set. These paths follows rules of a very constrained subset of logical expressions (Horn clause with chain structures), which can be learned efficiently with any machine learning approach (Lao et al., 2011). These path features or random walk features are associated with a value, i.e., probability of reaching target t from source s following path π_i.

Paths in PRA correspond to a specific class of Horn clauses. For example, for a relation r = *AthletePlaysForSeries* and the path $\pi = \langle\ AthletePlaysForTeam,\ TeamPlaysInSeries \rangle$, corresponding Horn clause is as

$$AthletePlaysForTeam(s, y) \wedge$$
$$TeamPlaysInSeries(y, t) \rightarrow$$
$$AtheletePlaysForSeries(s,\ t)$$

6.3 Path Rank Algorithm

The basic goal of path rank algorithm is to predict the missing link between source and target in a knowledge graph. For example, consider a part of knowledge graph in Figure 2. Here, to know about profession of Amitabh, we can consider profession of his friends or family of Amitabh. These consideration form some rules to answer a particular query. These learned rules are very useful and important but not all of them are accurate. Relational learning process also includes the task of finding useful rules very efficiently and make an accurate prediction based on a weighted combination of these rules. In terms of PRA, these rules are called as path types. In our example one path type can be $\langle HasSon,\ Profession\ \rangle$

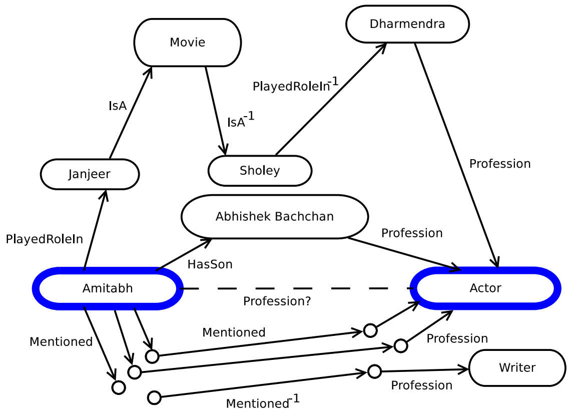


Figure 2: Link Prediction with inference over knowledge base. IsA^{-1} is the inverse of IsA. $PlayedRoleIn^{-1}$ is the inverse of $PlayedRoleIn$. The $Mentioned$ edges are the edges which represent each related entity with *mentioned* edge appeared in some sentence.

Consider a simple knowledge base as an edge-labeled graph K = (C, T) where C is the set of concepts and T is the set of labeled edges (c, r, c'). A *path type* in K is defined as a sequence of edge types $\pi = \langle\ r_1, r_2 ...., r_n \rangle,\ r_i \in R$. Given a path type $\pi = \langle\ r_1, r_2...., r_n\rangle$, and a starting node $s = v_0$, $P(s\ \rightarrow t;\ \pi)$ is defined as the probability of reaching t from s by a random walk that follows π. Suppose the random walk is at node v_i by traversing edges labeled $\langle\ r_1, r_2...., r_i\rangle$. Then v_{i+1} is selected at random from all nodes reachable from v_i by edges labeled r_{i+1}. A path type π is active for pair (s,t) if $P(s\ \rightarrow t;\ \pi)\ >\ 0$. Thus probability is defined with a recursive function as

$$P(s \rightarrow t; \pi) = \sum_z P(s \rightarrow z; \pi')P(z \rightarrow t; r) \tag{2}$$

Where r is the last relation in the path π, and π' is its prefix, such that adding r to π' gives π. If π is the empty path, i.e., no edge between u and v, $P(u\ \rightarrow\ v;\ \pi)\ =\ 1$ if u = v and 0 otherwise. The probability of a particular relation r between two nodes u and v is defined as P($u\ \ \rightarrow v;\ r) = 1/|r(u)|$ if r(u,v) and 0 otherwise. r(u) represents range of the relation r for node u.

6.3.1 Relational learning using PRA

For the task of prediction whether r(s,t) is true, $P(s\ \ \rightarrow\ t;\ \pi)$ is used as a key feature. Let $Q = \pi_1, ..., \pi_n$ be a set of path types that occur in the graph with $|\pi_i| \leq l$, where $|\pi_i|$ is the length of path π_i, and θ_π is the weight assigned to π.

$$score(s, t) = \sum_{\pi \in Q} P(s \rightarrow t; \pi)\theta_\pi \tag{3}$$

Score(s,t) in Equation 3 encodes a PRA model's confidence that nodes s and t are connected by the relation r. The learning problem for PRA is to assign proper weights to different path types, so that the scoring function has high values for node t with respect to query node s where r(s, t) is true and low values for other nodes (Lao, 2012).

6.4 Feature selection for PRA

There may exist exponential number of path types depend on the length of the maximum path length l. Since feature space is very large, we need to perform feature selection to allow effective learning. Consider a set of training queries (s_i, G_i) for i = 1...n, where G_i is the set of good answer to query s_i. G_i is defined as $G_i = \{t|r(s,t)\}$. The probability of reaching any correct answer following π reflects the accuracy of a path type π

$$acc(\pi) = \frac{1}{n}\sum_i P(s_i \rightarrow G_i; \pi) \tag{4}$$

The hits of a path π shows the number of queries for which path π results to any correct answer:

$$hits(\pi) = \sum_i I(P(s_i \rightarrow G_i; \pi) > 0) \tag{5}$$

where $I()$ is the indicator function. PRA includes only those paths which obey the following two conditions

1. $\alpha\ \leq\ acc(\pi)$
2. $\beta\ \leq\ hits(\pi)$

Where thresholds α and β are tuned empirically on training data.

6.5 Relational Learning with Subgraph Feature Extraction

Subgraph feature extraction (SFE) is a simpler, efficient and more expressive approach than PRA for computation of feature matrices from graphs. Gardner and Mitchell (2015) have shown that random walk probabilities computed by PRA provided no benefit to performance on this task so they could safely be dropped. SFE computes much richer features than paths between two nodes in a graph.

PRA is a two step process, where the first step finds potential path types between node pairs to use as features in a statistical model. The second step generates feature matrix by computing random walk probabilities associated with each path type and node pairs. Second step is computationally very expensive (computation time is proportional to the average out-degree of the graph to the power of the path length for each cell in the feature matrix). Whereas SFE proposes another way to generate feature matrix over node pairs in a graph with the aim to improve efficiency and the expressivity of the model.

For each node u in the data (where u can be a source node or a target node), SFE constructs a subgraph centered around u using k random walks. Each random walk that starts from u, follows some path type π and ends at some intermediate node i. SFE keeps all of these (π, i) pairs as the characterization of the subgraph around u and refers this subgraph as G_u. To construct a feature vector for a source-target pair (s_j, t_j), SFE takes the subgraphs G_{s_j} and G_{t_j} and merges them on the intermediate nodes i. If there exists an intermediate node i in both G_{s_j} and G_{t_j}, SFE takes the path types π corresponding to i and combines subgraphs (reversing the path types coming from target node t_j). If there exist an intermediate node t_j in some path type from source node s_j, no combination of path types is necessary. Else if there exists an intermediate node s_j in some path type coming from target node t_j, reverse the path type and no combination is needed. This creates a feature space which is exactly the same as that constructed by PRA (Gardner and Mitchell, 2015). SFE takes all of these combined path types as binary features for pair (s_j, t_j) to construct the feature vector. SFE may not get a complete charactcterization of the graph for nodes having higher degree. The reason is, unconstrained random walks are quite low to find important path types for higher degree nodes using few random walks.

While using a BFS instead of random walks to obtain the subgraphs G_{s_j} and G_{t_j} for each node pair, an increment in the number of path type features is found with an adequate increase in performance. Some expressive features in SFE feature space are discussed in the subsequent sections.

6.5.1 PRA-style Features

These are the same features as discussed above. These features are generated by intersecting the subgraphs G_{s_j} and G_{t_j} on the intermediate nodes.

6.5.2 Path bigram features

For any path π between a source node s and a target node t, a feature for each relation bigram is used in feature set of SFE. In the example in Figure 3, bigram features are ”BIGRAM:@START@-ALIAS”,”BIGRAM:ALIAS-is married to”,”BIGRAM:is married to-ALIAS”, and ”BIGRAM:ALIAS-@END@”.

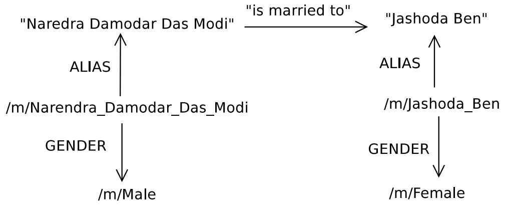


Figure 3: An example graph with two nodes in freebase

6.5.3 One-sided Features

SFE uses one-sided path to describe an edge sequence that starts at a source or a target node in the data, but does not necessarily terminate at a corresponding target or source node. These one-sided path types are used to model which sources and targets are good candidates for participating in a particular relation. For the graph described in Figure 3, one-sided features are ”SOURCE:-GENDER-:male”, ”TARGET:-GENDER-:female”, ”SOURCE:-ALIAS-:Narendra Damodar Das Modi”, and ”SOURCE:-ALIAS-is married to-:-:Jashoda Ben”.

6.5.4 Vector Space Similarity Features

SFE uses factorization of the knowledge base tensor (Gardner et al., 2014) to obtain vector representations of relations and replaces each edge type in a PRA-style path with the edges that are similar to it in the vector space. A special ”any edge” symbol is used to represent that all other edge types are similar to this edge type. In Figure 3 ”spouse of” is considered similar to ”is married to”, vector space similarity features are ”VECSIM:-ALIAS-is married to-ALIAS-”, ”VECSIM:-ALIAS-spouse of-ALIAS -”, ”VECSIM:-ALIAS-@ANY_REL@-ALIAS-” and ”VECSIM:-@ANY_REL@-is married to-ALIAS-”.

6.5.5 Any Relation Features

This feature allows any path type that used a surface relation to match any other surface relation with non-zero probability. This feature replace such path type by @ANY_REL@ symbol and follow the same process to learn the relation.

7 Conclusion

In this survey paper, we have discussed different approaches which are widely used for relation extraction task. We have also discussed importance of relation extraction techniques in natural language processing field. Several approaches such as tree kernel outperforms feature based approaches in supervised learning. For relation extraction, convolutional neural network based approaches using word embedding and feature embedding (Sahu et al., 2016) have shown F1-score of 71.16 and feature based state of art system Rink et al. (2011) obtained F-Score of 73.7 for relation extraction and classification task with 2010 i2b2/VA relation data set.

In link prediction techniques, SFE is faster and performs better than PRA on link prediction task. Gardner and Mitchell (2015) have shown experimentally that SFE can reduce running time by an order of magnitude and SFE improves mean average precision from .432 to .528 and mean reciprocal rank from .850 to .933 compared to PRA.

References

Hyeran Byun and Seong-Whan Lee. 2002. Applications of support vector machines for pattern recognition: A survey. In *Pattern recognition with support vector machines*, Springer, pages 213–236.

Matt Gardner and Tom Mitchell. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1488–1498.

Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases .

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 415.

Gumwon Hong. 2005. Relation extraction using support vector machine. In *International Conference on Natural Language Processing*. Springer, pages 366–377.

Nanda Kambhatla. 2004. 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, page 22.

Ni Lao. 2012. *Efficient random walk inference with knowledge bases*. Ph.D. thesis, Pennsylvania State University.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 529–539.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Anne-Lyse Minard, Anne-Laure Ligozat, Asma Ben Abacha, Delphine Bernhard, Bruno Cartoni, Louise Deléger, Brigitte Grau, Sophie Rosset, Pierre Zweigenbaum, and Cyril Grouin. 2011. Hybrid methods for improving information access in clinical documents: concept, assertion, and relation identification. *Journal of the American Medical Informatics Association* 18(5):588–593.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. volume 2, pages 365–371.

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2009. Dependency tree kernels for relation extraction from natural language text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pages 270–285.

Thomas C Rindflesch and Marcelo Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of biomedical informatics* 36(6):462–477.

Bryan Rink, Sanda Harabagiu, and Kirk Roberts. 2011. Automatic extraction of relations between medical concepts in clinical texts. *Journal of the American Medical Informatics Association* 18(5):594–600.

Graciela Rosemblat, Dongwook Shin, Halil Kilicoglu, Charles Sneiderman, and Thomas C Rindflesch. 2013. A methodology for extending domain coverage in semrep. *Journal of biomedical informatics* 46(6):1099–1107.

Sunil Kumar Sahu, Ashish Anand, Krishnadev Oruganty, and Mahanandeeshwar Gattu. 2016. Relation extraction from clinical texts using domain invariant convolutional neural network. In *BioNLP at ACL-2016*. volume 15, page 206.