Techniques for Jointly Extracting Entities and Relations: A Survey

Sachin Pawar[1,2], Pushpak Bhattacharyya[2], and Girish K. Palshikar[1]

[1] TCS Research & Innovation, Pune, India-411013
[2] Indian Institute of Technology Bombay, Mumbai, India-400076
sachin7.p@tcs.com, pb@cse.iitb.ac.in, gk.palshikar@tcs.com

Abstract. Relation Extraction is an important task in Information Extraction which deals with identifying semantic relations between entity mentions. Traditionally, relation extraction is carried out after entity extraction in a "pipeline" fashion, so that relation extraction only focuses on determining whether any semantic relation exists between a pair of extracted entity mentions. This leads to propagation of errors from entity extraction stage to relation extraction stage. Also, entity extraction is carried out without any knowledge about the relations. Hence, it was observed that jointly performing entity and relation extraction is beneficial for both the tasks. In this paper, we survey various techniques for jointly extracting entities and relations. We categorize techniques based on the approach they adopt for joint extraction, i.e. whether they employ joint inference or joint modelling or both. We further describe some representative techniques for joint inference and joint modelling. We also describe two standard datasets, evaluation techniques and performance of the joint extraction approaches on these datasets. We present a brief analysis of application of a general domain joint extraction approach to a Biomedical dataset. This survey is useful for researchers as well as practitioners in the field of Information Extraction, by covering a broad landscape of joint extraction techniques.

Keywords: Relation Extraction · Entity Extraction · Joint Modelling · End-to-end Relation Extraction

1 Introduction

Entities such as PERSON or LOCATION are the most basic units of information in any natural language text. Mentions of such entities in a sentence are often linked through well-defined semantic relations (e.g., EMPLOYEE_OF relation between a PERSON and an ORGANIZATION). The task of Relation Extraction (RE) deals with identifying such relations automatically. Apart from the general domain entities of types such as PERSON or ORGANIZATION, there can be domain-specific entities and relations. For example, in Biomedical domain, an example relation type of interest can be SIDE_EFFECT between entities of types DRUG and ADVERSE_EVENT.

A lot of approaches [27, 7, 2, 19, 16] have been proposed to address the relation extraction task. Most of these traditional Relation Extraction approaches assume the information about entity mentions is available. Here, information about entity mentions

consists of their boundaries (words in a sentence constitute a mention) as well as their entity types. Hence, in practice, any **end-to-end relation extraction** system needs to address 3 sub-tasks: i) identifying boundaries of entity mentions, ii) identifying entity types of these mentions and iii) identifying appropriate semantic relation for each pair of mentions. The first two sub-tasks of end-to-end relation extraction correspond to the *Entity Detection and Tracking (EDT)* task defined by the the Automatic Content Extraction (ACE) program [4] and the third sub-task corresponds to the *Relation Detection and Characterization (RDC)* task.

Traditionally, the three sub-tasks of end-to-end relation extraction are performed serially in a "pipeline" fashion. Hence, the errors in any sub-task are propagated to the subsequent sub-tasks. Moreover, this "pipeline" approach only allows *unidirectional flow of information*, i.e. the knowledge about entities is used for extracting relations but not vice versa. To overcome these problems, it is necessary to perform some or all of these sub-tasks jointly. In this paper, we survey various end-to-end relation extraction approaches which jointly address entity extraction and relation extraction.

2 Problem Definition

The problem of end-to-end relation extraction is defined as follows:
Input: A natural language sentence S
Output: i) Entity Extraction: List of entity mentions occurring in S. Here, each entity mention is identified in terms of its boundaries and entity type. ii) Relation Extraction: List of pairs of entity mentions for which any pre-defined semantic relation holds.

E.g., S = `Paris, John's sister, is staying in New York.` Here, the expected output of an end-to-end relation extraction system is shown in the table 1.

Table 1: Expected output of end-to-end relation extraction system (For definitions of entity and relation types, see Section 7.1)

Entity Extraction	Relation Extraction
`Paris` : PER	⟨ `Paris`, `John` ⟩ : PER-SOC
`John` : PER	⟨ `John`, `sister` ⟩ : PER-SOC
`sister` : PER	⟨ `Paris`, `New York` ⟩ : PHYS
`New York` : GPE	⟨ `sister`, `New York` ⟩ : PHYS

3 Motivating Example

Any particular semantic relation generally holds between entity mentions of some specific entity types. E.g., social (PER-SOC) relation holds between two persons (PER); employee-employer (EMP-ORG) relation holds between a person (PER) and an organization (ORG) or a geo-political entity (GPE). Hence, information about entity types certainly helps relation extraction. Traditional "pipeline" approaches for relation extraction approaches use features based on entity types. However, in these "pipeline"

approaches there is no bidirectional flow of information; i.e., entity extraction sub-task does not utilize any knowledge / features based on relation information. When entity and relation extraction are jointly addressed, such bidirectional flow is possible. Thus improving performance of both the entity extraction and the relation extraction.

Consider an example sentence: `Paris, John's sister, is staying in New York.` Most of the state-of-the-art Named Entity Recognition (NER) tools incorrectly identify `Paris` as a mention of type LOC[3] and not as PER. Here, if an entity extraction algorithm has some evidence that `Paris` is involved in a social (PER-SOC) relation, then it would prefer to label `Paris` as a PER than as a LOC. This is because social relation is only possible between two persons. Thus, the information about relations helps in determining entity types of entity mentions. This is the motivation behind designing algorithms which jointly extract entities and relations.

4 Overview of Techniques

Various techniques have been proposed for jointly extracting entities and relations since 2002. Table 2 summarizes most of the techniques from the literature of joint extraction. We visualize each of these techniques from two aspects of joint extraction: *joint model* and *joint inference*. Most of the techniques exploit either one of these aspects. But some recent techniques have exploited both of these aspects.

Here, by *joint model*, we mean that a single model is learned for both the tasks of entity and relation extraction. For example, a single joint neural network model can be learned and both the tasks of entity and relation extraction share the same parameters. Overall, joint models can be of various types as shown in Table 2. Moreover, by *joint inference*, we mean that the decision about entity and relation labels is taken jointly at a global level (usually a sentence). Here, there may be separate underlying local models

[3] E.g., Stanford CoreNLP 3.9.1 NER identifies `Paris` as a city name

Table 2: Overview of various techniques for joint extraction of entities and relations

Approach	Joint Model	Joint Inference	Model Type	Inference Technique	Evaluation		
					ACE'04	ACE'05	CoNLL'04
Roth and Yih [21]	✗	✓		Belief Network	✗	✗	✗
Roth and Yih [22]	✗	✓		ILP	✗	✗	✓
Kate and Mooney [8]	✗	✓		Parsing	✗	✗	✓
Chan and Roth [3]	✗	✓		Rules	✓	✗	✗
Li and Ji [11]	✓	✓	Structured Prediction	Beam search	✓	✓	✗
Miwa and Sasaki [13]	✓	✓	Table+Structured Prediction	Beam search	✗	✗	✓
Gupta et al. [5]	✓	✗	Neural (RNN)		✗	✗	✓
Pawar et al. [14]	✗	✓		MLN	✓	✗	✗
Miwa and Bansal [12]	✓	✗	Neural (Bi/tree-LSTM)		✓	✓	✗
Pawar et al. [15]	✓	✓	Table+Neural (NN, LSTM)	MLN	✓	✗	✗
Katiyar and Cardie [9]	✓	✗	Neural (Bi-LSTM)		✓	✓	✗
Ren et al. [20]	✓	✗	Embedded Representations		✗	✗	✗
Zheng et al. [26]	✓	✓	Neural (Bi-LSTM)	Joint Label	✗	✗	✗
Zhang et al. [25]	✓	✓	Table+Neural (Bi-LSTM)	Global optimization	✗	✓	✓
Bekoulis et al. [1]	✓	✓	CRF, Neural (Bi-LSTM)	Parsing	✓	✗	✓
Wang et al. [24]	✓	✓	Neural (Bi-LSTM)	Parsing	✗	✗	✗
Li et al. [10]	✓	✗	Neural (Bi-LSTM)		✗	✗	✗

for entity and relation extraction. Overall, there are several joint inference / decoding techniques which are shown in Table 2.

5 Joint Inference Techniques

Here, we describe a few techniques used for joint inference:
Integer Linear Programming (ILP): Here, a *global* decision is taken by using Integer Linear Programming which is consistent with some domain constraints . This approach was proposed by Roth and Yih [22]. They first learn independent *local* classifiers for entity and relation extraction. During inference, given a sentence, a global decision is produced such that the domain-specific or task-specific constraints are satisfied. Often these constraints capture mutual compatibility of entity and relation types. A simple example of such constraints is: both the arguments of the PER-SOC relation should be PER. Consider our example sentence – `Paris, John's sister, is staying in New York.` Here, the entity extractor identifies two mentions `John` and `Paris` and also predicts entity types for these mentions. For `John`, let the predicted probabilities be : Pr(PER) = 0.99 and Pr(ORG) = 0.01. For `Paris`, let the predicted probabilities be : Pr(GPE) = 0.75 and Pr(PER) = 0.25. Also, the relation extractor identifies the relation PER-SOC between the two mentions. If we accept the best suggestions given by the local classifiers, then the global prediction is that the relation PER-SOC exists between the PER mention `John` and the GPE mention `Paris`. But this violates the domain-constraint mentioned earlier. Hence the global decision which satisfies all the specified constraints would be to label both the mentions as PER and mark the PER-SOC relation between them.
Markov Logic Networks (MLN): Similar to ILP, MLN provides another framework for taking a global decision consistent with the domain constraints. MLN combines first order logic with probability. The domain rules or domain knowledge is represented in an MLN using weighted first order logic rules. Stronger the belief about any rule, higher is its associated weight. Inference in such an MLN gives the most probable true groundings of certain (query) predicates, while ensuring maximum weighted satisfiability of the rules. Pawar et al. [14, 15], use MLN for joint inference for extracting entities and relations. As compared to ILP, MLN provides better representability in the form of first order logic rules. For example, the above-mentioned rule "both the arguments of the PER-SOC relation should be PER" can be written as:

$$\text{PER-SOC}(x, y) \Rightarrow \text{PER}(x) \wedge \text{PER}(y)$$

In addition to the rules for ensuring compatibility of entity and relation types, MLN can easily represent other complex domain knowledge. For example, the rule "a person can be employed at only one organization at a time" can be written as:

$$\text{EMP-ORG}(x, y) \wedge \text{PER}(x) \wedge \text{ORG}(y) \wedge \text{ORG}(z) \wedge (y \neq z) \Rightarrow \neg\text{EMP-ORG}(x, z)$$

Joint Label: Zheng et al. [26] proposed a novel tagging scheme for joint extraction of entities and relations. This tagging scheme reduces the joint extraction task to a tagging problem. Intuitively, a single tag is assigned to a word which encodes entity as well as relation label information, automatically leading to joint inference. Figure 1 de-

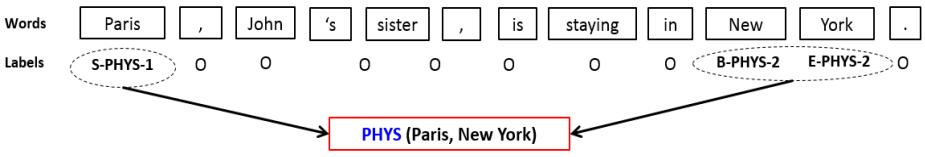

Fig. 1: The proposed new tagging scheme for an relation instance PHYS (`Paris`, `New York`) in our example sentence.

picts an example sentence and its annotations as per the proposed new tagging scheme. The tag "O" represents the "Other" tag, which means that the corresponding word is not part of the expected relation tuples. The other tags consist of three parts: i) the word position in the entity, ii) the relation type, and iii) the relation role (argument number). The **BIES** (**B**egin, **I**nside, **E**nd, **S**ingle) encoding scheme is used for marking entity boundaries. The relation type information is obtained from a predefined set of relations and the relation role information is represented by the numbers 1 and 2. Let $Entity_1$ and $Entity_2$ be the first and second entity arguments of a relation type RT, respectively. Words in $Entity_1$ are marked with the relation role 1 for RT. Similarly, words in $Entity_2$ are marked with the relation role 2. Hence, the total number of tags is $2 \times 4 \times NumRelationTypes + 1$. Here, the multiplier 4 represents the entity boundary tags **BIES** and other multiplier 2 represents two entity arguments for each relation type. For example shown in Figure 1, the words `New` and `York` are part of the entity mention which is the second argument of PHYS relation and hence are marked with the tags B-PHYS-2 and E-PHYS-2, respectively. One limitation of this approach is that currently it can not model the scenario where a single entity mention is involved in multiple relations with multiple other entity mentions. Hence, the other relation PER-SOC (`Paris`, `John`) in which `Paris` is involved, can not be handled.

Beam Search: Li and Ji [11] proposed an approach for *incremental* joint extraction of entities and relations. They formulated the joint extraction task as a structured prediction problem to reveal the linguistic and logical properties of the hidden structures. Here, the output structure of each sentence was interpreted as a graph in which entity mentions are nodes and relations are directed arcs labelled with relation types. They designed several local as well as global features to characterize and score these structures. Hence, the joint extraction problem was reduced to predicting a structure with the highest score for any given sentence. They proposed a joint decoding / inference approach for this structured prediction task using beam-search. Intuitively, at the i^{th} token, k best partial assignments / structures are maintained and extended further. Similarly, beam-search based inference was also employed by Miwa and Sasaki [13] where the output structure for a sentence was a table representation.

Parsing: Kate and Mooney [8] proposed a parsing based approach which uses a graph called as *card-pyramid*. The graph is so called because it encodes mutual dependencies among the entities and relations in a graph structure which resembles pyramid constructed using playing cards. This is a tree-like graph which has one root at the highest level, internal nodes at intermediate levels and leaves at the lowest level. Each entity in the sentence correspond to one leaf and if there are n such leaves then the graph has n

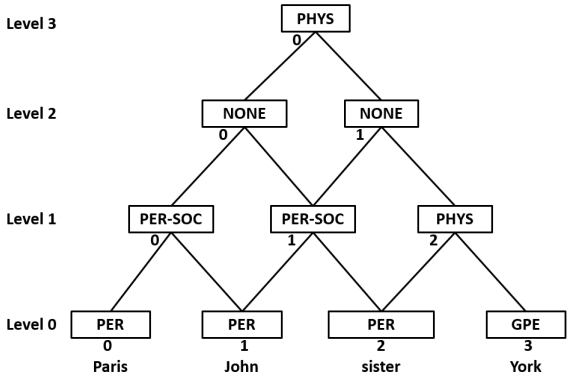

Fig. 2: Card-pyramid graph for the sentence `Paris, John's sister, is staying in New York.`

levels. Each level l contains one less node than the number of nodes in the $(l-1)$ level. The node at position i in level l is parent of nodes at positions i and $(i+1)$ in the level $(l-1)$. Each node in the higher layers (i.e. layers except the lowest layer), corresponds to a possible relation between the leftmost and rightmost nodes under it in the lowest layer. Figure 2 shows this *card-pyramid* graph for an example sentence. To jointly label the nodes in the card-pyramid graph, the authors propose a parsing algorithm analogous to the bottom-up CYK parsing algorithm for Context Free Grammar (CFG) parsing. The grammar required for this new parsing algorithm is called Card-pyramid grammar and its consists of following production types:

- Entity Productions: These are of the form $EntityType \rightarrow Entity$, e.g. PER$\rightarrow$`John`. Similar to the ILP based approach, a local entity classifier is trained to compute the probability that entity in the RHS being of the type given in the LHS of the production.
- Relation Productions: These are of the form $RelationType \rightarrow EntityType1$ $EntityType2$, e.g. PHYS$\rightarrow$PER GPE. A local relation classifier is trained to obtain the probability that the two entities in the RHS are related by the type given in the LHS of the production.

Given the entities in a sentence, the Card-pyramid grammar, and the local entity and relation classifiers, the card-pyramid parsing algorithm attempts to find the most probable labelling of all of its nodes which corresponds the entity and relation types. One limitation of this approach is that only entity type identification happens jointly with relation classification, i.e. boundary detection of entity mentions should be done as a pre-processing step and does not happen jointly. Recently, Bekoulis et al. [1] and Wang et al. [24] proposed joint extraction techniques which use dependency parsing like approaches for joint inference. Also, they allow multiple heads for a node (word) to represent participation in multiple relations simultaneously with other nodes.

6 Joint Models

Here, we describe a few joint models which have been employed for joint extraction of entities and relations:

Structured Prediction: In most of the earlier approaches for joint extraction of entities and relations, it was assumed that the boundaries of the entity mentions are known. Li and Ji [11] presented an incremental joint framework for simultaneous extraction of entity mentions and relations, which also incorporates the problem of boundary detection for entity mentions. The authors proposed to formulate the problem of joint extraction of entities and relations as a structured prediction problem. They aimed to predict the output structure ($y \in Y$) for a given sentence ($x \in X$), where this structure can be viewed as a graph modelling entity mentions as nodes and relations as directed arcs with relation types as labels. Following linear model is used to predict the most probable structure y' for x where $f(x, y)$ is the feature vector that characterizes the entire structure.

$$y' = \arg\max_{y \in Y(x)} f(x, y) \cdot W$$

The score of each candidate assignment is defined as the inner product of the feature vector $f(x, y)$ and feature weights W. The number of all possible structures for any given sentence can be very large and there does not exist a polynomial-time algorithm to find the best structure. Hence, they apply beam-search to expand partial configurations for the input sentence incrementally to find the structure with the highest score.

Neural Models: Here, predictions for both the tasks of entity and relation extraction are carried out using a single joint neural model, where at least some of the model parameters are shared across both the tasks. Joint modelling is realized through such parameter sharing where training for any task updates the parameters involved in both the tasks. Miwa and Bansal [12] presented a neural model for capturing both word sequence and dependency tree substructure information by stacking bidirectional tree-structured LSTMs (tree-LSTM) on bidirectional sequential LSTMs (Bi-LSTM). Their model jointly represents both entities and relations with shared parameters in a single model. The overview of the model is illustrated in the Figure 3. It consists of three representation layers: i) a word embeddings layer, ii) a word sequence based LSTM-RNN layer (sequence layer), and iii) a dependency subtree based LSTM-RNN layer (dependency layer). While decoding, entities are detected in greedy, left-to-right manner on the sequence layer. And relation classification is carried out on the dependency layers, where each subtree based LSTM-RNN corresponds to a relation candidate between two detected entities. After decoding the entire model structure, the parameters are updated simultaneously via backpropagation through time (BPTT). The dependency layers are stacked on the sequence layer, so the embedding and sequence layers are shared by both entity detection and relation classification, and the shared parameters are affected by both entity and relation labels. This is the first joint neural model which motivated several other joint models [9, 26]. This model was adopted for Biomedical domain by Li et al. [10]. In addition, they use Convolutional Neural Networks (CNN) for extracting morphological information (like prefix or suffix) from characters of words. Then each word in a sentence is represented by a concatenated vector of its word embeddings, POS embeddings and character-level representation by CNN. Character-level

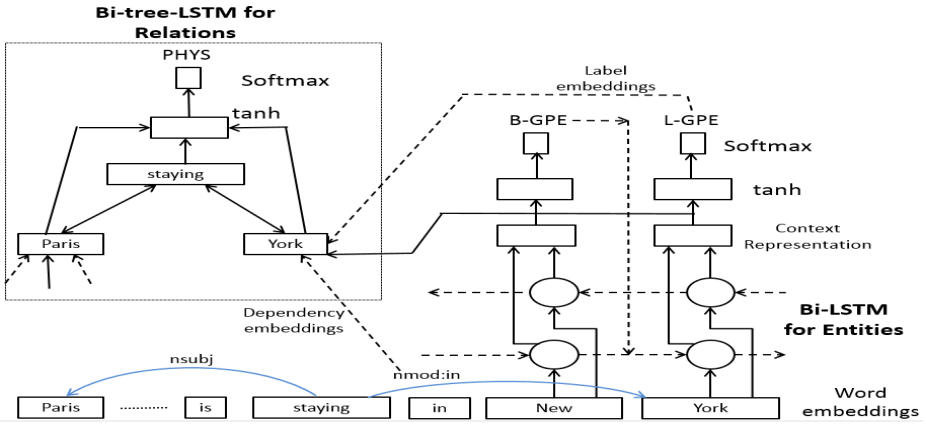

Fig. 3: End-to-end relation extraction model, with bidirectional sequential and bidirectional tree-structured LSTM-RNNs.

information is more useful in Biomedical domain because several biological entities share morphological or orthographic features, e.g., bacteria names `helicobacter` and `campylobacter` share the suffix `bacter`.

Table Representation: Another idea for jointly modelling entity and relation extraction tasks is *Table Representation* or *Table Filling*. It was first proposed by Miwa and Sasaki [13]. Here, a table is associated with each sentence where every table cell is labelled with an appropriate label so that the whole entities and relations structure in a sentence is represented in a single table. Table 3 depicts this table representation for an example sentence. The diagonal cells of the table represent the entity labels which capture both boundary and type information with the help of BILOU (**B**egin, **I**nside, **L**ast, **O**utside, **U**nit) or BIO encoding. E.g., in Table 3, the word `New` gets the label `B-GPE` as it is the first word of the complete entity mention `New York`. The off-diagonal cells represent relation labels. Here, relations between entity mentions are mapped to relations between the last words of the mentions. E.g., the PHYS relation between `sister` and `New York` is assigned to the cell corresponding to `sister` and `York`. $\perp$ represents no pre-defined relation between the corresponding words. As the table is symmetric, only upper or lower triangular part of the table needs to be labelled. Miwa and Sasaki [13] approach this table filling problem using structure learning approach similar to Li and Ji [11]. They define a scoring function to evaluate a possible label assignment to a table and build a model which predicts the most probable label assignment for a table which maximizes the scoring function. During inference, beam search is used which assigns labels to cells one by one and keeps the top K best assignments when moving from a cell to the next cell. Finally, it returns the best assignment when labels are assigned to all the cells. The authors propose various strategies to arrange the cells in two dimensions to a linear order. They also integrate various label dependencies into the scoring function to avoid illegal label assignments. E.g., cell corresponding to the i^{th} and j^{th} words should never be assigned any valid relation label if any of the words are labelled with entity label O.

Table 3: Table Representation for an example sentence

	Paris	,	John	's	sister	,	is	staying	in	New	York	.
Paris	U-PER	⊥	PER-SOC	⊥	⊥	⊥	⊥	⊥	⊥	⊥	PHYS	⊥
,		O	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
John			U-PER	⊥	PER-SOC	⊥	⊥	⊥	⊥	⊥	⊥	⊥
's				O	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
sister					U-PER	⊥	⊥	⊥	⊥	⊥	PHYS	⊥
,						O	⊥	⊥	⊥	⊥	⊥	⊥
is							O	⊥	⊥	⊥	⊥	⊥
staying								O	⊥	⊥	⊥	⊥
in									O	⊥	⊥	⊥
New										B-GPE	⊥	⊥
York											L-GPE	⊥
.												O

The table representation idea has further motivated several other joint extraction approaches. Pawar et al. [15] use a similar table representation but instead of using BILOU encoding to represent entity boundaries, they introduced a new relation WEM (Within Entity Mention) between head word[4] of an entity mention and other words in the same entity mention. E.g., they would assign entity labels O and GPE to the words New and York, respectively and assign relation label WEM to the cell corresponding to New and York. Further, they train a neural network based model to predict an appropriate label for each cell in the table. They also employ Markov Logic Networks (MLN) based inference at a sentence level to incorporate various dependencies among entity and relation labels. Other recent approaches proposed by Zhang et al. [25] and Gupta et al. [5] build upon the same table representation idea and use Recursive Neural Networks (RNN) and Long Short-Term Memory (LSTM) based models.

7 Experimental Evaluation

In this section, we describe some of the most widely used datasets for end-to-end relation extraction and summarize the reported results on those datasets. We also describe the evaluation methodology and other experimental analysis.

7.1 Datasets

ACE 2004: It is the most widely used dataset in the relation extraction literature and is available from Linguistic Data Consortium (LDC) as catalogue LDC2005T09. It annotates both entity and relation types information in an XML like format. It identifies 7 entity types[5]: (i) PER (person), (ii) ORG (organization), (iii) LOC (location), (iv) GPE

[4] Head word is generally the last word of noun phrase entities but not always. E.g., for Bank of America, the head word is Bank. Head word is that word of an entity mention through which the mention is linked to the rest of the sentence in its dependency tree.

[5] www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-edt-v4.2.6.pdf

(geo-political entity), (v) FAC (facility), (vi) VEH (vehicle), and (vii) WEA (weapon). Additionally, it identifies 22 fine-grained relation types which are grouped into 6 coarse-grained relation types[6]: (i) EMP-ORG (employee-organization or subsidiary relationships), (ii) GPE-AFF (affiliations of PER/ORG to an GPE entity), (iii) PER-SOC (social relationships between two PER entities), (iV) ART (agent-artifact relationship), (v) PHYS (physical / located at), (vi) OTHER-AFF (other PER/ORG affiliations). Chan and Roth [3] used this dataset for the first time for evaluating end-to-end relation extraction. They ignored the original DISC (discourse) relation as it was only for the purpose of the discourse. They used only news wire and broadcast news subsections of this dataset which consists of 345 documents and 4011 positive relation instances. All the later approaches followed the same methodology for producing comparable results.

ACE 2005: This dataset [23] is also available from LDC as catalogue LDC2006T06. It annotates the same entity types a that of ACE 2004. ACE 2005 also kept the relation types PER-SOC, ART and GPE-AFF of ACE 2004, but it split PHYS into two relation types PHYS and a new relation type PART-WHOLE. The DISC relation type was removed, and the relation type OTHER-AFF was merged into EMP-ORG. It was observed that ACE 2005 improved on both annotation quality and relation type definition, as compared to ACE 2004. Li and Ji [11] used this dataset for the first time for evaluating end-to-end relation extraction. Ignoring two small subsets (*cts* and *un*) from informal genres, they selected the remaining 511 documents. These were randomly split into 3 parts: (i) training (351), (ii) development (80), and (iii) blind test set (80). All the later approaches followed the same methodology for producing comparable results.

7.2 Evaluation of End-to-End Relation Extraction

As discussed earlier, the end-to-end relation extraction system is expected to identify: (i) boundaries of entity mentions, (ii) entity types of these entity mentions, and (iii) relation type (if any) for each pair of entity mentions. Hence, evaluation of end-to-end relation extraction is often done at 2 levels:

1. **Entity extraction:** Here, only entity extraction performance is evaluated. Two entity mentions are said to be *matching* if both have same boundaries (i.e. contain exactly the same sequence of words) and same entity type. Any predicted entity mention is counted as a true positive (TP) if it matches with any of the gold-standard entity mentions in the same sentence, otherwise it is counted as a false positive (FP). Both TP or FP are counted for the predicted entity type. Similarly, for each gold-standard entity mention, if there no matching predicted entity mention in the same sentence, then a false negative (FN) is counted for the gold-standard entity type. For each entity type, using precision, recall and F1 are computed using its TP, FP and FN counts. F1-scores across all entity types are micro-averaged for computing overall entity extraction performance.

[6] `www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-rdc-v4.3.2.PDF`

Table 4: Performance of various approaches on the ACE 2004 dataset. The numbers are micro-averaged and obtained after 5-fold cross-validation. Actual folds used by each approach may differ.

Approach	**Entity Extraction**			**Entity+Relation Extraction**		
	P	**R**	**F**	**P**	**R**	**F**
Pipeline [11]	81.5	74.1	77.6	58.4	33.9	42.9
Chan and Roth [3]				42.9	38.9	40.8
Li and Ji [11]	83.5	76.2	79.7	60.8	36.1	45.3
Pawar et al. [14]	79.0	80.1	79.5	52.4	41.3	46.2
Miwa and Bansal [12]	80.8	82.9	**81.8**	48.7	48.1	48.4
Pawar et al. [15]	81.2	79.7	80.5	56.7	44.5	**49.9**
Katiyar and Cardie [9]	81.2	78.1	79.6	46.4	45.3	45.7
Bekoulis et al. [1]	81.0	81.3	81.2	50.1	44.5	47.1

2. **Entity+Relation extraction:** Here, end-to-end relation extraction performance is evaluated. Any predicted or gold-standard relation mention consists of a pair of entity mentions along with their entity types, and an associated relation type. Hence, two relation mentions are said to be *matching* only if both the entity mentions match and associated relation types are same. Each gold-standard relation mention is counted as a TP if there is a *matching* predicted entity mention, otherwise it is counted as FN. Similarly, each predicted relation mention is counted as an FP unless there is any *matching* gold-standard relation mention. For each relation type, precision, recall and F1 are computed using its TP, FP and FN counts. F1-scores across all relation types are micro-averaged for computing overall entity extraction performance.

Analysis of Results: Tables 4 and 5 show the results of various approaches on the ACE 2004 and ACE 2005 datasets, respectively. The F1-scores still below 60% indicate how challenging the task of end-to-end relation extraction is. Li and Ji [11] carried out an interesting experiment where two human annotators were asked to perform end-to-end relation extraction manually on the ACE 2005 test dataset. The human F1-score for

Table 5: Performance of various approaches on the ACE 2005 dataset. The numbers are micro-averaged and obtained on a test split of 80 documents. The (_) performance numbers are not reported in the original paper.

Approach	**Entity Extraction**			**Entity+Relation Extraction**		
	P	**R**	**F**	**P**	**R**	**F**
Pipeline [11]	83.2	73.6	78.1	65.1	38.1	48.0
Li and Ji [11]	85.2	76.9	80.8	65.4	39.8	49.5
Miwa and Bansal [12]	82.9	83.9	83.4	57.2	54.0	55.6
Katiyar and Cardie [9]	84.0	81.3	82.6	55.5	51.8	53.6
Zhang et al. [25]	_	_	**83.6**	_	_	**57.5**

Table 6: Example sentences from the ACE 2004 dataset illustrating how the joint extraction of entities and relations helps in determining entity type of **its**. Entity mentions of interest are highlighted in **bold**

S1	U.S. District Court Judge Murray Schwartz in Wilmington, Del., ruled that Camelot Music could not deduct interest on loans it took out against life insurance on **its** 1,430 **employees** in 1990 through 1993.
	EntityType (its) = ORG, EntityType (employees) = PER, RelType (⟨its, employees⟩) = EMP-ORG
S2	our choice is the choice of permanent, comprehensive and just peace, and our aim is to liberate our land and to create our independence state in palestinian blast land with jerusalem as **its capital** and the return of our refugees to their homes.
	EntityType (its) = GPE, EntityType (capital) = GPE, RelType (⟨its, capital⟩) = PHYS

this task was observed to be around 70%. Moreover, F1-score of the inter-annotator agreement (the entity / relation extractions where both the annotators agreed) was only about 51.9%. This analysis clearly establishes the high difficulty level of the task.

Another important aspect of the ACE datasets to note is the nature of its entity mentions. Overall, three types of entity mentions are annotated in the ACE datasets: (i) name mentions (generally proper nouns, e.g. John, United States), (ii) nominal mentions (generally common nouns, e.g. guy, employee), and (iii) pronoun mentions (e.g. he, they, it). Unlike the traditional Named Entity Recognition (NER) task which extracts only the name mentions, the ACE entity extraction task focusses on extracting all the three types of mentions. This makes it more challenging task yielding lower accuracies. Especially for pronoun mentions like its in the example sentences in Table 6, determining the entity type is more challenging. This is because, the mention its is observed both as ORG or as GPE in the training data depending on the context. In the sentence S1 in Table 6, the knowledge that its is related to employees through the EMP-ORG relation, helps in labelling its as ORG. Similarly, in the sentence S2, the knowledge that its is related to capital through the PHYS relation, helps in labelling its as GPE. Hence, these examples illustrate that unlike pipeline methods, in joint extraction methods, both the tasks of entity extraction and relation extraction help each other.

7.3 Domain-specific Entities and Relations

Except Li et al. [10], all other joint extraction approaches in Table 2 are evaluated on *general* domain datasets like ACE 2004 or ACE 2005. There is no previous study on how well the approaches designed for general domain work for domain-specific entities and relations. In this section, we present the results of our experiments where we apply a general domain technique on a Biomedical dataset. As a representative general domain approach, we choose Pawar et al. [15] which is the best performing approach on the ACE 2004 dataset.

Table 7: Performance of various approaches on the ADE dataset. The numbers are micro-averaged and obtained using 10-fold cross-validation. Actual folds used by each approach may differ.

Approach	**Entity Extraction**			**Entity+Relation Extraction**		
	P	**R**	**F**	**P**	**R**	**F**
Li et al. [10]	82.7	86.7	84.6	67.5	75.8	**71.4**
Pawar et al. [15] (GloVe vectors)	80.0	82.4	81.2	65.8	66.6	66.2
Pawar et al. [15] (PubMed vectors)	82.1	84.0	83.0	68.5	68.0	68.2
Pawar et al. [15] (GloVe vectors, Lenient)	82.8	85.2	84.0	70.6	71.3	70.9
Pawar et al. [15] (PubMed vectors, Lenient)	85.0	86.8	85.9	73.0	73.7	73.3

Li et al. [10] evaluate their end-to-end relation extraction approach on the **Adverse Drug Event (ADE)** dataset [6]. This dataset contains sentences from PubMed abstracts annotated with entity types DRUG, ADVERSE_EVENT and DOSAGE. It also contains annotations for two relation types: (i) DRUG-AE between a DRUG and an ADVERSE_EVENT it causes, and (ii) DRUG-DOSAGE between a DRUG and its DOSAGE. Li et al. [10] evaluated their model only on a subset of the ADE dataset containing sentences with at least one instance of the DRUG-AE relation. They also ignored 120 relation instances containing nested gold annotations, e.g., `lithium intoxication`, where `lithium` causes `lithium intoxication`. We also followed the same methodology for creating a dataset for our experiments. We ended up with a dataset of 4228 distinct sentences[7] containing 6714 relation instances. Following is an example sentence and annotations from this dataset: `After infliximab treatment, additional sleep studies revealed an increase in the number of apneic events and SaO2 dips suggesting that TNFalpha plays an important role in the pathophysiology of sleep apnea.` There are two annotated relation instances of DRUG-AE for this sentence: (i) ⟨`infliximab`, `increase in the number of apneic events`⟩, and (ii) ⟨`infliximab`, `SaO2 dips`⟩.

Analysis of Results: Table 7 shows the results of both the methods (Li et al. [10] and Pawar et al. [15]) on the ADE dataset for end-to-end extraction of the `DRUG-AE` relation. Li et al. used 300 dim word embeddings pre-trained on PubMed corpus [18]. For Pawar et al., we experimented with two types of word embeddings: 100-dim GloVe embeddings trained on Wikipedia corpus [17] (as reported in the original paper) as well as 300-dim embeddings trained on PubMed corpus [18]. As the ADE dataset is also derived from PubMed abstracts, the PubMed word embeddings perform better than GloVe embeddings. Even though it is designed for the general domain, Pawar et al. [15] produces comparable results with respect to Li et al. [10].

[7] Li et al. [10] mentions number of sentences in their dataset to be 6821 which seems to be a typo because the original paper [6] for ADE dataset mentions that there are only 4272 sentences containing at least one drug-related adverse effect mention. After ignoring the 120 relation instances of nested annotations, this number comes down to 4228 in our dataset.

Upon detailed analysis of errors, we found that the major source of errors was incorrect boundary detection for entities of type ADVERSE_EVENT. As compared to ACE datasets, the entities in the ADE dataset can have more complex syntactic structures. E.g., it is very rare in case of the ACE entities to be noun phrases (NP) subsuming prepositional phrases (PP), but in the ADE dataset, we frequently encounter entities like `increase in the number of apneic events`. We also observed that the boundary annotations for the ADVERSE_EVENT entities are inconsistent. E.g., the complete phrase `severe mucositis` is annotated as an ADVERSE_EVENT but in case of `Severe rhabdomyolysis`, only `rhabdomyolysis` is annotated as an ADVERSE_EVENT. Hence, we carried out a lenient version of evaluation where a predicted ADVERSE_EVENT $AE_{predicted}$ is considered to be matching any gold-standard ADVERSE_EVENT AE_{gold} if $AE_{predicted}$ contains AE_{gold} as a prefix or suffix and $AE_{predicted}$ has at most one extra word as compared to AE_{gold}. E.g., even if AE_{gold} = `rhabdomyolysis` and $AE_{predicted}$ = `Severe rhabdomyolysis`, we consider both of them to be matching. But if AE_{gold} = `severe mucositis` and $AE_{predicted}$ = `mucositis`, we do not consider them to be matching because the predicted mention is missing a word which is expected as per the gold mention. This lenient evaluation leads to a much better performance as shown in Table 7.

8 Conclusion

In this paper, we surveyed various techniques for jointly extracting entities and relations. We first motivated the need for developing joint extraction techniques as opposed to traditional "pipeline" approaches. We then summarized more than a decade's work in joint extraction of entities and relations in the form of a table. In that table, we categorized techniques based on the approach they adopt for joint extraction, i.e. whether they employ joint inference or joint modelling or both. We further described some of the representative techniques for joint inference and joint modelling. We also described standard datasets and evaluation techniques; and summarized performance of the joint extraction approaches on these datasets. We presented a brief analysis of application of a general domain joint extraction approach on the ADE dataset from Biomedical domain. We believe that this survey would be useful for researchers as well as practitioners in the field of Information Extraction. Also, these joint extraction techniques would motivate new techniques even for other NLP tasks such as Semantic Role Labelling (SRL) where predicates and arguments can be extracted jointly.

References

1. Bekoulis, G., Deleu, J., Demeester, T., Develder, C.: Joint entity recognition and relation extraction as a multi-head selection problem. arXiv preprint arXiv:1804.07847 (2018)
2. Bunescu, R., Mooney, R.: A shortest path dependency kernel for relation extraction. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. pp. 724–731. Association for Computational Linguistics, Vancouver, British Columbia, Canada (October 2005), `http://www.aclweb.org/anthology/H/H05/H05-1091`

3. Chan, Y.S., Roth, D.: Exploiting syntactico-semantic structures for relation extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 551–560. Association for Computational Linguistics, Portland, Oregon, USA (June 2011), http://www.aclweb.org/anthology/P11-1056
4. Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S., Weischedel, R.M.: The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In: LREC. vol. 2, p. 1 (2004)
5. Gupta, P., Schütze, H., Andrassy, B.: Table filling multi-task recurrent neural network for joint entity and relation extraction. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2537–2547 (2016)
6. Gurulingappa, H., Rajput, A.M., Roberts, A., Fluck, J., Hofmann-Apitius, M., Toldo, L.: Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. Journal of biomedical informatics **45**(5), 885–892 (2012)
7. Jiang, J., Zhai, C.: A systematic exploration of the feature space for relation extraction. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference. pp. 113–120. Association for Computational Linguistics, Rochester, New York (April 2007), http://www.aclweb.org/anthology/N/N07/N07-1015
8. Kate, R.J., Mooney, R.: Joint entity and relation extraction using card-pyramid parsing. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning. pp. 203–212. Association for Computational Linguistics, Uppsala, Sweden (July 2010), http://www.aclweb.org/anthology/W10-2924
9. Katiyar, A., Cardie, C.: Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 917–928 (2017)
10. Li, F., Zhang, M., Fu, G., Ji, D.: A neural joint model for entity and relation extraction from biomedical text. BMC bioinformatics **18**(1), 198 (2017)
11. Li, Q., Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 402–412. Association for Computational Linguistics, Baltimore, Maryland (June 2014), http://www.aclweb.org/anthology/P14-1038
12. Miwa, M., Bansal, M.: End-to-end relation extraction using lstms on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1105–1116. Association for Computational Linguistics, Berlin, Germany (August 2016), http://www.aclweb.org/anthology/P16-1105
13. Miwa, M., Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1858–1869. Association for Computational Linguistics, Doha, Qatar (October 2014), http://www.aclweb.org/anthology/D14-1200
14. Pawar, S., Bhattacharyya, P., Palshikar, G.: End-to-end relation extraction using markov logic networks. In: Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2016), LNCS 9624. Springer (2016)
15. Pawar, S., Bhattacharyya, P., Palshikar, G.: End-to-end Relation Extraction using Neural Networks and Markov Logic Networks. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. vol. 1, pp. 818–827 (2017)
16. Pawar, S., Palshikar, G.K., Bhattacharyya, P.: Relation extraction: A survey. arXiv preprint arXiv:1712.05191 (2017)

17. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
18. Pyysalo, S., Ginter, F., Moen, H., Ananiadou, S.: Distributional semantics resources for biomedical text processing. LBM 2013
19. Qian, L., Zhou, G., Kong, F., Zhu, Q., Qian, P.: Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). pp. 697–704. Coling 2008 Organizing Committee, Manchester, UK (August 2008), http://www.aclweb.org/anthology/C08-1088
20. Ren, X., Wu, Z., He, W., Qu, M., Voss, C.R., Ji, H., Abdelzaher, T.F., Han, J.: Cotype: Joint extraction of typed entities and relations with knowledge bases. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1015–1024. International World Wide Web Conferences Steering Committee (2017)
21. Roth, D., Yih, W.t.: Probabilistic reasoning for entity & relation recognition. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7. ACL (2002)
22. Roth, D., Yih, W.t.: A linear programming formulation for global inference in natural language tasks. In: Ng, H.T., Riloff, E. (eds.) HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004). pp. 1–8. Association for Computational Linguistics, Boston, Massachusetts, USA (May 6 - May 7 2004)
23. Walker, C., Strassel, S., Medero, J., Maeda, K.: Ace 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia **57** (2006)
24. Wang, S., Zhang, Y., Che, W., Liu, T.: Joint extraction of entities and relations based on a novel graph scheme. In: IJCAI. pp. 4461–4467 (2018)
25. Zhang, M., Zhang, Y., Fu, G.: End-to-end neural relation extraction with global optimization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1730–1740 (2017)
26. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1227–1236 (2017)
27. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). pp. 427–434. Association for Computational Linguistics, Ann Arbor, Michigan (June 2005). https://doi.org/10.3115/1219840.1219893, http://www.aclweb.org/anthology/P05-1053