Relation Extraction : A Survey

Sachin Pawar[a,b], Girish K. Palshikar[a], Pushpak Bhattacharyya[b]

[a] *TCS Research, Tata Consultancy Services Ltd.*
[b] *Department of CSE, Indian Institute of Technology Bombay*

Abstract

With the advent of the Internet, large amount of digital text is generated everyday in the form of news articles, research publications, blogs, question answering forums and social media. It is important to develop techniques for extracting information automatically from these documents, as lot of important information is hidden within them. This extracted information can be used to improve access and management of knowledge hidden in large text corpora. Several applications such as Question Answering, Information Retrieval would benefit from this information. Entities like persons and organizations, form the most basic unit of the information. Occurrences of entities in a sentence are often linked through well-defined relations; e.g., occurrences of person and organization in a sentence may be linked through relations such as *employed_at*. The task of Relation Extraction (RE) is to identify such relations automatically. In this paper, we survey several important supervised, semi-supervised and unsupervised RE techniques. We also cover the paradigms of Open Information Extraction (OIE) and Distant Supervision. Finally, we describe some of the recent trends in the RE techniques and possible future research directions. This survey would be useful for three kinds of readers - i) Newcomers in the field who want to quickly learn about RE; ii) Researchers who want to know how the various RE techniques evolved over time and what are possible future research directions and iii) Practitioners who just need to know which RE technique works best in various settings.

Keywords: Relation Extraction, Supervised Learning, Kernel Methods, Unsupervised Learning, Semi-supervised Learning, Open Information Extraction, Distant Supervision

1. Introduction

It is well-known that a lot of tacit and experiential knowledge is present in document repositories (e.g., reports, emails, resumes, papers, proposals, blogs

Email address: sachinpawar@cse.iitb.ac.in (Sachin Pawar)

etc.) that are created and maintained within an enterprise or across the Web. Extracting this knowledge, disseminating it when needed and reusing it to improve decision-making and operational efficiency of practical tasks are important. The essential goal of *information extraction (IE)* [79], [109] is to extract a specific kind of information from a given document repository and output it to a structured repository such as a relational table or an XML file. IE is an important problem that involves natural language processing, computational linguistics and text mining. IE is an useful first step in a wide range of knowledge management systems. In addition, IE is also useful in other tasks such as information retrieval, question-answering (e.g., to answer questions like `Where is the Taj Mahal?`) and so forth.

Information that users want to extract from documents is often of the following 3 kinds: (1) named entities, (2) relations and (3) events. In this paper, the focus is on Relation Extraction (RE). A *named entity (NE)* is often a word or phrase that represents a specific real-world object. As an example, `Barack Obama` is a NE, and it has one specific *mention* in the following sentence: `Barack Obama is visiting India in January, 2015.`. A *NE mention* in a particular sentence can be using the name itself (`Barack Obama`), nominal (`US President`), or pronominal (`he`). NEs are often categorized into various *generic NE types*: PERSON, ORGANIZATION, LOCATION, DATE, TIME, PHONE, ZIPCODE, EMAIL, URL, AMOUNT etc. Other generic NEs include: FILM-TITLE, BOOK-TITLE etc. In *fine-gained NER*, the problem is to identify generic NE which are hierarchically organized; e.g., PERSON may be sub-divided into POLITICIAN, SCIENTIST, SPORTSPERSON, FILMSTAR, MUSICIAN etc. *Domain-specific NE* consist of, for example, names of proteins, enzymes, organisms, genes, cells etc., in the biological domain. NE in the manufacturing domain are: MANUFACTURER, PRODUCT, BRAND-NAME, FEATURE etc. *Named entity recognition (NER)* is the task of identifying all the mentions (occurrences) of a particular NE type in the given documents. For example: `In a strategic reshuffle at [Bank of America - Merrill Lynch]`$_{ORG}$`, [Atul Singh]`$_{PERSON}$` has taken over as managing director of Global Wealth and Investment Management in [India]`$_{LOCATION}$`.` NER is an important sub-problem in IE; see [91, 81] for surveys of techniques for NER.

A *relation* usually denotes a well-defined (having a specific meaning) relationship between two or more NEs. Some examples of relations are the MEMBER-AFFILIATION relation between PERSON and ORG, HAS relation between PRODUCT and FEATURE, AUTHOR-OF relation between PERSON and BOOK-TITLE and so forth. Example:

`[Bill Gates]`$_{PERSON}$` announced that [John Smith]`$_{PERSON}$` will be [the chief scientist]`$_{PERSON}$` of [Microsoft Corporation]`$_{ORG}$`.`

`The [Epson WorkForce 840's]`$_{PRODUCT}$` [500-page paper capacity]`$_{FEATURE}$` is convenient for high-volume office printing, and you can stock two different types of paper in [a pair of size-adjustable trays]`$_{FEATURE}$`.`

We focus on binary relations and assume that both the argument NE mentions that participate in a relation mention occur in the same sentence. Note also that a relation need not exist between every pair of NE mentions in the

given sentence. Example: `[John Smith]`$_{PERSON}$ `had visited [Bank of America]`$_{ORG}$ `in August 2003.`
The task of *relation extraction (RE)* consists of identifying mentions of the relations of interest in each sentence of the given documents. Relation extraction is a very useful next step in IE, after NER.

Successful RE requires detecting both the argument mentions, along with their entity types chaining these mentions to their respective entities, determining the type of relation that holds between them. Relation extraction faces many challenges. First, there is a vast variety of possible relations, which vary from domain-to-domain. Non-binary relations present special challenges. Supervised machine learning techniques applied to RE face the usual difficulty of a lack of sufficient training data. The notion of a relation is inherently ambiguous and there is often an inherent ambiguity about what a relation "means", which is often reflected in high inter-annotator disagreements. As the expression of a relation is largely language-dependent, it makes the task of RE also language-dependent. Most of the work that we survey is concerned with English, and extending these techniques to non-English languages is not always easy.

1.1. Datasets for Relation Extraction

Automatic Content Extraction (ACE) [32] [1], [2] is an evaluation conducted by NIST to measure the tasks of Entity Detection and Tracking (EDT) and Relation Detection and Characterization (RDC). ACE defines the following NE types: PERSON, ORG, LOCATION, FACILITY, GEO_POLITICAL_ENTITY (GPE), WEAPON etc. GPE refers to a geographically defined regions with a political boundary, e.g. countries, cities. The EDT task consists of detecting mentions of these NEs, and identifying their co-references. The RDC task consists of detecting relations between entities identified by the EDT task. The Figure 1.1 shows various relation types and subtypes defined in the ACE 2003 [76] and ACE 2004 [77] datasets. ACE 2004 dataset [77] is the most widely used dataset in the literature to report the performance of various relation extraction techniques. Some examples of the ACE 2004 relation types are shown in the Table 1.1. ACE released two more datasets, namely - ACE 2005 [124] and ACE 2007 [114]. Hachey et al. [45] describes the ACE 2004 and ACE 2005 datasets and their standardization.

1.2. Relation Extraction : Global level Vs Mention level

The term "Relation Extraction" is often used in the literature to refer to either global level RE or mention level RE. In this survey, we cover both the types. Global RE system is expected to produce a list of entity pairs for which a certain semantic relation exists. It generally takes a large text corpus as input and produces such a list as output. On the other hand, mention level RE system takes as input an entity pair as well as the sentence which contains it.

[1] http://www.nist.gov/speech/tests/ace/
[2] http://www.ldc.upenn.edu/Projects/ACE

ACE 2003			ACE 2004		
Type	**Subtype**	**Count**	**Type**	**Subtype**	**Count**
AT	based-in	496	PHYS	LOCATED	745
	located	2879		NEAR	87
	residence	395		PART-WHOLE	384
NEAR	relative-location	288	PER-SOC	BUSINESS	179
PART	other	6		FAMILY	130
	part-of	1178		OTHER	56
	subsidiary	366	EMP-ORG	EMPLOY-EXEC	503
ROLE	affiliate-partner	219		EMPLOY-STAFF	554
	citizen-of	450		EMPLOY-undetermined	79
	client	159		MEMBER-OF-GROUP	192
	founder	37		SUBSIDIARY	209
	general-staff	1507		PARTNER	12
	management	1559		OTHER	82
	member	1404	ART	USER/OWNER	200
	other	174		INVENTOR/MANUFACTURER	9
	owner	274		OTHER	3
SOCIAL	associate	119	OTHER-AFF	ETHNIC	39
	grandparent	10		IDEOLOGY	49
	other-personal	108		OTHER	54
	other-professional	415	GPE-AFF	CITIZEN/RESIDENT	273
	other-relative	86		BASED-IN	216
	parent	149		OTHER	40
	sibling	23	DISC	DISC	279
	spouse	89			

Figure 1: Relation Types and Sub-Types in ACE 2003 and ACE 2004 datasets, along with their occurrence counts

.

Relation Type	Examples
EMP_ORG	Indian minister, employee of Microsoft
PHYS	people from the valley,
PER_SOC	his brother, wife of the President
GPE_AFF	Indian singer, citizens of Japan
OTHER_AFF	Christian people
ART	my house, British helicopters

Table 1: Examples of ACE 2004 Relation types. The entity mentions involved in the relations, are underlined

It then identifies whether a certain relation exists for that entity pair. Consider the entity mentions `Obama` and `India` in the sentence : `Obama is visiting India today.` Here, the mention level RE system would identify the PHYS relation between `Obama` and `India`. Consider another sentence : `Obama likes India's culture.` Here, mention level RE system should identify that no relation exists between `Obama` and `India` in this particular sentence. Automatic context Extraction (ACE) program [77] called mention level RE with a more appropriate name : Relation Detection and Characterization (RDC).

1.3. Previous Surveys

A comprehensive survey of Information Extraction was presented by Sarawagi [109] which covered some RE techniques. The first dedicated survey of RE techniques was carried out by Bach and Badaskar [4] but it does not cover many of the

recent advancements in the field. Another recent survey was presented by de Abreu et al. [30] which covers various RE techniques used specifically for Portuguese language. Cohen and Hersh [23] presented a comprehensive survey of early text mining work in biomedical domain, including extraction of biological entities and relations. Zhou et al. [152] surveyed most of the recent biomedical RE approaches.

In this paper, we survey various RE techniques which are classified into several logical categories : (i) supervised techniques including features-based and kernel based methods (Section 2), (ii) a special class of techniques which jointly extract entities and relations (Section 3) (ii) semi-supervised (Section 4), (iii) unsupervised (Section 5), (iv) Open Information Extraction (Section 6) and (v) distant supervision based techniques (Section 7). Some recent advanced RE techniques are discussed in the section 8. Finally, we conclude in the section 9 by discussing some of the potential future research directions for RE.

2. Supervised Approaches

Supervised approaches focus on RE at the mention level. These approaches require labelled data where each pair of entity mentions is labelled with one of the pre-defined relation types. A special relation type NONE is used to label those pairs where none of the pre-defined relation types hold. In general, RE is formulated as a multi-class classification problem with each class corresponding to a different relation type (including NONE). These approaches are broadly classified into two types: Feature-based and Kernel-based methods.

2.1. Feature-based Methods

In feature-based methods, for each relation instance (i.e. pair of entity mentions) in the labelled data, a set of features is generated and a classifier (or an ensemble of classifiers) is then trained to classify any new relation instance. Kambhatla [57] described various lexical, syntactic and semantic features for extracting features. Consider the entity pair `<leaders, Venice>` in the sentence : `Top leaders of Italy's left-wing government were in Venice.` Table 2 lists various features derived for this entity pair.

Kambhatla [57] trained a maximum entropy classifier with 49 classes : two for each relation subtype (ACE 2003 has 24 relation subtypes and each subtype gives rise to 2 classes considering order of relation arguments) and a NONE class for the case where the two mentions are not related. Building upon Kambhatla's work, Zhou et al. [44] explored some more features to improve the RE performance further. Some of the important additional features are as follows:
Word based features: words between the two mentions (classified into 3 bins: the first word, last word and other words); first and second words before and after the mentions; headwords of the mentions; flag indicating whether any one of the mention is contained within another
Base phrase chunking based features: phrase heads between the two mentions (classified into 3 bins: the first, the last and other phrase heads); the first

Feature Types	Example
Words: Words of both the mentions and all the words in between	`M11_leaders`, `M21_Venice`; `B1_of`, `B2_Italy`, `B3_'s`, `B4_left-wing`, `B5_government`, `B6_were`, `B7_in`
Entity Types: Entity types of both the mentions	`E1_PERSON`, `E2_GPE`
Mention Level: Mention types (NAME, NOMINAL or PRONOUN) of both the mentions	`M1_NOMINAL`, `M2_NAME`
Overlap: #words separating the two mentions, #other mentions in between, flags indicating whether the two mentions are in the same NP, VP or PP	`7_Words_Apart`, `2_Mentions_In_Between` (`Italy & government`), `Not_Same_NP`, `Not_Same_VP`, `Not_Same_PP`
Dependency: Words, POS and chunk labels of words on which the mentions are dependent in the dependency tree, #links traversed in dependency tree to go from one mentions to another	`M1W_were`, `M1P_VBD`, `M1C_VP`, `M2W_in`, `M2P_IN`, `M2C_PP`, `DepLinks_3`
Parse Tree: Path of non-terminals connecting the two mentions in the parse tree, and the path annotated with head words	`PERSON-NP-S-VP-PP-GPE`, `PERSON-NP:leaders-S -VP:were-PP:in-GPE`

Table 2: Various feature types with examples described by Kambhatla [57]

and second phrase heads before and after the mentions; path of phrase labels connecting the two mentions with and without augmentation with head words **Features based on semantic resources:** country name list is used to distinguish between *citizen_of* and *residence* relation types - when the first (second) mention is a country name, a feature `CountryET2` (`ET1Country`) is generated; personal relative trigger word list is used to differentiate 6 personal social relation subtypes. It is gathered from WordNet by collecting all the words having the semantic class "relative". This list is then classified into different categories representing each of the social relation subtype. The feature `SC1ET2` (`ET1SC2`) is generated when the first (second) mention is found in the trigger list where ET2 (ET1) is type of second (first) mention's entity type and SC1 (SC2) is semantic class of the first (second) mention.

Zhou et al. [44] employed a SVM classifier using these features and achieved better performance than Kambhatla's system. As SVM is a binary classifier, to achieve multi-class classification *one vs. others* strategy is used. They also analysed the results to find contributions by various types of features. The phrase based chunking features were observed to be contributing the most to the increased accuracy. The syntactic features based on dependency tree and parse tree contributed only slightly. A major reason for this is that most of the relations in the ACE data are short-distance relations and simple features like word and chunking features are enough to identify such relations.

A systematic study of the feature space for RE is conducted by Jiang and Zhai [56] and they also evaluated the effectiveness of different feature subspaces. They defined a unified graphic representation of the feature space, and experimented with 3 feature subspaces, corresponding to sequences, syntactic parse trees and dependency parse trees. Experimental results showed that each subspace is effective by itself, with the syntactic parse tree subspace being the most effective. Also, combining the three subspaces did not generate much improve-

ment. They observed that within each feature subspace, using only the basic unit features can already give reasonably good performance and adding more complex features may not improve the performance much.

Some more interesting features are described by Nguyen et al. [84] who used SVM for identifying relations between Wikipedia entities. They semi-automatically created list of keywords providing cues for each relation type. E.g. for PHYS relation, words like `located`, `headquartered`, `based` act as keywords. They came up with a novel concept of *core tree* to represent any relation instance. This *core tree* not only consists of the shortest path connecting two entity mentions in the dependency tree but also additional paths connecting nodes on the shortest path to keywords in the sentence. Then subtrees of this *core tree* are mined to act as features.

Chan and Roth [17] described an interesting approach for supervised RE based on the observation that all ACE 2004 relation types are expressed in one of several constrained syntactico-semantic structures.

1. *Premodifier*: An adjective or a proper noun modifies another noun (`Indian minister`)
2. *Possessive*: First mention is in possessive case (`Italy's government`)
3. *Preposition*: Two mentions are related via a preposition (`governor of RBI`)
4. *Formulaic*: Two mentions are written is some specific form (`Mumbai, India`)

These structures can be identified by using some simple rules and patterns. Authors observed that identifying a appropriate syntactico-semantic structure first and then using a specialized RE model which leverages these structures, results in better performance.

One of the major problem that arises in supervised RE methods is that of *Class Imbalance*. This happens because number of negative instances (i.e. entity pairs having no meaningful relation) greatly outnumber number of positive instances (i.e. entity pairs having any one of the pre-defined relation type). This *Class Imbalance* results in a higher precision and a lower recall as classifiers tend to overproduce the NONE class. Kambhatla [58] presented a novel solution for this problem based on voting among a committee of classifiers that significantly boosts the recall in such situations.

Once the features are designed, feature-based methods can simply use any classifier from the Machine Learning literature. Most of the efforts in these methods are spent in designing the "right" set of features. Arriving at such a features set requires careful analysis of contribution of each feature and knowledge of underlying linguistic phenomena.

2.2. Kernel Methods

The overall performance of feature-based methods largely depends on effectiveness of the features designed. The main advantage of kernel based methods is that such explicit feature engineering is avoided. In kernel based methods, kernel functions are designed to compute similarities between representations of two relation instances and SVM (Support Vector Machines) is employed for classification. Various kernel based RE systems propose different representations for relation instances like sequences, syntactic parse trees etc. Most of the

techniques measure the similarity between any two representations (say trees) in terms of number of shared sub-representations (subtrees) between them.

2.2.1. Sequence Kernel

Relation instances are represented as sequences and the kernel computes number of shared subsequences between any two sequences. Motivated by the string subsequence kernel (Lodhi et al. [70]), Bunescu and Mooney [80] proposed a sequence kernel for RE. The simplest way to construct a sequence to represent a relation instance is to simply consider the sequence of words from the first mention to the second one in the sentence. Rather than having each sequence element as a singleton word, the authors proposed to generalize each word to a feature vector. Each relation instance is then represented as a sequence of feature vectors, one feature vector for each word. The features come from the following domains:

- Σ_1: Set of all words
- Σ_2: Set of all POS tags = {NNP, NN, VBD, VBZ, IN, $\cdots$}
- Σ_3: Set of all generalized POS tags = {NOUN, VERB, ADJ, ADV, $\cdots$}
- Σ_4: Set of entity types = {PER, ORG, LOC, GPE, $\cdots$}

Consider the relation instance formed by the entity pair `Italy-government` from our example sentence. Table 3 shows the sequence of feature vectors for this instance, where each row is a feature vector. It is clear that the domain

Word	POS tag	Generalized POS tag	Entity Type
`Italy`	NNP	NOUN	GPE
`'s`	POS	POS	O
`left-wing`	JJ	ADJ	O
`government`	NN	NOUN	ORG

Table 3: Example of sequence of feature vectors (Sequence s)

Word	POS tag	Generalized POS tag	Entity Type
`India`	NNP	NOUN	GPE
`'s`	POS	POS	O
`summer`	NN	NOUN	O
`capital`	NN	NOUN	GPE

Table 4: Example of sequence of feature vectors (Sequence t)

for sequences of feature vectors is $\Sigma_X = \Sigma_1 \times \Sigma_2 \times \Sigma_3 . \times \Sigma_4$. The aim is to design a kernel function which finds shared subsequences u belonging to the domain $\Sigma_U^* = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \cup \Sigma_4$. Given two sequences s, t of feature vectors, Bunescu and Mooney [80] defined Generalized Subsequence kernel, $K_n(s, t, \lambda)$ which computes number of weighted feature sparse subsequences u of length n such that,

- $u \prec s[ii]$ and $u \prec t[jj]$, for some index sequences ii, jj of length n
- the weight of u is $\lambda^{l(ii)+l(jj)}$, where $0 < \lambda < 1$ and $l(ii)$ is the length of the subsequence which is the difference between largest and smallest index in ii. Sparser the subsequence, lower is its weight.

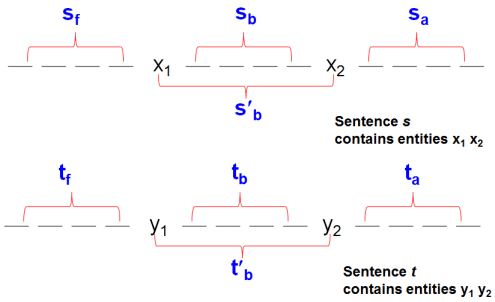

Figure 2: Illustration of sequence formation for various subkernels of overall relation kernel

Here, $\prec$ indicates *component-wise belongs to* relation, i.e. if $ii = (i_1, i_2, \cdots, i_{|u|})$ and $u \prec s[ii]$ then $u[1] \in s[i_1], u[2] \in s[i_2], \cdots u[|u|] \in s[i_{|u|}]$. Suppose, $ii = (1, 2, 4)$ then (NNP, 's, NN)$\prec s[ii]$ (Sequence s as shown in Table 3). Considering the sequences s (Table 3) and t (Table 4), some of the shared sparse subsequences of length $n = 3$ are:

(NNP, 's, NN); (NOUN, 's, NN); (NNP, POS, NN); (NOUN, 's, NOUN)

The generalized subsequence kernel for sequences s and t is computed efficiently using following recursive formulation:

- $K_0^{'}(s,t) = 1$, for all s, t
- $K_i^{"}(sx, ty) = \lambda K_i^{"}(sx, t) + \lambda^2 K_{i-1}^{'}(s,t) \cdot c(x,y)$
- $K_i^{'}(sx, t) = \lambda K_i^{'}(s,t) + K_i^{"}(sx,t)$
- $K_n(sx,t) = K_n(s,t) + \sum_j \lambda^2 K_{n-1}^{'}(s, t[1:j-1) \cdot c(x, t[j])$

Here, x and y are feature vectors and $c(x,y)$ is number of common features between x and y. sx is a sequence of feature vectors constructed by appending feature vector x to the sequence s.

Relation Kernel (rK) is defined as a sum of 4 subkernels, each of which captures a specific type of pattern and is based on the generalized subsequence kernel. The detailed description of these subkernels along with types of patterns they capture are explained with the help of the Figure 2.2.1 below:

Fore-Between subkernel (fbK): Counts number of common fore-between patterns, i.e. number of shared subsequences between $s_f s'_b$ and $t_f t'_b$. (president PER of ORG)

Between subkernel (bK): Counts number of common between patterns, i.e. number of shared subsequences between s'_b and t'_b. (PER joined ORG)

Between After subkernel (baK) : Counts number of common between-after patterns, i.e. number of shared subsequences between $s'_b s_a$ and $t'_b t_a$. (PER chairman of ORG announced)

Modifier subkernel (mK): When there are no other words in between two entity mentions and the first mention acts as a modifier for the other, modifier patterns are useful. This subkernel counts number of common modifier patterns, i.e. number of shared subsequences between $x_1 x_2$ and $y_1 y_2$. Example of

such modifier pattern is : `Serbian general`, where the first mention `Serbian` modifies the second mention `general`. The overall relation kernel is defined as:

$$rK(s,t) = fbK(s,t) + bK(s,t) + baK(s,t) + mK(s,t)$$

Bunescu and Mooney [80] then used SVM classifier based on this relation kernel (rK). They also experimented with two different scenarios as follows:
Scenario 1: Only one multi-class SVM is trained where each relation type is corresponds to one class and an extra class NONE to represent no-relation cases.
Scenario 2: One binary SVM is trained only to decide whether any relation holds or not where all relation types are combined as a single class. Another multi-class SVM is then trained which decides appropriate relation type for positive instances identified by the binary SVM classifier. This scenario was found to be yielding better performance than the other one.

2.2.2. Syntactic Tree Kernel

Structural properties of a sentence are encoded by its constituent parse tree. The tree defines the syntax of the sentence in terms of constituents such as noun phrases (NP), verb phrases (VP), prepositional phrases (PP), POS tags (NN, VB, IN, etc.) as non-terminals and actual words as leaves. Constituent parse tree for our example sentence is shown in the figure 3. The syntax is usually governed by Context Free Grammar (CFG). The task of constructing a constituent parse tree for a given sentence, is called as *parsing*. Collins et al. [25] and Miller et al. [73] proposed statistical parsing models to extract relations from text where they considered the parse trees augmented with information about entities and relations. We focus on the approaches which make use of parse trees already produced by some parsers.

Collins and Duffy [24] proposed Convolution Parse Tree Kernel (K_T) to compute similarity between any two syntactic trees. It computes number of common subtrees shared by two syntactic parse trees. Here, subtree is defined as any subgraph of a tree which satisfies two conditions - i) it should include more than one node, and ii) entire productions must be included at every node. The kernel is designed in such a way that each possible subtree becomes a dimension in the projected space. The image of a syntactic tree T in transformed space is $h(T) = [h_1(T), h_2(T), \cdots h_n(T)]$, where $h_i(T)$ denotes number of occurrences of i^{th} subtree in the tree T and n denotes number of all possible subtrees (subtree vocabulary size). For any two trees T_1, T_2, the value of kernel is simply the inner product of their images in the transformed space, i.e. $K_T(T_1, T_2) = h(T_1) \cdot h(T_2)$.

Efficient Computation: It is not feasible to explicitly construct the image vector, as number of all possible subtrees is huge. Hence, the kernel has to be computed efficiently without actually iterating through all possible subtrees. Let $I_i(n) = 1$ if i^{th} subtree is seen rooted at node n and 0 otherwise. Let N_1 and N_2 be sets of nodes in trees T_1 and T_2 respectively.

$$h_i(T_1) = \sum_{n_1 \in N_1} I_i(n_1), h_i(T_2) = \sum_{n_2 \in N_2} I_i(n_2)$$

$$h(T_1) \cdot h(T_2) = \sum_i h_i(T_1) h_i(T_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) h_i(n_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1, n_2)$$

Here, $C(n_1, n_2)$ counts number of common subtrees rooted at n_1 and n_2, which can be computed in polynomial time using following recursive definition.
1. If the productions at n_1 & n_2 are different then, $C(n_1, n_2) = 0$
2. If the productions at n_1 & n_2 are same and n_1, n_2 are pre-terminals then, $C(n_1, n_2) = 1$
3. If the productions at n_1 & n_2 are same and n_1, n_2 are not pre-terminals then,

$$C(n_1, n_2) = \prod_{j=1}^{nc(n_1)} (1 + C(ch(n_1, j), ch(n_2, j)))$$

$nc(n)$ denotes number of children of n and $ch(n, j)$ denotes j^{th} child-node of n.
Relation Instance Representation: A sentence containing N_e entity mentions gives rise to $\binom{N_e}{2}$ relation instances. Hence, it is also important to decide which part of the complete syntactic tree characterizes a particular relation instance. Zhang et al. [146] described five cases to construct a tree representation for a given relation instance which are shown in the figure 3. These representations are for the relation instance constituting entity mentions `leaders` (E1) and `government` (E2).
1. Minimum Complete Tree (MCT): It is the complete subtree formed by the lowest common ancestor of the two entities.
2. Path-enclosed Tree (PT): It is the smallest subtree including both the entities. It can also be described as the subtree enclosed by the shortest path connecting two entities in the parse tree of the sentence.
3. Context-sensitive Path Tree (CPT): It is the extended version of PT where one word left of first entity and one word right of second entity are included.
4. Flattened Path-enclosed Tree (FPT): It is the modified version of PT where the non-POS non-terminal nodes which are having a single in and out arcs, are bypassed.
5. Flattened Context-sensitive Path Tree (FCPT): It is the modified version of CPT where the non-POS non-terminal nodes which are having a single in and out arcs, are bypassed.

In their experimental analysis, Zhang et al. [146] found the Path-enclosed Tree (PT) performs the best when used for computing K_T. Zhou et al. [154] extended this work further by automatically determining a dynamic context-sensitive tree span for RE by extending the Path-enclosed Tree (PT) to include necessary context information. It also proposed a context-sensitive convolution tree kernel, which in addition to context-free subtrees, considers context-sensitive subtrees also by considering their ancestor node paths as their contexts. Another approach to dynamically determine the tree span, was proposed by Qian et al. [97]. They used the information about *constituent dependencies* to keep the nodes and their head children along the path connecting the two mentions and removed the other noisy information from the syntactic parse tree. In any Context Free Grammar (CFG) rule, the parent node depends on the head child and this is what the authors called as *constituent dependencies*. Another extension to the syntactic tree kernel was proposed by Qian et al. [98] where the parse tree is augmented with entity features such as entity type, subtype, and mention level. Khayyamian et al. [60] proposed a generalized version of

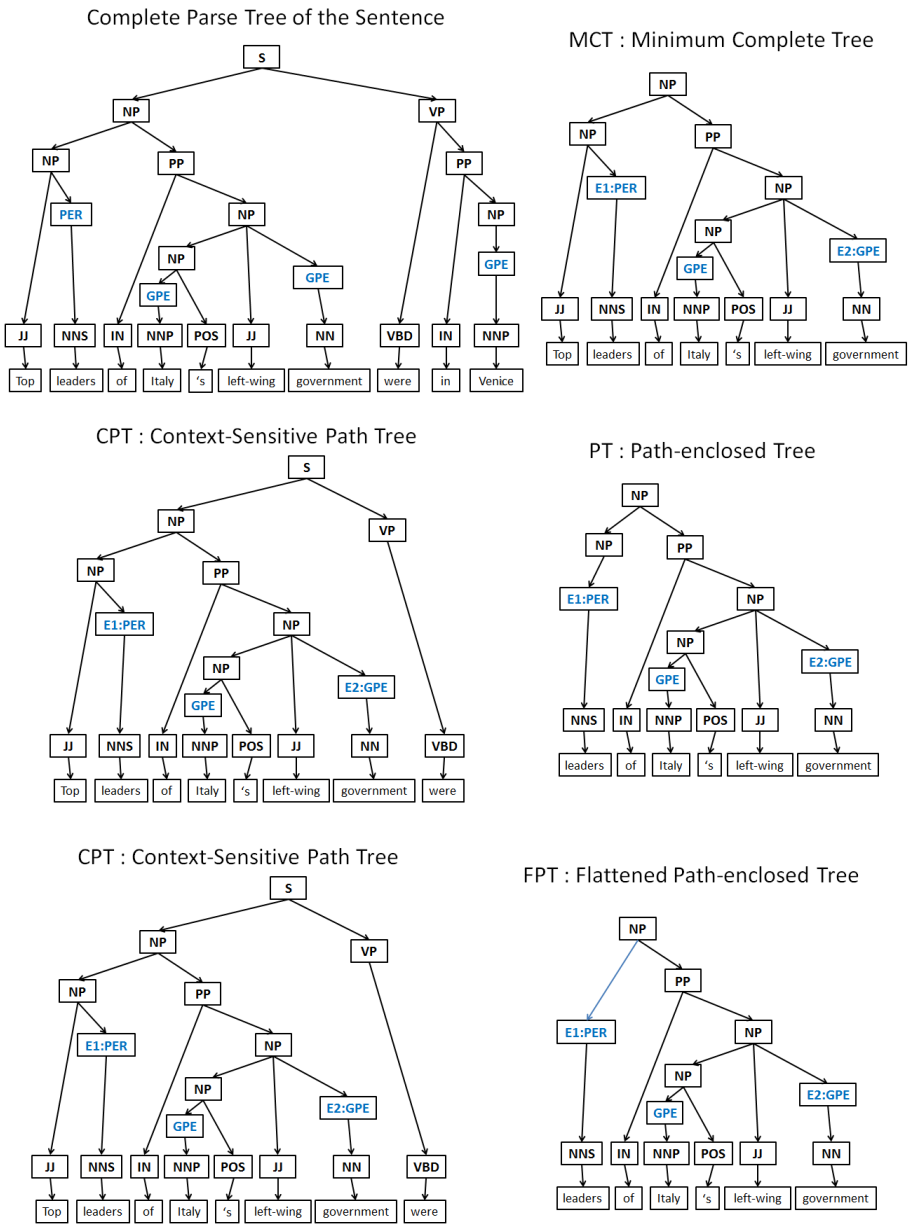

Figure 3: Various tree representations described in Zhang et al. [147]

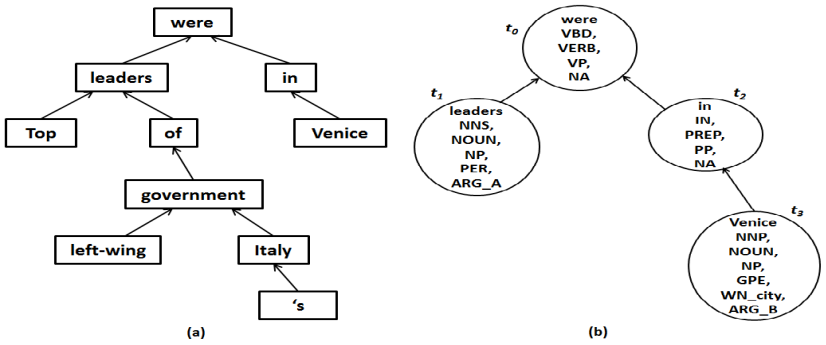

Figure 4: Dependency Tree for the example sentence and augmented dependency tree representation for the relation instance `leaders-Venice`

syntactic tree kernel by Collins and Duffy [24] for better RE performance. Most of the work in RE using syntactic parse tree kernels is discussed in detail by Zhang et al. [148] and Zhou et al. [153]. Recently, Sun and Han [119] proposed an extension to the basic syntactic tree kernel, named Feature-enriched Tree Kernel (FTK). Here, the authors proposed to annotate the nodes in a syntactic tree with a set of discriminant features (like WordNet senses, context information, properties of entity mentions, etc.) and FTK is designed to compute the similarity between such enriched trees.

2.2.3. Dependency Tree Kernel

Grammatical relations between words in a sentence are encoded by its dependency tree. Words in the sentence become the nodes in the tree and dependency relations among them become the edges. Each word except the root has exactly one parent in the tree. The directions of the edges are generally shown as pointing from dependent word to its parent. Dependency tree for our example sentence is shown in the figure 4(a). Culotta and Sorensen [27] proposed a kernel to compute similarity between two dependency trees. Their work was an extension of the tree kernel proposed by Zelenko et al. [141] which was defined for shallow parse tree representations.

Relation Instance Representation: For each entity mention pair in a sentence, smallest subtree of the sentence's dependency tree which contains both the mentions, is considered. Each node in dependency tree is also augmented with additional features such as POS tag, generalized POS tag, chunk tag, entity type, entity level (name, nominal, pronoun), WordNet hypernyms and relation argument (ARG_A indicating first mention and ARG_B indicating second mention). In case of our example sentence, the relation instance formed by mention pair `leaders-Venice` is represented by augmented dependency tree as shown in the figure 4(b).

Formally, a relation instance is represented by a augmented dependency tree T having nodes $\{t_0 \cdots t_n\}$ where each node t_i has the features $\phi(t_i) = \{v_1 \cdots v_d\}$. Let, $t_i[c]$ denote all children of t_i and $t_i.p$ denote parent of t_i. Also, $t_i[jj]$ de-

notes a particular subset of children of t_i where $jj = j_1, j_2, \cdots j_{l(jj)}$ (such that $j_1 < j_2 < \cdots < j_{l(jj)}$, $l(jj)$:Length of sequence, Sparseness $d(jj) = j_{l(jj)} - j_1 + 1$) is an ascending sequence of indices. In case of the tree in the figure 4(b), $t_0[c] = t_0[\{0,1\}] = \{t_1, t_2\}$ and $t_1.p = t_0$. For comparison on any two nodes t_i, t_j two functions are defined,

1. **Matching function ($m(t_i, t_j)$):** It returns 1 if some important features are shared between t_i and t_j, otherwise it returns 0.
2. **Similarity function ($s(t_i, t_j)$):** Unlike the binary matching function, similarity function returns a positive real value as a similarity score between t_i and t_j. It is defined as,

$$s(t_i, t_j) = \sum_{v_q \in \phi(t_i)} \sum_{v_r \in \phi(t_j)} C(v_q, v_r)$$

where $C(v_q, v_r)$ is a compatibility function between two feature values v_q and v_r. In the simplest form, $C(v_q, v_r) = 1$ if $v_q = v_r$ and 0 otherwise.

The overall dependency tree kernel $K(T_1, T_2)$ which measures similarity between two dependency trees T_1 and T_2 rooted at t_{10} and t_{20} respectively, is defined as follows:
$K(T_1, T_2) = 0$, if $m(t_{10}, t_{20}) = 0$
$K(T_1, T_2) = s(t_{10}, t_{20}) + K_c(t_{10}[c], t_{20}[c])$, otherwise
where $K_c(t_i[c], t_j[c])$ is a kernel function over children of t_i and t_j which is defined as:

$$\sum_{\substack{ii,jj \\ l(ii)=l(jj)}} \lambda^{d(ii)+d(jj)} \left(\sum_{s=1}^{l(ii)} K(t[i_s], t[j_s]) \right) \prod_{s=1}^{l(ii)} m(t_i[i_s], t_j[j_s])$$

Intuitively, whenever a pair of *matching* nodes is found, all possible *matching* subsequences of their children found. Two subsequences are said to be *matching* subsequences when all nodes within them are *matching* pairwise. Similarity scores of all nodes within such *matching* subsequences are then summed up to get overall similarity of children nodes. The constant $0 < \lambda < 1$ acts as a decay factor that penalizes sparser subsequences.

A special *contiguous* kernel is also defined by the authors which constrains the children subsequences ii such that $d(ii) = l(ii)$. In addition to these sparse (K_0) and contiguous (K_1) tree kernels, the authors also experimented with a bag-of-words kernel (K_2) which treats the tree as a vector of features over nodes, disregarding the tree structure. They also experimented with two composite kernels : $K_3 = K_0 + K_1$ and $K_4 = K_1 + K_2$, and found that K_4 achieves the best performance on the ACE dataset.

Harabagiu et al. [47] proposed to enhance this dependency tree kernel with the semantic information obtained from shallow semantic parsers using PropBank [63] and FrameNet [5]. There are other approaches which also used kernels defined over dependency trees for RE like the one by Reichartz et al. [100].

2.2.4. Dependency Graph Path Kernel

Bunescu and Mooney [14] proposed a novel dependency path based kernel for RE. The main intuition was that the information required to assert a relationship between two entities in a sentence is typically captured by the shortest path between the two entities in the dependency graph. Kernel is then designed

to capture similarity between shortest dependency paths representing two relation instances. Consider the dependency graph (figure 4(a)) of our example sentence. For the relation instance `<leaders, Venice>`, the shortest path is : `leaders`→`were`←`in`←`Venice`.

Completely lexicalized paths would lead to data sparsity. Hence, words are categorized into word classes with varying degrees of generality. These word classes are POS tags, generalized POS tags, Named Entity types. Following is an example of a generalized path where each word in the sequence has been generalized.

$$\begin{bmatrix} \texttt{leaders} \\ NNS \\ Noun \\ PER \end{bmatrix} \times [\rightarrow] \times \begin{bmatrix} \texttt{were} \\ VBD \\ Verb \end{bmatrix} \times [\leftarrow] \times \begin{bmatrix} \texttt{in} \\ IN \end{bmatrix} \times [\leftarrow] \times \begin{bmatrix} \texttt{Venice} \\ NNP \\ Noun \\ GPE \end{bmatrix} \tag{1}$$

Each possible path is considered a feature. The above generalized path gives rise to features such as `leaders`→`were`←`in`←`Venice`, `NNS`→`were`←`in`←`Venice`, `NNS`→`VBD`←`in`←`GPE` and so on. There are $4 \times 1 \times 3 \times 1 \times 2 \times 1 \times 4 = 96$ such features. Shorted Dependency Path Kernel computes number of common path features shared by two relation instances. Let two relation instances R_1 and R_2 be represented by their respective shortest paths: $R_1 = x_{11}x_{12}\cdots x_{1m}$, $R_2 = x_{21}x_{22}\cdots x_{2n}$, then the kernel is computed as,
$K(R_1, R_2) = \prod_{i=1}^{n} c(x_{1i}, x_{2i})$ when $m = n$
$K(R_1, R_2) = 0$ when $m \neq n$

Consider another generalized path as follows:

$$\begin{bmatrix} \texttt{John} \\ NNP \\ Noun \\ PER \end{bmatrix} \times [\rightarrow] \times \begin{bmatrix} \texttt{went} \\ VBD \\ Verb \end{bmatrix} \times [\leftarrow] \times \begin{bmatrix} \texttt{to} \\ IN \end{bmatrix} \times [\leftarrow] \times \begin{bmatrix} \texttt{London} \\ NNP \\ Noun \\ GPE \end{bmatrix} \tag{2}$$

The value of Dependency Path Kernel for the above two relation instances (Eq. 1, Eq. 2) is computed to be $2 \times 1 \times 2 \times 1 \times 1 \times 1 \times 2 = 8$, i.e. number of common featured shared by both the relation instances is 8. The dependency path kernel explained above imposes a hard constraint that the two paths should have exactly the same number of nodes. In order to make it more flexible, Wang [127] proposed a *convolution dependency path kernel* which finds number of common subsequences shared by two dependency path sequences.

2.2.5. Composite Kernels

A composite kernel combines the information captured by the individual kernels. For example, a composite kernel which combines syntactic tree kernel and sequence kernel, uses syntactic information captured by the tree kernel as well as lexical information captured by the sequence kernel. It is important to ensure that the combination of two individual kernels is also a valid kernel function. Some of the valid ways of combining two individual kernels are : sum, product, linear combination.

Zhang et al. [146] used syntactic tree kernel for RE. This work was extended by Zhang et al. [147] by experimenting with another kernel function, the Entity Kernel (K_E) which captures similarity of pairs of entity mentions. Each entity

mention is characterized by various features such as headword, entity type, entity subtype and mention type (name, nominal and pronoun). Entity Kernel computes number of shared features between two pairs of entity mentions. Using Syntactic Tree Kernel (K_T) and Entity Kernel (K_E), two composite kernels are constructed to compute similarity between two relation instances R_1, R_2:

- Linear Combination, $K_{LC} = \alpha \cdot NK_E + (1 - \alpha) \cdot NK_T$
- Polynomial Expansion, $K_{PE} = \alpha \cdot (1 + NK_E)^2 + (1 - \alpha) \cdot NK_T$

where NK_E (NK_T) is the normalized version of Entity (Syntactic Tree) Kernel.

$$NK_E(R_1, R_2) = \frac{K_E(R_1, R_2)}{\sqrt{K_E(R_1, R_1)K_E(R_2, R_2)}}$$

Composite Kernels were found to be performing better than the individual kernels and K_{PE} displays the best performance. Normalization of individual kernels before combining is necessary to ensure that value of one kernel does not overwhelm the value of another.

Zhao and Grishman [151] presented a RE approach which combines information from three different levels of NLP processing: tokenization, sentence parsing and deep dependency analysis. Individual kernel functions are designed to capture each source of information. Then composite kernels are developed to combine these individual kernels so that processing errors occurring at one level can be overcome by information from other levels. Nguyen et al. [89] investigated effectiveness of combining various kernels capturing syntactic and semantic information. Syntactic information was captured by individual kernels based on constituent and dependency trees, whereas semantic information is captured by entity types and lexical sequences. Wang et al. [125] proposed a new composite kernel for RE which uses a sub-kernel defined using *relation topics*. A training set of around 7000 existing Wikipedia relations is automatically created (using a technique similar distant supervision explained later) by making use of Wikipedia infoboxes. Then, the *relation topics* are defined over these existing relations. By leveraging this knowledge extracted from the Wikipedia relation repository, the authors reported a significant improvement in RE performance. Wang et al. [126] further explained the application of this kernel based on *relation topics* in the Question Answering framework *DeepQA*.

2.3. Evaluation

Some of the papers in supervised RE report their performance on non-standard datasets, but some others report their performance on standard datasets such as ACE 2003 and ACE 2004. Similar to multi-class classification, the performance of supervised RE systems is evaluated in terms of *precision*, *recall* and *F-measure* of non-NONE classes. The table 5 shows performance of various approaches on the ACE 2004 dataset with 5-fold cross-validation. Even though the same dataset is used by various approaches, the actual splits/folds used in the 5-fold cross-validation might be different. Nevertheless, these figures provide a rough idea of comparative performances of these approaches. It can be observed that the kernel-based methods generally outperform the feature-based

Type	Approach	P	R	F
Features based	LX, ST and DT based features [56]	0.737	0.694	0.715
	Additional features based on Syntactico-Semantic structures [17]	0.754	0.68	0.715
Kernel based	Composite kernel combining individual LX, ST and DT kernel [151]	0.692	0.705	0.7035
	Composite kernel combining ST and EN kernels [147]	0.761	0.684	0.721
	ST kernel with dynamically determined tree span [154]	0.822	0.702	0.758
	Composite kernel combining ST and DT kernels along with semantic information [89]	0.766	0.67	0.715
	ST kernel where parse tree is augmented with entity features [98]	0.792	0.674	0.728
	ST kernel with dynamically determined tree span [97]	0.83	0.72	**0.771**

Table 5: Comparative RE performance of various supervised approaches on 7 major relation types in ACE 2004 dataset (5-fold cross validation, LX: lexical, ST: syntactic tree, DT: dependency tree)

methods and among the various kernel-based methods, the methods based on syntactic tree kernel perform the best.

3. Joint Extraction of Entities and Relations

All of the RE techniques explained in the previous section assume that the knowledge about boundaries and types of entity mentions are known before hand. If such knowledge is not available, in order to use these techniques for extracting relations, one needs to first apply some entity mentions extraction technique. Once entity mentions and their entity types are identified, then RE techniques can be applied. Such a "pipeline" method is prone to propagation of errors from the first phase (extracting entity mentions) to the second phase (extracting relations). To avoid this propagation of errors, there is a line of research which models or extracts entities and relations jointly.

3.1. Integer Linear Programming based Approach

Roth and Yih [28] proposed a model, which first learns independent local classifiers for entity extraction and RE. During inference, given a sentence, a global decision is produced such that the domain-specific or task-specific constraints are satisfied. A simple example of such constraints is : both the arguments of the `PER-SOC` relation should be `PER`. Consider the sentence - `John married Paris.` Here, the entity extractor identifies two mentions `John` and `Paris` and also predicts entity types for these mentions. For the first entity, let the predicted probabilities be : Pr(`PER`) = 0.99 and Pr(`ORG`) = 0.01. For the second entity, 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. This problem of taking a global decision consistent with the constraints, is solved by using a Integer Linear Programming approach by Roth and Yih [28]. This Integer Linear Program minimizes the sum of assignment cost function and constraint cost function. The assignment cost function is designed in such a way that if the most probable prediction of a local classifier is chosen then the least cost is incurred. The constraint cost function is designed to impose cost for breaking constraints between connected entities and relations.

The experiments reported significant improvement in the RE performance using ILP for global inference, as compared to the *pipeline* method which first identifies entity types and using these predicted types relation classifier is run. This global inference approach even improves quality of entity classification which is impossible in the *pipeline* approach. Roth and Yih [108] described the global inferencing for entity and relation extraction in further details where they explore some other global inferencing techniques than ILP like the Viterbi Algorithm. Chan and Roth [16] proposed an extension to the original ILP framework for incorporating background knowledge such as hierarchy of relation types, coreference information, etc. The ILP based approach for global inferencing was also used by Choi et al. [22] for joint extraction of entities and relations in the context of opinion information extraction. Here, entities were of two types - source of the opinion and expression of opinion. The only relation considered was the linking relation between the two entities.

3.2. Graphical Models based Approach

The first attempt of using graphical models approach for jointly identifying entities and relations was by Roth and Yih [107]. They proposed a framework where *local* independent classifiers are learned for entities and relations identification. The dependencies between entities and relations are encoded though a bayesian belief network which is a bipartite, directed acyclic graph. Entities are represented as nodes in one layer in bipartite graph whereas relations are represented as nodes in the other layer. Each relation instance node R_{ij} has two incoming edges from its argument entity instance nodes E_i, E_j. Given the feature vector X which characterizes the sentence, the local entity and relation classifiers are used to compute $Pr(E_i|X)$ and $Pr(R_{ij}|X)$, respectively. The constraints are encoded through the conditional probabilities $Pr(R_{ij}|E_i, E_j)$, which can be set manually or estimated from the entities and relations labelled corpus. The joint probability of the nodes in the bayesian network is maximized to get the most probable label assignments for entity and relation nodes, i.e. $(e_1, e_2, \cdots e_n, r_{12}, r_{21}, \cdots r_{n(n-1)})$ would be,

$$arg \max_{e_i, r_{jk}} Pr(E_1, E_2, \cdots E_n, R_{12}, R_{21}, \cdots R_{n(n-1)}) \tag{3}$$

The joint probability expression involving $Pr(E_i|X)$, $Pr(R_{ij}|X)$ and $Pr(R_{ij}|E_i, E_j)$ is not very clear as it is not explicitly mentioned by the authors. They experimented with two specific relations *Born_in* and *Kill* and found that performance

of relation classification using bayesian network is better than the independent relation classifier. But similar improvement for entity classification using was not observed. Possible reason for this can be the fact that very few entities were involved in some relation in the datasets used by the authors.

Yu and Lam [140] proposed a framework based on undirected, discriminative probabilistic graphical model to jointly perform the tasks of entity identification and relation extraction. Moreover, unlike most of the other approaches, the knowledge about entity mention boundaries is not assumed and is incorporated as a part of the model. Only thing that simplifies the problem a little is that the relations are always assumed to be between a principal entity and other secondary entities in the sentence, as the focus is on the encyclopaedia articles where each article is about a principle entity. i.e. arbitrary relations among the secondary entities are not allowed.

Most of the approaches for joint modelling only focus on two tasks at a time, i.e. entity extraction and relation extraction. Singh et al. [112] is the first approach which even models co-references jointly with entity mentions and relations. The task of co-reference resolution is to link entity mentions within a document which refer to the same real-word entity. They proposed a single, joint undirected graphical model that represents the various dependencies between these three tasks. Unlike most other approaches for RE where the modelling is at a sentence level, in this approach a model captures all the entity mentions within a *document* along with the relations and co-references amongst them. The challenge here is to handle such a large number of variables in a single model, which is addressed by the authors through an extension to belief propagation algorithm that sparsifies the domains of variables during inference.

3.3. Card-Pyramid Parsing

Another interesting approach for joint extraction of entities and relations was proposed by Kate and Mooney [59] and used a graph (not probabilistic graphical model) 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 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 5 shows this *card-pyramid* graph for our example sentence.

The aim is 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:

1. **Entity Productions** of the form $EntityType \rightarrow Entity$, e.g. `PER`→`leaders`.

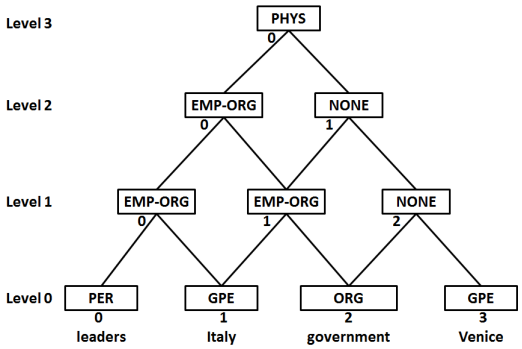

Figure 5: Card-pyramid graph for our example sentence

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.
2. **Relation Productions** of the form $RelationType \rightarrow EntityType1\ EntityType2$, e.g. `PHYS`→`PER GPE`. A local relation classifier is trained to predict the probability that the relation type in the LHS holds between the two entities in the RHS 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.

3.4. Structured Prediction

In most of the approaches for joint extraction of entities and relations, it is assumed that the boundaries of the entity mentions are known. Li and Ji [68] presented an incremental joint framework for simultaneous extraction of entity mentions and relations, which also incorporates the problem of boundary detection for entity mentions. Earlier approaches modelled independent *local* classifiers for identifying entities and relations. Even though optimal global decision taken later, interaction between entity extraction and RE modules is prohibited during training. Hence, the authors proposed to re-formulate this problem as a structured prediction problem. They try 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.

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

The score of each candidate assignment is defined as the inner product of the feature vector $f(x, y)$ and feature weights $\vec{w}$. 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. For decoding, they employed the idea of semi-Markov chain proposed by Sarawagi and Cohen [110], in which each state corresponds to a segment of the input sequence, instead of treating individual tokens/words as states.

Features: Along with various *local* features used for entity and relation extraction, a major advantage of this framework is that arbitrary features for both the tasks can be easily exploited. Some of *global* features used for entity extraction try to capture long distance dependencies among the entity mentions like,

1. Co-reference consistency: Co-reference links between two segments are determined in the same sentence using some simple heuristic rules. A global feature is encoded to check whether two co-referential segments share the same entity type.
2. Neighbour coherence: The entity types of the two neighbouring segments are linked together as a global feature.
3. Part-of-whole consistency: If an entity mention is semantically part of another mention (connected by a *prep_of* dependency link), they should be assigned the same entity type. e.g., in `some of Italy's leaders`, `some` and `leaders` should get same entity type `PER`.

Some of the *global* features designed for RE are:

1. Triangle constraint: Multiple entity mentions are unlikely to be fully connected with the same relation type. A negative feature was used to penalize any structure that contains this type of formation.
2. Inter-dependent compatibility: If two entity mentions are connected by a dependency link, they tend to have compatible relations with other entities. e.g., in the sentence `John and Mary visited Germany`, the *conj_and* dependency link between the mentions `John` and `Mary` indicates that they may share the same relation type with some third entity mention `Germany`.

Another approach for joint extraction of entities and relations was proposed by Miwa and Sasaki [78] which uses a table structure. This table represents entity and relation structures in a sentence. If the number of words in a sentence is n, then the table is $n \times n$ lower triangular matrix where the i^{th} diagonal cell represents the entity type of the i^{th} word. Any (i, j) cell represents the relation type (if any) between the entity mention headed at i^{th} word and another entity mention headed at j^{th} word. With this table representation, the joint extraction problem is mapped to a table-filling problem where labels are assigned to the cells in the table. Similar to the approach of Li and Ji [68], various local and global features are captured to assign a score to any labels assignment in a table.

Majority of the approaches which jointly extract entities and relations, report a significant improvement over the basic pipeline approach. Joint extraction not only improves performance of relation extraction but also proves to be effective for entity extraction. Because, unlike pipeline methods, joint model facilitates the use of relations information for entity extraction. It is difficult to compare various methods for joint modelling because no single, standard dataset is used for reporting results. Some of these approaches (like [107, 28, 59]) achieve

joint modelling only through joint inference as local classifiers for entities and relations are trained independently. Some recent approaches (like [68, 112]) perform actual joint learning where a single model is learned for extracting both entities and relations. There are a few but consistent contributions to this line of research over the time and still there is a scope for more sophisticated joint models in future.

4. Semi-supervised Approaches

It is cost, effort and time intensive task to generate labelled data for RE. Major motivation behind designing semi-supervised techniques is two-fold: i) to reduce the manual efforts required to create labelled data; and ii) exploit the unlabelled data which is generally easily available without investing much efforts. In this section, we describe some major semi-supervised RE approaches.

4.1. Bootstrapping Approaches

Generally, bootstrapping algorithms require a large unlabelled corpus and a few *seed* instances of the relation type of interest. e.g. in order to learn model/patterns for extracting the relation *CaptialOf*, the seed examples can be `<Beijing, China>`, `<New Delhi, India>`, `<London, England>` etc. Given these seed examples, a bootstrapping algorithm is expected to extract similar other entity pairs having the same relation type, e.g. `<Paris, France>`. The first such bootstrapping algorithm named DIPRE (Dual Iterative Pattern Relation Expansion) was proposed by Brin [12]. The intuition behind this algorithm is *Pattern Relation Duality*, which is:

- Given a good set of patterns, a good set of tuples (entity pairs following a certain relation type) can be found.
- Given a good set of tuples, a good set of patterns can be learned

Combination of above two properties provides great power and it is the basis of the DIPRE algorithm. Table 6 shows an overview of DIPRE algorithm.

Input: Seed set S of tuples, i.e. entity pairs known to be related with certain relation type R
Output: Set S grown over multiple iterations
1. Find all *occurrences* of the tuples from the seed set S on the Web
2. Learn *patterns* from these *occurrences*
3. Search the web using these *patterns* and find new tuples and add to the set S
4. Go to step 1 and iterate till there are no new tuples to be added

Table 6: Overview of DIPRE [12] algorithm

Patterns for capturing relation type R between two entities E_1 and E_2 are represented as a 5-tuple: *(order, urlprefix, prefix, middle, suffix)*. Here, *order* is a boolean value and other values are strings. When *order* = true, a pair of entities (E_1, E_2) matches the above pattern if URL of the page matches *urlprefix* and contains the text : <*prefix*> E_1 <*middle*> E_2 <*suffix*>

Example of such a pattern is *(true, "en.wikipedia.org/wiki/", `City of, is capital of, state`)* and it matches a text like `City of Mumbai is capital of Maharashtra state.`

Starting with just 3 seed examples of *author, book* pairs and using the corpus of around 24 million web pages, the DIPRE algorithm was able to generate a list of 15257 unique *author, book* pairs.

Agichtein and Gravano [1] built upon the bootstrapping based idea of DIPRE and they developed a system named *SnowBall*. The two main aspects on which *SnowBall* proposed advancements over DIPRE are : i) Pattern representation and generation; and ii) Evaluation of patterns and tuples.

Pattern Representation and Generation: One of the key advancement that *SnowBall* proposed over the DIPRE, was the inclusion of named entity tags (PER, ORG, LOC, etc.) in the patterns. The presence of named entities in the patterns made them more meaningful and reduced the number of false positives extracted by them.In case of DIPRE patterns, it is expected that *prefix*, *suffix* and *middle* strings of the patterns should match exactly. This condition hampers the coverage of the patterns. *SnowBall* patterns are designed to overcome this problem so that the minor variations in the text (e.g. misspellings, extra articles) do not cause a mismatch. Instead of having string patterns, *SnowBall* represents the context of entities (i.e. *prefix*, *suffix* and *right*) using word vectors in the vector space model. Higher the dot product between two context word vectors, higher is the similarity of the contexts.

Evaluation of Patterns and Tuples: *SnowBall* discards all those patterns which are not precise enough, i.e. the patterns which are more likely to extract false positives. One way to discard such patterns is to filter out all the patterns not supported by some minimum number of seed examples. *SnowBall* also computes *confidence* for each pattern, but this computation assumes that one of the two named entities (say NE_1) is more important than the other. Suppose, the *confidence* of pattern p is to be computed which extracts a candidate tuple $t_{curr} = (e_1, e_2)$ where e_1 is of type NE_1 and e_2 is of type NE_2. If there was a high confidence tuple $t_{prev} = (e_1, e_2')$ generated in the previous iteration with the same entity (e_1) of type NE_1 as in t_{curr}, then this function compares entities e_2 and e_2' which are of type NE_2. If these two are the same, then the tuple t_{curr} is considered a *positive* match for the pattern p. Otherwise, it is considered as a *negative* match. The *confidence* of p is then defined as follows:

$$Conf(p) = \frac{\#positive\_p}{\#positive\_p + \#negative\_p} \tag{4}$$

Here, $\#positive\_p$ and $\#negative\_p$ are the numbers of positive and negative matches for p, respectively. The confidence score of a candidate tuple is then computed using the confidence scores for patterns extracting it. Evaluation of patterns and tuples is the major differentiator between DIPRE and *SnowBall*, as low confidence patterns and tuples are discarded in each iteration avoiding most of the incorrect extractions.

Most of the bootstrapping based techniques, apply relation patterns when both the entities are present as *name* mentions. Gabbard et al. [39] explored the

use of co-reference information for improving the performance (especially the recall) of the bootstrapped RE system. Other major bootstrapping approaches are by Ravichandran and Hovy [99], Pantel and Pennacchiotti [92], Greenwood and Stevenson [43], Rosenfeld and Feldman [105], Blohm and Cimiano [9], Xu et al. [133], Xu [131], Carlson et al. [15] and Xu et al. [132].

For mention-level RE i.e. ACE RDC task, Zhang [150] proposed a bootstrapping based algorithm *BootProject* on top of SVMs. The SVM classifier uses the similar features used by the techniques discussed in the section 2.1. They generalized the Co-training [10] algorithm by relaxing following restrictions on multiple feature "views": mutual exclusivity, conditional independence and sufficiency for classification. Sun [116] observed that bootstrapping approaches like DIPRE and *SnowBall* are not good in extracting general relations like `EMP-ORG` relation in ACE 2004 dataset. They proposed a two-stage bootstrapping approach where the first stage is similar to *SnowBall* whereas the second stage takes the patterns learned by the first stage as inputs and tries to extract more *nominals* (like `manager`, `CEO`, etc.) which indicate the general relation type `EMP-ORG`. Features based on such learned list of nominals are then incorporated into the supervised RE system for improved performance. The similar problem was addressed by Sun [118] by using word clustering. In this approach, a large unlabelled corpus was used to learn word clusters, so that the words occurring in the similar context are grouped together in the same cluster. This is very useful to discover many new words (i.e. words not observed in the limited labelled data) which can be crucial to properly classify the relations. Features based on these word clusters were incorporated into the supervised RE system to achieve better performance. For mention level RE, Pawar et al. [93] proposed a semi-supervised approach using EM algorithm.

It should be noted that the performance of bootstrapping based algorithm depends on the choice of initial *seed* examples. Analysis of quality of *seeds* chosen in bootstrapping algorithms is provided by Vyas et al. [123] and Kozareva and Hovy [65].

4.2. Active Learning

Active Learning [111] techniques are being widely used by Machine Learning community in order to reduce annotation effort required to create labelled data. The key idea behind active learning is that the learning algorithm is allowed to ask for true labels of some selected unlabelled instances. Various criterion have been proposed to choose these instances with the common objective of learning the underlying hypothesis quickly with a very few instances. The key advantage of active learning is that performance comparable with supervised methods is achieved through a very few labelled instances.

Sun and Grishman [117] presented an active learning system *LGCo-Testing*. It is based on an active learning approach Co-testing [52] in the co-training [10] setting. For applying Co-testing, the authors proposed to create two views of the relation instances - i) a local view based on the features capturing the entity mentions being connected and other characteristics of the containing sentence;

and ii) a global view based on the distributional similarity of the phrases connecting two entity mentions, using a large corpus. Suppose, for an instance of `PHYS` type, the connecting phrase is `travelled to`, then examples of other phrases similar to it are `arrived in`, `visited`, etc. Distributional similarity would assign a high similarity between two phrases, if these phrases are observed in the similar context in a large corpus. A Maximum Entropy classifier is trained using the features from the local view. As a classifier using global view, a nearest neighbour classifier is used which uses the distributional similarity to find the nearest neighbours. The *LGCo-testing* was further improved in terms of efficiency by Fu and Grishman [38]. Recently, a bilingual active learning approach for RE was proposed by Yanan et al. [136] and the two languages were Chinese and English. Some of the other approaches using Active Learning for RE, are by Small and Roth [106] and Zhang et al. [145].

4.3. Label Propagation Method

Label Propagation is a graph based semi-supervised method proposed by Zhu and Ghahramani [130] where labelled and unlabelled instances in the data are represented as nodes in a graph with edges reflecting the similarity between nodes. In this method, the label information for any node is propagated to nearby nodes through weighted edges iteratively and finally the labels of unlabelled examples are inferred when the propagation process is converged. The first attempt of using Label Propagation method for RE was by Chen et al. [20]. They represented each entity pair (i.e. relation instance) in the dataset as a node in a graph and also associate a feature vector with it. The feature vector consists of various features characterizing the relation instance as described in the section 2.1. The graph is completely connected with each edge between relation instances R_i and R_j having following weight,

$$W_{ij} = \exp(\frac{s_{ij}}{\sigma^2})$$

Here, s_{ij} is the similarity between the feature vectors associated with R_i and R_j. σ^2 is used to scale the weights, which the authors set to average similarity between labelled instances. Considering feature vector as probability distribution over features, the authors use JS divergence to compute distance between any two relation instances. Similarity between two instances is then inversely proportional to this distance. Chen et al. [20] observed that this algorithm performed better than SVM and bootstrapping approaches. One of the major advantages of the label propagation is that the labels of the unlabelled instances are not only decided by the nearby labelled instances but also by the nearby unlabelled instances.

4.4. Other Methods

Jiang [55] applied multi-task transfer learning to solve a weakly-supervised RE problem. This special problem setting is that only a few seed instances of the relation type of interest are available but a large amount of labelled instances of other relation types is also available. The author observed that different

relation types can share certain common structures, e.g. ACE relations EMP-ORG and GPE-AFF share the common syntactic structure where two entity mentions are connected through the preposition `of` (EMP-ORG : `employees of TCS`; GPE-AFF : `residents of India`). The proposed framework uses a multi-task transfer learning method along with human guidance in the form of entity type constraints. The commonality among different relation types is modelled through a shared weight vector, enabling the knowledge learned from other relation types to be transferred to the target relation type.

4.5. Evaluation

Some of the semi-supervised techniques focus on mention-level RE (e.g. [20, 116, 118]) and these can be evaluated in the similar manner as that of supervised techniques given some labelled dataset is available. Most of the bootstrapping based techniques (like [12, 1]) do not attempt to capture every mention of entity pairs, rather these techniques create a list of entity mention pairs exhibiting a particular relation type. While *precision* can be measured easily by verifying all the extracted pairs, it is difficult to estimate the *recall* as the number of true relation mentions in the unlabelled data is not available. In order to measure the *recall*, a smaller subset of unlabelled data can be considered and all the relation mentions within it can be manually identified.

5. Unsupervised Relation Extraction

In this section, we discuss some of the important unsupervised RE approaches which do not require any labelled data.

5.1. Clustering based approaches

One of earliest approaches for completely unsupervised RE was proposed by Hasegawa et al. [48]. They only require a NER tagger to identify named entities in the text so that the system focuses only on those named entity mentions. The approach can be described in following steps:

1. The named entities in the text corpora are tagged
2. Co-occurring named entity pairs are formed and their contexts are recorded
3. Context similarities among the pairs identified in the step 2, are computed
4. Using the similarity values computed in previous step, the pairs are clustered
5. As each of these clusters represent one relation, a label is automatically assigned to each cluster describing the relation type represented by it

Named Entity (NE) pairs and context: Two named entities are said to be co-occurring if there are at most N intermediate words in between them. Pairs of all such co-occurring named entities are formed. All occurrences of a particular NE pair are observed and *all* the intermediate words for *all* such occurrences are recorded as the context for that NE pair. The words occurring to the left of first NE and the words occurring to the right of second NE are not considered to be part of the context. This is one of the limitations of this approach, as not all relations are expressed through using only intermediate words, e.g. `CEO`

`of ORG, PER announced the financial results.` Also, the order of NEs is given importance, i.e. the pair (NE_1, NE_2) is considered to be different than (NE_2, NE_1) and their contexts are also recorded separately.

Context similarity computation: For each NE pair, a word vector is formed using all the words occurring in its context. Each word is weighted by $TF \times IDF$, where TF is frequency of the word in the context and IDF is inverse document frequency which is inversely proportional to number of NE pairs in whose context the word occurs. The authors use an interesting way to encode the order of NEs in the computation of TF. If a word w occurs L times in the context of (NE_1, NE_2) and M times in the context of (NE_2, NE_1), then in the word vector for (NE_1, NE_2), TF of the word w is computed as $L - M$. The intuition behind this is that this would be effective to detect the direction of a relation if the arguments of a relation have the same NE types. If the NE types of two NE pairs do not match, the similarity between the two pairs is considered to be 0. The similarity of the contexts of two NE pairs is computed as the *Cosine Similarity* between their word vectors. The similarity value varies from -1 to 1, where 1 indicates that the contexts of two NE pairs are matching exactly and NEs occur in the same order. The similarity -1 indicates that the NE pairs have exactly the same context words but the order of NEs in them is reverse.

Clustering and Labelling: Using the similarity values, the NE pairs are clustered using hierarchical clustering with complete linkage. The resultant clusters are also labelled automatically using the high frequency words in the contexts of all the NE pairs in the cluster.

Chen et al. [19] proposed some improvements in Hasegawa et al.'s [48] basic clustering approach. They developed an unsupervised feature selection method to remove uninformative noisy words from similarity computation. Another similar approach for unsupervised RE from Wikipedia texts, was proposed by Yan et al. [135]. Here, instead of NE pairs, they form *Concept* pairs by using Wikipedia structure. For a Wikipedia article, its title becomes the *principal* concept and it is paired with other *secondary* concepts linking the current article to other Wikipedia articles. They proposed a two-step clustering approach from grouping the concept pairs with same relation type. In the first step, the concept pairs are clustered using similarity of the deep linguistic patterns linking the two concepts. These linguistic patterns are derived from the dependency trees of the containing sentences. Given the high quality of Wikipedia content, these patterns are usually more precise than the surface words context similarity computed in Hasegawa et al. [48]. Once these highly precise clusters are formed in the first step, in order to improve the coverage, the second step of clustering is carried out on the remaining unclustered concept pairs. This clustering step uses the cluster centroids created in the first step and is based on simple surface patterns learned from larger Web corpus.

Another interesting line of research , is based on inducing relation types by generalizing dependency paths. Lin and Pantel [69] proposed DIRT (Discovery of Inference Rules) algorithm which is based on distributional similarity hypothesis. Rather than applying this hypothesis for discovering similar words, the authors use it to discover similar dependency paths which tend to link

the same set of words. Poon and Domingos [96] presented a similar approach for USP (Unsupervised Semantic Parsing), which recursively clusters fragments of dependency trees such that various syntactic variations conveying the same meaning are grouped together. This idea of clustering expressions conveying the same meaning was extended to bilingual case by Lewis and Steedman [67] for English and French. They learned the clusters of semantically similar English and French expressions representing some relations. Rather than using bilingual parallel corpus, the authors exploited the alignments between named entities in two languages. Another method for clustering of relation instances was proposed by Yao et al. [138]. Their method makes use of generative probabilistic models, similar to LDA based topic models [29]. These models represent relation instances in the form of entity mention types and various features based on connecting dependency path, which are generated by underlying hidden relation types. Here, the relation types play the role similar to underlying topics in the usual topic models for documents. The model also incorporates constraints between relation type and types of the entity mentions. The topic models proposed by Yao et al. were extended by de Lacalle and Lapata [31] by integrating them with general domain knowledge. The domain knowledge is encoded as First Order Logic (FOL) rules which apply *Must-link* and *Cannot-link* constraints on the relation instances.

5.2. Other approaches

One of the major non clustering based approach for unsupervised relation extraction is the URES (Unsupervised RE System) by Rosenfeld and Feldman [104]. The only input required by the URES system is the definitions of the relation types of interest. A relation type is defined as a small set of keywords indicative of that relation type and entity types of its arguments. e.g. for the relation type *Acquisition*, the keywords can be `acquired, acquisition`. The URES system is a direct successor of the KnowItAll system [90] which extracts facts from the web. The focus of KnowItAll is primarily on extracting entities and URES builds on that to extract relations. Feldman and Rosenfeld [37] further boosted the performance of URES by introducing a simple rule based NER. Another interesting approach was proposed by Romano et al. [103] which is based on using unsupervised paraphrase acquisition for RE. The text expressions that convey roughly the same meaning, are called as *paraphrases*. The approach begins with one text expression (and corresponding syntactic structure like dependencies structure) representing the target relation and finds its paraphrases using an unsupervised paraphrase acquisition approach. For example, starting from the initial expression `X interact with Y`, paraphrase acquisition algorithm would produce new expressions - `X bind to Y`, `X activate Y`, `X stimulate Y`, `interaction between X and Y`, etc.

6. Open Information Extraction

Traditional RE focuses on precise, pre-specified set of relations. Extensive human involvement is generally required in designing extraction rules or for

creating labelled training data. Hence, it is difficult make such systems work in a different domain. To overcome these limitations, the paradigm of Open Information Extraction (Open IE) was first proposed by Banko et al. [6], in the form of the TextRunner system. Open IE systems automatically discover possible relations of interest using the text corpus without any human involvement. Hence, no additional efforts are required to switch to a different domain.

The TextRunner system [6] consists of following three core modules:

1. Self-supervised Learner: Using some heuristic rules, it automatically labels a set of extracted entity tuples as positive or negative. Here, positive class indicates that the corresponding tuple represents some valid relation. After automatic labelling, each tuple is mapped to a feature vector representation and a Nave Bayes classifier is trained.

2. Single Pass Extractor: It makes a single pass over entire corpus and obtains POS and NP (base noun phrases) information for all sentences. For each sentence, each pair of NPs (E_1 and E_2) becomes a candidate tuple and the corresponding relation string R is found by examining the text in between. For each word occurring in between, it is heuristically decided whether to include it in R. Each candidate tuple is presented to the Nave Bayes classifier and only those tuples which are classified as "positive" are extracted and stored.

3. Redundancy-based Assessor: After extractions are performed over entire corpus, TextRunner automatically merges some of the tuples where both the entities and relations are identical. For each tuple, number of distinct sentences containing it is also recorded and the assessor then uses these counts to assign a probability of correctness to each tuple.

Banko et al. [7] proposed to use Conditional Random Field based, self-supervised sequence classifier O-CRF instead of Naive Bayes classifier used in TextRunner and observed better performance. Another improvement to TextRunner was suggested by Wu and Weld [129] in their Wikipedia-based Open Extractor (WOE) system. They used Wikipedia infoboxes to more accurately generate training data for the Self-supervised Learner module. Similar approach of using Wikipedia infoboxes was adopted by Weld et.al. [128] in their *Kylin* open IE system. Bootstrapping methods like Snowball [1] significantly reduce the number of initial training examples, but these methods do not perform open IE. Zhu et al. [155] proposed a bootstrapping approach *StatSnowball* which can even perform open IE along with traditional RE.

Fader et al. [35] proposed *ReVerb*, an advanced Open IE system which improves over TextRunner by overcoming following limitations of TextRunner:

1. Incoherent Extractions: No meaningful interpretation of extracted relation phrases can be made. Such extractions are result of a word-by-word decision making about whether to include a word in a relation phrase.

2. Uninformative Extractions: These extractions omit critical information and are generally caused by improper handling of relation phrases that are expressed by Light Verb Constructions (LVCs). LVCs are multi-word expressions composed of a verb and a noun, with the noun carrying semantic content. e.g. `is the author of`. From the sentence `John made a promise to Alice`, the TextRunner makes an uninformative extraction `(John, made, a promise)`

whereas correct extraction is `(John, made a promise to, Alice)`.
3. Overly-specific Extractions: TextRunner may extract very specific relation phrases which are not useful. e.g. `(The Obama administration, is offering only modest greenhouse gas reduction targets at, the conference)`

To overcome the above limitations, the *ReVerb* algorithm proposes following two constraints on relation phrases to be extracted.
Syntactic Constraint: The relation phrases are constrained to match the POS tag pattern mentioned in the Table 7. This constraint limits relation phrases to be either one of following: a verb (e.g. `invented`); a verb immediately followed by a preposition (e.g. `born at`); a verb followed by nouns, adjectives or adverbs ending in preposition (e.g. `has atomic weight of`); multiple adjacent matches merged into a single relation phrase (e.g. `wants to extend`) Incoher-

$V \mid VP \mid VW^*P$
$V = verb\ particle?\ adv?$
$W = (noun \mid adj \mid adv \mid pron \mid det)$
$P = (prep \mid particle \mid inf.\ marker)$

Table 7: Syntactic Constraint

ent extractions are avoided because there are no isolated word-level decisions about whether to include a word in relation phrase. The decision is taken for a sequence of words whether that sequence satisfies the POS pattern. Uninformative extractions are avoided because nouns are also allowed as a part of relation phrase and relations expressed via LVCs are also captured.
Lexical Constraint: To avoid overly-specific relation phrases, a lexical constraint is applied that considers only those relation phrases as valid which take at least k distinct argument pairs. A valid relation phrase like `took control over` occurs with multiple distinct arguments like `(Germany, took control over, Austria)` and `(Modi, took control over, administration)`.

ReVerb differs from TextRunner in the manner in which the relation phrases are identified. Relation phrases are identified "holistically" rather that the word-by-word decision in TextRunner, resulting in more meaningful relation phrases. *ReVerb* follows a "relations first" approach rather than TextRunner's "arguments first" approach, as it first identifies a valid relation phrase and then extracts the arguments. This helps not to confuse a noun in relation phrase as an argument. e.g. `promise` in `made a promise to`.

Etzioni et al. [34] observed that almost 65% of *ReVerb*'s extraction had a correct relation phrase but an incorrect arguments. They proposed an Open IE system R2A2 which is an improvement over *ReVerb* and contains an additional module *ArgLearner* for identifying arguments. *ReVerb* is found to outperform all the previous open IE systems like TextRunner and WOE. And R2A2 was observed to achieve better precision and recall than even *ReVerb*. In order to be efficient on the web scale, most of the open IE systems do not perform deep syntactic analysis. Gamallo [40] used robust and fast dependency parsing in their open IE system *DepOE* on the Web scale to achieve more precise extraction than *ReVerb*. Mesquita et al. [72] presented a comprehensive comparison of 8 Open Information Extraction techniques for their efficiency and effective-

ness. They analysed the trade-off between the complexity of NLP processing (i.e. from simpler POS tagging to more complex Semantic Role Labelling) versus effectiveness. Some of the major advantages of Open IE systems are their unsupervised nature and scalability to the Web scale. Recently, Open IE has been an active area of research within RE systems. One of the major limitation of these systems is that the same semantic relation may be represented by multiple relation phrases and some post-processing is required to consolidate such various representations of the same relation type.

7. Distant Supervision

Distant Supervision, proposed by Mintz et al. [75], is an alternative paradigm which does not require labelled data. The idea is to use a large semantic database for automatically obtaining relation type labels. Such labels may be noisy, but the huge amount of training data is expected to offset this noise. Similar ideas of creating "weakly" labelled training data, were earlier proposed by Craven and Kumlien [26], Bunescu and Mooney [13] and Nguyen et al. [83]. Distant Supervision combines advantages of both the paradigms : Supervised and Unsupervised. It combines thousands of features using a probabilistic classifier as in the case of supervised paradigm. Also, it extracts a large number of relations from large corpora of any domain as in the case of unsupervised paradigm. Mintz et al. [75] used Freebase [11] as a semantic database which stores pairs of entities for various relations.

Labelling heuristic: If two entities participate in a relation, any sentence that contains both of them might express that relation. For example, Freebase contains entity pair `<M. Night Shyamalan, The Sixth Sense>` for the relation /film/director/film, hence both of the following sentences are considered to be positive examples for that relation:

1. `M. Night Shyamalan gained international recognition when he wrote and directed 1999's The Sixth Sense.`
2. `The Sixth Sense is a 1999 American supernatural thriller drama film written and directed by M. Night Shyamalan.`

Negative Instances: The above mentioned heuristic can be used to obtain only positive instances for various relation types but not the negative instances. In order to train a classifier both the types of instances are necessary. Entity pairs which do not appear in any Freebase relation are randomly selected and treated as negative instances. Some entity pairs may be incorrectly labelled as negative due to incompleteness of Freebase.

Using the automatically obtained labelled data, a multi-class Logistic Classifier with Gaussian regularization is trained. Various lexical, syntactic and named entity type features are used for training. There were several subsequent efforts to improve upon the approach by Mintz et al. [75] like Yao et al. [139], Hoffmann et al. [51], Reidel et al. [101], Nguyen and Moschitti [87], Takamatsu et al. [122] and Krause et al. [66].

One of the major shortcoming of the traditional distant supervision based approaches was that they failed to model overlapping relations, i.e. the fact

that for the same pair of entities, there can be multiple valid relations. e.g. *FoundedBy*(`Steve Jobs, Apple`) and *CEO*(`Steve Jobs, Apple`). Two of the major approaches to handle this problem were proposed by Hoffmann et al. [50] and Surdeanu et al. [120]. The Multi-instance Multi-label learning based approach (MIML-RE) by Surdeanu et al. [120], models latent relation labels for multiple instances (occurrences) of an entity pair. It also models dependencies between labels for a single entity pair.

MIML-RE uses a novel graphical model for representing "multiple instances" as well as "multiple labels" of an entity pair. A mention level relation classifier is used to identify relation label for each *mention* of an entity pair using features derived from the context of the mention. There is another set of classifiers, one per each distinct relation label, which operate at the entity pair level. These are binary classifiers indicating whether a specific relation holds for an entity pair. These classifiers can learn that two relation labels like *BornIn* and *SpouseOf* cannot be generated jointly for the same entity pair. If mention level classifier in the lower layer assigns both of these labels for different mentions of the same tuple, then one of them can be cancelled by the Entity pair level classifiers. It can also learn when the two labels tend to appear jointly, like *CapitalOf* and *ContainedIn*. In order to learn various parameters of this graphical model, hard discriminative EM algorithm is used.

MIML-RE outperforms not only the traditional distant supervision approach by Mintz et al. [75] but also the approach modelling multiple instances by Hoffmann et al. [50]. One of the major advantage that MIML-RE has over the Hoffmann et al. [50] is its entity pair level classifiers. The datasets used for evaluation are Riedel's dataset [101] and KBP dataset. The KBP dataset was constructed by Surdeanu et al. [120] using the resources distributed for the 2010 and 2011 KBP shared tasks [54, 53]. These datasets are widely used in the distant supervision related literature.

Recently, distant supervision for RE has become a very active field of research with several new approaches to overcome some of the specific problems. Due to incompleteness of semantic database used for labelling by distant supervision, many *negative* instances are actually false negatives. To overcome this problem, Min et al. [74] proposed an algorithm which learns only from positive and unlabelled examples. Xu et al. [134] addressed this problem of false negative training examples by adapting the information retrieval technique of pseudorelevance feedback. Zhang et al. [143] analyzed comparative performance of distant supervision and crowd-sourcing which is also an alternative low-cost method of obtaining labelled data. Pershina et al. [94] proposed a *Guided DS* approach which shows that when small amount of human labelled data is available along with distantly labelled, a significant improvement in RE performance is observed. One problem with MIML-RE's data likelihood expression is that it is a non-convex formulation. Grave [42] proposed a new approach based on discriminative clustering which leads to a convex formulation. Other recent approaches with various improvements over the basic distant supervision approach, are proposed by Zhang et al. [149], Chen et al. [21], Han and Sun [46], Nagesh et al. [82], Liu [137] and Koch et al. [64].

Distant supervision can not be applied if the relation of interest is not covered explicitly by the knowledge base (like FreeBase). Zhang et al. [144] proposed a novel approach of *Ontological Smoothing* to address this problem when at least some seed examples of the relation of interest are available. *Ontological Smoothing* generates a mapping between the relation of interest and the knowledge-base. Such mappings are used to generate additional training examples along with seed examples and distant supervision is then used to learn the relation extractor. An interesting study carried out by Nguyen et al. [88], reported that joint distant and direct supervision can significantly improve the RE performance as compared to the systems which use only gold-standard labelled data like ACE 2004. They mapped some YAGO [115] relation types to the seven ACE 2004 relation types and created a distantly supervised labelled dataset using Wikipedia text. Two separate relation classifiers were trained - one using only ACE 2004 labelled data and other using distantly supervised labelled data combined with ACE 2004 data. The linear combination of the probabilities obtained from both of these classifiers was considered as the final probability. This joint classifier provided around 3% improvement in F-measure compared to the classifier using only ACE 2004 training data.

A first attempt of using Active Learning with distantly supervised RE, was reported by Angeli et al. [3]. They proposed to provide a partial supervision to MIML-RE with the help of active learning. A novel selection criteria was proposed for selecting relation instances for labelling. This criteria prefers relation instances which are both uncertain (high disagreement in committee of classifiers) and representative (similar to large number of unlabelled instances). The annotations were obtained through crowdsourcing from Amazon Mechanical Turk and a significant improvement over MIML-RE was observed.

8. Recent Advances in Relation Extraction

In this section, we describe some recent advances in the field of RE.

Universal Schemas: Riedel et al. [102] proposed to use Universal Schemas, which are union of relation types of existing structured databases (e.g. FreeBase, Yago) and all possible relation types in the form of surface forms used in Open IE. They proposed an approach to learn asymmetric implicature among these universal relation types. This implicature helps to infer possible new relations between an entity pair given a set of existing relations for that entity pair from some structured database. If a city and a country are known to be related with a relation type *CapitalOf*, it can be inferred that the relation type *LocatedIn* also holds true for that pair, but not vice versa as the implicature is asymmetric. Other similar approaches were proposed by Chang et al. [18] and Fan et al. [36].

n-ary Relation Extraction: The relations among more than two entities are generally referred to as *Complex* or *Higher Order* or *n-ary* relations. Example of an n-ary relation is `EMP-ORG-DESG` which represents relation between a person, the organization where he/she is employed and his/her designation. This relation exists for entities `(John Smith, CEO, ABC Corp.)` in the sentence : `John Smith is the CEO of ABC Corp.` One of the earliest attempt to address

this problem was by McDonald et al. [71]. They used well-studied binary RE to initially find relations between all possible entity pairs. The output of this binary RE was represented as a graph where entities correspond to nodes and valid relations between entities correspond to edges. The authors then proposed to find maximal cliques in this graph such that each clique corresponds to some valid n-ary relation. They demonstrated effectiveness of this approach on the Biomedical domain dataset. Another recent approach for n-ary RE is an Open IE system named Kraken proposed by Akbik and Löser [2]. Zhou et al. [152] surveyed several complex RE approaches in the biomedical domain.

Another perspective to look at n-ary RE problem as Semantic Roles Labelling (SRL). The SRL task is to identify predicate and its arguments in a given sentence automatically. One of the fundamental SRL approach is by Gildea and Jurafsky [41] and some standard semantic roles labelled datasets are PropBank [63] and FrameNet [5].

Cross-sentence Relation Extraction: Most of the techniques that we have discussed, focus on intra-sentential RE, i.e. entity mentions constituting a relation instance occur in the same sentence. Swampillai and Stevenson [121] proposed an approach to extract both intra-sentential and inter-sentential relations. Some examples of an inter-sentential relation is shown in the Table 8.

No.	Sentences	Relation
1	The youngest son of ex-dictator Suharto disobeyed a summons to surrender himself to prosecutors Monday and be imprisoned for corruption.	PER-SOC
	Hutomo ‘‘Tommy’’ Mandala Putra, 37, was sentenced to 18 months in prison on Sept. 22 by the Supreme Court, which overturned an earlier acquittal by a lower court.	
2	Computer intruders broke into Microsoft Corp. and were able to view some of the company’s source code, the basic program instructions, for a future software product, the company said Friday.	EMP-ORG
	‘‘The situation appears to be narrower than originally thought,’’ said the spokesman, Mark Murray.	

Table 8: Examples of relations spanning across two consecutive sentences. The entity mentions between which the relation holds, are underlined.

The authors adapted the structured features (like parse tree paths) and techniques for intra-sentential RE for the inter-sentential situation. Generally, it can be observed that most of the cases (like Example 1 in table 8 but not Example 2) of inter-sentential RE can be addressed through co-reference resolution [33]. In Example 1, `son` in the first sentence actually refers to `Hutomo ‘‘Tommy’’ Mandala Putra` in the second sentence. Given that intra-sentential RE technique can detect `PER-SOC` relation between `son` and `Suharto`, using the co-reference we can get the required inter-sentence relation.

Convolutional Deep Neural Network: Zeng et al. [142] explored the feasibility of performing RE without any complicated NLP preprocessing like parsing. They employed a convolutional DNN to extract lexical and sentence level features. They observed that the automatically learned features yielded excellent results and can potentially replace the manually designed features that are

based on the various pre-processing tools like syntactic parser. Other similar techniques which use Recursive Neural Networks were proposed by Socher et al. [113] and Hashimoto et al. [49].

Cross-lingual Annotation Projection: Entities and relations labelled data is available only for a few *resource-rich* languages like English, Chinese and Arabic. Kim et al. [61] proposed a technique to project relation annotations from a *resource-rich* source language (English) to a *resource-poor* target language (Korean) by utilizing parallel corpus. Direct projection was used in the sense that the projected annotations were determined in a single pass by considering only alignments between entity candidates. Kim and Lee [62] proposed a graph-based projection approach which utilizes a graph that is constructed with both entities and context information and is operated in an iterative manner.

Domain Adaptation: The fundamental assumption of the supervised systems is that the training data and the test data are from the same distribution. But when there is a mismatch between these data distributions, the RE performance of supervised systems tends to degrade. This generally happens when supervised systems are used to classify out-of-domain data. In order to address this problem, domain adaptation techniques are needed. The first such study for RE was carried out by Plank and Moschitti [95]. They reported that the out-of-domain performance of kernel-based systems can be improved by embedding semantic similarity information obtained from word clustering and latent semantic analysis (LSA) into syntactic tree kernels. Nguyen and Grishman [86] proposed an adaptation approach which generalizes lexical features using both word cluster and word embedding [8] information. Another approach by Nguyen et al. [85] proposed to use only the relevant information from multiple source domains which results in accurate and robust predictions on the unlabelled target-domain data.

9. Conclusion and Future Research Directions

To the best of our knowledge, we presented a first comprehensive survey of relation extraction techniques. We clarified the usage of the term "Relation Extraction" which can refer to either mention-level RE (ACE RDC task) and global RE. We first described supervised techniques including important feature-based and kernel based techniques. We discussed how these techniques evolved over a period of time and how they are evaluated. We observed that among all the supervised techniques, syntactic tree kernel based techniques were the most effective. They produced the best results either when combined with some other kernel to form a composite kernel or when dynamically determined tree span was used. The best reported result after almost a decade of efforts on the ACE 2004 dataset is around 77%. Hence, we think that there is still some room for improvement here.

We also covered joint modelling techniques which jointly extract entity mentions and relations between them. From a practical point of view, this problem is quite important because good entity extraction performance is a must for achieving good RE performance. The joint modelling techniques allows two-way

information flow between these tasks and try to a achieve better performance for both as compared to isolated models. We then focussed on some of the important semi-supervised and unsupervised techniques. There has been a lot of work in these areas, as increasing amount of efforts are being put to reduce the dependence on the labelled data. We also covered the paradigms of Open IE and Distant Supervision based RE, which require negligible human supervision. Recently, there has been a continuously increasing trend in RE research towards distant supervision based techniques.

Though the state-of-the-art for RE has improved a lot in the last decade, there are still many promising future research directions in RE. We list some of these potential directions below:

1. There have been several techniques for joint modelling of entity and relation extraction. However, the best reported F-measure on ACE 2004 dataset when gold-standard entities are not given, is still very low at around 48%. This almost 30% lower than the F-measure achieved when gold-standard entity information is assumed. Hence, there is still some scope of improvement here with more sophisticated models.
2. There has been little work for extracting n-ary relations, i.e. relations involving more than two entity mentions. There is a scope for more useful and principled approaches for this.
3. Most of the RE research has been carried out for English, followed by Chinese and Arabic, as ACE program released the datasets for these 3 languages. It would be interesting to analyse how effective and language independent are the existing RE techniques. More systematic study is required for languages with poor resources (lack of good NLP pre-processing tools like POS taggers, parsers) and free word order, e.g. Indian languages.
4. Depth of the NLP processing used in most of the RE techniques, is mainly limited to lexical and syntax (constituency and dependency parsing) and few techniques use light semantic processing. It would be quite fruitful to analyse whether deeper NLP processing such as semantics and discourse level can help in improving RE performance.

For new entrants in the field, this survey would be quite useful to get introduced to the RE task. They would also get to know about various types of RE techniques and evaluation methods. This survey is also useful for the practitioners as they can get a quick overview of the RE techniques and decide which technique best suits their specific problem. For researchers in the field, this survey would be useful to get an overview about most of the RE techniques proposed in last decade or so. They can learn about how the techniques evolved over time, what are the pros and cons of each technique and what is the relative performance of these techniques. We have also pointed out some of the recent trends in RE techniques and which would be useful for the researchers. We have also listed some open problems in RE which can lead to new research in future.

Acknowledgement

The authors would like to thank Swapnil Hingmire for his efforts of reviewing the draft and providing several useful suggestions for improvement.

References

[1] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.

[2] Alan Akbik and Alexander Löser. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 52–56. Association for Computational Linguistics, 2012.

[3] Gabor Angeli, Julie Tibshirani, Jean Y Wu, and Christopher D Manning. Combining distant and partial supervision for relation extraction. In *Proc. The 2014 Conference on Empirical Methods on Natural Language Processing*, 2014.

[4] Nguyen Bach and Sameer Badaskar. A survey on relation extraction. *Language Technologies Institute, Carnegie Mellon University*, 2007.

[5] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[6] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[7] Michele Banko, Oren Etzioni, and Turing Center. The tradeoffs between open and traditional relation extraction. In *ACL*, volume 8, pages 28–36, 2008.

[8] Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, pages 932–938, 2001.

[9] Sebastian Blohm and Philipp Cimiano. Using the web to reduce data sparseness in pattern-based information extraction. In *Knowledge Discovery in Databases: PKDD 2007*, pages 18–29. Springer, 2007.

[10] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT : Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann Publishers, 1998.

[11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[12] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999.

[13] Razvan Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Annual meeting-association for Computational Linguistics*, volume 45, page 576, 2007.

[14] Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

[15] Andrew Carlson, Justin Betteridge, Estevam R Hruschka Jr, and Tom M Mitchell. Coupling semi-supervised learning of categories and relations. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, pages 1–9. Association for Computational Linguistics, 2009.

[16] Yee Seng Chan and Dan Roth. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160. Association for Computational Linguistics, 2010.

[17] Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 551–560. Association for Computational Linguistics, 2011.

[18] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579, 2014.

[19] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Unsupervised feature selection for relation extraction. In *Proceedings of IJCNLP*, 2005.

[20] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 129–136. Association for Computational Linguistics, 2006.

[21] Liwei Chen, Yansong Feng, Songfang Huang, Yong Qin, and Dongyan Zhao. Encoding relation requirements for relation extraction via joint inference. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 818–827, 2014.

[22] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439. Association for Computational Linguistics, 2006.

[23] Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71, 2005.

[24] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in neural information processing systems*, pages 625–632, 2001.

[25] Michael Collins and Scott Miller. Semantic tagging using a probabilistic context free grammar. Technical report, DTIC Document, 1998.

[26] Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86, 1999.

[27] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

[28] W. Yih D. Roth. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8, 2004.

[29] Andrew Ng David Blei and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

[30] Sandra Collovini de Abreu, Tiago Luis Bonamigo, and Renata Vieira. A review on relation extraction with an eye on portuguese. *Journal of the Brazilian Computer Society*, 19(4):553–571, 2013.

[31] Oier Lopez de Lacalle and Mirella Lapata. Unsupervised relation extraction with general domain knowledge. In *EMNLP*, pages 415–425, 2013.

[32] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation.

[33] Pradheep Elango. Coreference resolution: A survey. *University of Wisconsin, Madison, WI*, 2005.

[34] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[35] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[36] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y Chang. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 839–849, 2014.

[37] Ronen Feldman and Benjamin Rosenfeld. Boosting unsupervised relation extraction by using ner. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 473–481. Association for Computational Linguistics, 2006.

[38] Lisheng Fu and Ralph Grishman. An efficient active learning framework for new relation types. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP), Nagoya, Japan*, pages 692–698, 2013.

[39] Ryan Gabbard, Marjorie Freedman, and Ralph Weischedel. Coreference for learning to extract relations: yes, virginia, coreference matters. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 288–293. Association for Computational Linguistics, 2011.

[40] Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 10–18. Association for Computational Linguistics, 2012.

[41] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

[42] Edouard Grave. A convex relaxation for weakly supervised relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1580–1590, 2014.

[43] Mark A Greenwood and Mark Stevenson. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 29–35. Association for Computational Linguistics, 2006.

[44] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual*

meeting on association for computational linguistics, pages 427–434. Association for Computational Linguistics, 2005.

[45] Ben Hachey, Claire Grover, and Richard Tobin. Datasets for generic relation extraction. *Natural Language Engineering*, 18(01):21–59, 2012.

[46] Xianpei Han and Le Sun. Semantic consistency: A local subspace based method for distant supervised relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 718–724, 2014.

[47] Sanda Harabagiu, Cosmin Adrian Bejan, and Paul Morarescu. Shallow semantics for relation extraction. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1061–1066. Morgan Kaufmann Publishers Inc., 2005.

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

[49] Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Simple customization of recursive neural networks for semantic relation classification. In *EMNLP*, pages 1372–1376, 2013.

[50] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

[51] Raphael Hoffmann, Congle Zhang, and Daniel S Weld. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295. Association for Computational Linguistics, 2010.

[52] Steve Minton Ion Muslea and Craig Knoblock. Selective sampling with redundant views. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence.*, 2000.

[53] Heng Ji, Ralph Grishman, and Hoa Dang. Overview of the tac 2011 knowledge base population track. In *Proceedings of the Text Analytics Conference*, 2011.

[54] Heng Ji, Ralph Grishman, Hoa Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *Proceedings of the Text Analytics Conference*, 2010.

[55] Jing Jiang. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1012–1020. Association for Computational Linguistics, 2009.

[56] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *HLT-NAACL*, pages 113–120, 2007.

[57] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004*, 2004.

[58] Nanda Kambhatla. Minority vote: at-least-n voting improves recall for extracting relations. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 460–466. Association for Computational Linguistics, 2006.

[59] Rohit J Kate and Raymond J Mooney. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 203–212. Association for Computational Linguistics, 2010.

[60] Mahdy Khayyamian, Seyed Abolghasem Mirroshandel, and Hassan Abolhassani. Syntactic tree-based relation extraction using a generalization of collins and duffy convolution tree kernel. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 66–71. Association for Computational Linguistics, 2009.

[61] Seokhwan Kim, Minwoo Jeong, Jonghoon Lee, and Gary Geunbae Lee. A cross-lingual annotation projection approach for relation detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 564–571. Association for Computational Linguistics, 2010.

[62] Seokhwan Kim and Gary Geunbae Lee. A graph-based cross-lingual projection approach for weakly supervised relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 48–53. Association for Computational Linguistics, 2012.

[63] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*. Citeseer, 2002.

[64] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S Weld. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901, 2014.

[65] Zornitsa Kozareva and Eduard Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 618–626. Association for Computational Linguistics, 2010.

[66] Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. Large-scale learning of relation-extraction rules with distant supervision from the web. In *The Semantic Web–ISWC 2012*, pages 263–278. Springer, 2012.

[67] Mike Lewis and Mark Steedman. Unsupervised induction of cross-lingual semantic relations. In *EMNLP*, pages 681–692, 2013.

[68] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the Association for Computational Linguistics*, 2014.

[69] Dekang Lin and Patrick Pantel. Dirt discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM, 2001.

[70] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[71] Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 491–498. Association for Computational Linguistics, 2005.

[72] Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, 2013.

[73] Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 226–233. Association for Computational Linguistics, 2000.

[74] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, pages 777–782, 2013.

[75] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the*

4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1003–1011. Association for Computational Linguistics, 2009.

[76] et al. Mitchell, Alexis. Tides extraction (ace) 2003 multilingual training data ldc2004t09. web download. Philadelphia: Linguistic Data Consortium, 2004.

[77] et al. Mitchell, Alexis. Ace 2004 multilingual training corpus ldc2005t09. web download. Philadelphia: Linguistic Data Consortium, 2005.

[78] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, 2014.

[79] M.-F. Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Springer, 2006.

[80] Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2005.

[81] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26, 2007.

[82] Ajay Nagesh, Gholamreza Haffari, and Ganesh Ramakrishnan. Noisy or-based model for relation extraction using distant supervision. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1937–1941, 2014.

[83] Dat PT Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Exploiting syntactic and semantic information for relation extraction from wikipedia. In *IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2007)*, 2007.

[84] Dat PT Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from wikipedia using subtree mining. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1414. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[85] Minh Luan Nguyen, Ivor W Tsang, Kian Ming A Chai, and Hai Leong Chieu. Robust domain adaptation for relation extraction via clustering consistency. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 807–817, 2014.

[86] Thien Huu Nguyen and Ralph Grishman. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 68–74, 2014.

[87] Truc-Vien T Nguyen and Alessandro Moschitti. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 277–282. Association for Computational Linguistics, 2011.

[88] Truc-Vien T Nguyen and Alessandro Moschitti. Joint distant and direct supervision for relation extraction. In *IJCNLP*, pages 732–740, 2011.

[89] Truc-Vien T Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1378–1387. Association for Computational Linguistics, 2009.

[90] Doug Downey Ana-Maria Popescu Tal Shaked Stephen Soderland Daniel S.Weld Oren Etzioni, Michael Cafarella and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Journal of Artificial Intelligence*, 2005.

[91] G. K. Palshikar. Techniques for named entity recognition: A survey. In *in S. Bruggemann, C. D'Amato (Ed.s), Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resource*, pages 191–217. IGI Global, 2012.

[92] Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006.

[93] Sachin Pawar, Pushpak Bhattacharyya, and Girish Keshav Palshikar. Semi-supervised relation extraction using em algorithm. 2013.

[94] Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of ACL*, 2014.

[95] Barbara Plank and Alessandro Moschitti. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *ACL (1)*, pages 1498–1507, 2013.

[96] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 1–10. Association for Computational Linguistics, 2009.

[97] Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. Exploiting constituent dependencies for tree kernel-based semantic

relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 697–704. Association for Computational Linguistics, 2008.

[98] Longhua Qian, Guodong Zhou, Qiaomin Zhu, and Peide Qian. Relation extraction using convolution tree kernel expanded with entity features. In *Proceedings of the 21st Pacific Asian Conference on Language, Information and Computation (PACLIC-21)*, pages 415–421, 2007.

[99] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

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

[101] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[102] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, pages 74–84, 2013.

[103] Lorenza Romano, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. Investigating a generic paraphrase-based approach for relation extraction. 2006.

[104] Benjamin Rosenfeld and Ronen Feldman. Ures: an unsupervised web relation extraction system. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 667–674. Association for Computational Linguistics, 2006.

[105] Benjamin Rosenfeld and Ronen Feldman. Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 600, 2007.

[106] Dan Roth and Kevin Small. Active learning for pipeline models. In *AAAI*, pages 683–688, 2008.

[107] Dan Roth and Wen-tau Yih. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[108] Dan Roth and Wen-tau Yih. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, pages 553–580, 2007.

[109] Sunita Sarawagi. Information extraction. *Foundations and trends in databases*, 1(3):261–377, 2008.

[110] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems*, 2004.

[111] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[112] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 1–6. ACM, 2013.

[113] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

[114] et al. Song, Zhiyi. Ace 2007 multilingual training corpus ldc2014t18. web download. Philadelphia: Linguistic Data Consortium, 2006.

[115] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[116] Ang Sun. A two-stage bootstrapping algorithm for relation extraction. In *RANLP*, pages 76–82, 2009.

[117] Ang Sun and Ralph Grishman. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1105–1112. ACM, 2012.

[118] Ang Sun, Ralph Grishman, and Satoshi Sekine. Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 521–529. Association for Computational Linguistics, 2011.

[119] Le Sun and Xianpei Han. A feature-enriched tree kernel for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 61–67, 2014.

[120] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics, 2012.

[121] Kumutha Swampillai and Mark Stevenson. Extracting relations within and across sentences. In *RANLP*, pages 25–32, 2011.

[122] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics, 2012.

[123] Vishnu Vyas, Patrick Pantel, and Eric Crestan. Helping editors choose better seed sets for entity set expansion. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 225–234. ACM, 2009.

[124] et al. Walker, Christopher. Ace 2005 multilingual training corpus ldc2006t06. dvd. Philadelphia: Linguistic Data Consortium, 2006.

[125] Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. Relation extraction with relation topics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1426–1436. Association for Computational Linguistics, 2011.

[126] Chang Wang, Aditya Kalyanpur, James Fan, Branimir K Boguraev, and DC Gondek. Relation extraction and scoring in deepqa. *IBM Journal of Research and Development*, 56(3.4):9–1, 2012.

[127] Mengqiu Wang. A re-examination of dependency path kernels for relation extraction. In *IJCNLP*, pages 841–846, 2008.

[128] Daniel S Weld, Raphael Hoffmann, and Fei Wu. Using wikipedia to bootstrap open information extraction. *ACM SIGMOD Record*, 37(4):62–68, 2009.

[129] Fei Wu and Daniel S Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics, 2010.

[130] Zhu Xiaojin and Ghahramani Zoubin. Learning from labeled and unlabeled data with label propagation. In *CMU CALD tech report CMU-CALD-02-107*, 2002.

[131] Fei-Yu Xu. *Bootstrapping Relation Extraction from Semantic Seeds*. PhD thesis, Saarland University, 2008.

[132] Feiyu Xu, Hans Uszkoreit, Sebastian Krause, and Hong Li. Boosting relation extraction with limited closed-world knowledge. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1354–1362. Association for Computational Linguistics, 2010.

[133] Feiyu Xu, Hans Uszkoreit, and Hong Li. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *ACL*, volume 7, pages 584–591, 2007.

[134] Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *ACL (2)*, pages 665–670, 2013.

[135] Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1021–1029. Association for Computational Linguistics, 2009.

[136] Longhua Qian Haotian Hui Yanan, Hu Guodong Zhou, and Qiaoming Zhu. Bilingual active learning for relation classification via pseudo parallel corpora. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 582–592, 2014.

[137] Liheng Xu Yang Liu, Kang Liu and Jun Zhao. Exploring fine-grained entity type constraints for distantly supervised relation extraction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2107–2116, 2014.

[138] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011.

[139] Limin Yao, Sebastian Riedel, and Andrew McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023. Association for Computational Linguistics, 2010.

[140] Xiaofeng Yu and Wai Lam. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1399–1407. Association for Computational Linguistics, 2010.

[141] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.

[142] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344, 2014.

[143] Ce Zhang, Feng Niu, Christopher Ré, and Jude Shavlik. Big data versus the crowd: Looking for relationships in all the right places. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 825–834. Association for Computational Linguistics, 2012.

[144] Congle Zhang, Raphael Hoffmann, and Daniel S Weld. Ontological smoothing for relation extraction with minimal supervision. In *AAAI*, 2012.

[145] Hong-Tao Zhang, Min-Lie Huang, and Xiao-Yan Zhu. A unified active learning framework for biomedical relation extraction. *Journal of Computer Science and Technology*, 27(6):1302–1313, 2012.

[146] Min Zhang, Jie Zhang, and Jian Su. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 288–295. Association for Computational Linguistics, 2006.

[147] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics, 2006.

[148] Min Zhang, GuoDong Zhou, and Aiti Aw. Exploring syntactic structured features over parse trees for relation extraction using kernel methods. *Information processing & management*, 44(2):687–701, 2008.

[149] Xingxing Zhang, Jianwen Zhang, Junyu Zeng, Jun Yan, Zheng Chen, and Zhifang Sui. Towards accurate distant supervision for relational facts extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[150] Zhu Zhang. Weakly-supervised relation classification for information extraction. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 581–588. ACM, 2004.

[151] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 419–426. Association for Computational Linguistics, 2005.

[152] Deyu Zhou, Dayou Zhong, and Yulan He. Biomedical relation extraction: From binary to complex. *Computational and mathematical methods in medicine*, 2014, 2014.

[153] Guodong Zhou, Longhua Qian, and Jianxi Fan. Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, 180(8):1313–1325, 2010.

[154] GuoDong ZHOU, Min Zhang, Dong Hong Ji, and Qiaoming Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. *EMNLP-CoNLL 2007*, page 728, 2007.

[155] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World wide web*, pages 101–110. ACM, 2009.