RELATION EXTRACTION WITH CLAUSE-BASED OPEN INFORMATION EXTRACTION

by

Duc Thuan Vo

Bachelor of Engineering, Can Tho University, Cantho, Vietnam, 2003

Master of Engineering, Asian Institute of Technology, Pathum Thani, Thailand, 2008

A Dissertation

presented to Ryerson University

in partial fulfillment of the

requirement for the degree of

Doctor of Philosophy

in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Relation Extraction with Clause-based Open Information Extraction

Doctor of Philosophy, 2020

Duc Thuan Vo

Electrical and Computer Engineering

Ryerson University

Abstract

Information Extraction (IE) is one of the challenging tasks in natural language processing. The goal of relation extraction is to discover the relevant segments of information in large numbers of textual documents such that they can be used for structuring data. IE aims at discovering various semantic relations in natural language text and has a wide range of applications such as question answering, information retrieval, knowledge presentation, among others. This thesis proposes approaches for relation extraction with clause-based Open Information Extraction that use linguistic knowledge to capture a variety of information including semantic concepts, words, POS tags, shallow and full syntax, dependency parsing in rich syntactic and semantic structures.

Within the plethora of Open Information Extraction that focus on the use of syntactic and dependency parsing for the purposes of detecting relations, incoherent and uninformative relation extractions can still be found. The extracted relations can be erroneous at times and fail to have a meaningful interpretation. As such, we first propose refinements to the grammatical structure of syntactic and dependency parsing with clause structures and clause types in an effort to generate propositions that can be deemed as meaningful extractable relations. Second, considering that choosing the most efficient seeds are pivotal to the success of the bootstrapping process when extracting relations, we propose an extended clause-based pattern extraction method with self-training for unsupervised relation extraction. The proposed self-training algorithm relies on the clause-based approach to extract a small set of seed instances in order to identify and derive new patterns. Third, we employ matrix factorization and collaborative filtering for relation extraction. To avoid the need for manually pre-defined schemas, we employ the notion of *universal schemas* that is formed as a collection of patterns derived from Open Information Extraction tools as well as from relation schemas of pre-existing datasets. While previous systems have trained relations only for entities, we exploit advanced features from relation characteristics such as clause types and semantic topics for predicting new relation instances. Finally, we present an event network representation for temporal and causal event relation extraction that benefits from existing Open IE systems to generate a set of triple relations that are then used to build an event network. The event network is bootstrapped by labeling the temporal and causal disposition of events that are directly linked to each other. The event network can be systematically traversed to identify temporal and causal relations between indirectly connected events.

Acknowledgements

My deepest thanks go to my supervisor, Dr. Ebrahim Bagheri, for sharing his broad and deep knowledge in natural language processing, information retrieval, and machine learning. I would like to show my sincere gratitude to him with all my heart and soul. During my time at Ryerson University, he has supported me both in my research and my life; especially, he knows how to encourage me with his enthusiasm, inspiration and careful instruction. Once again, I do appreciate everything he did for my success.

I would like to show my gratitude to the professors on my examination committee (alphabetical): Prof. Soosan Beheshti, Prof. Klaus Berberich, Prof. Mehdi Kargar, and Prof. Sri Krishnan for their concerns, advice, suggestions, and valuable feedback on my work.

I can never thank my fellow labmates at Ryerson's LS^3 enough, especially N. Arabzadeh, Dr. M. Bashari, Dr. F. Ensan, Y. Feng, Dr. H. Fani, H. Hosseini, A. Keikha, A. Pourmasoomi, A. Pourali, J. Mahdavi Moghaddam, A. Mirlohi, R. Hamidi Rad, M. Tamannaee, Dr. M. Noorian, K. Poots, Dr. Z. Noorian, Dr. F. Zarrinkalam, and Dr. T. Nguyen for their kindness and friendship. Furthermore, I would like to thank them for sharing their ideas, knowledge and passion with me.

I will always respect my dear parents who brought me to this world and gave me a firm will to pursue my passion for research; and my mother-in-law for her love and encouragement. I dedicate my sincere and unwavering thanks to my beloved wife and son for their love and support in the most difficult times of my research.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

IE Information Extraction

ACE Automatic Content Extraction

CNN Convolutional Neural Network

CRF Conditional Random Fields

DP Dependency Parsing

EE Event Extraction

HMM Hidden Markov Model

IRM Infinite Relational Model

KB Knowledge Base

K-CNN Knowledge-oriented Convolutional Neural Network

LDA Latent Dirichlet Allocation

LSTM Long Short-term Memory Unit

MUC Message Understanding Conference

NER Named Entity Recognition

NLP Natural Language Processing

NN Neural Network

Open IE Open Information Extraction

PCA Principle Component Analysis

POS Part of Speech

RDF Resource Description Framework

RE Relation Extraction

RNN Recursive Neural Network

SDP Shortest Dependency Path

SP Syntactic Parsing

SVM Support Vector Machine

Chapter 1

Introduction

1.1. Overview

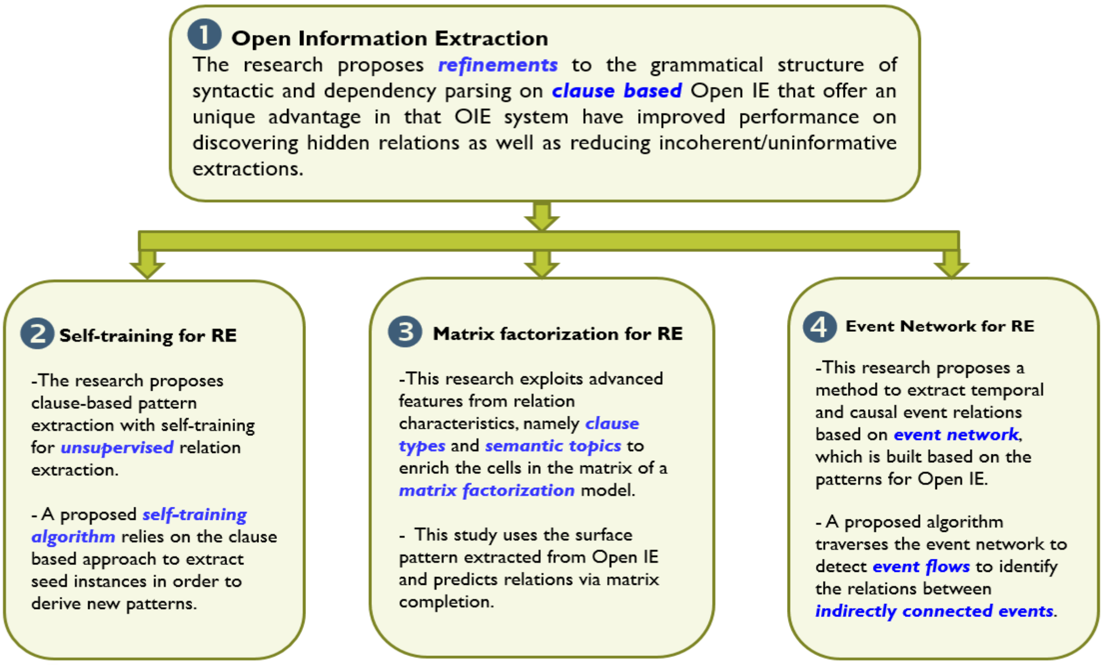
Information Extraction (IE) is an important task in natural language processing for extracting structured information from unstructured textual information. IE is growing as one of the active research areas in artificial intelligence for enabling computers to read and comprehend unstructured textual content (Etzioni et al., 2012). IE systems aim to distill semantic relations which present relevant segments of information on entities and relationships between them from large numbers of textual documents. The goal of IE is to extract and represent information in a tuple of two entities and a relationship between them automatically. For instance, given the sentence “Donald Trump is the President of the United States”, they venture to extract the relation tuple PresidentOf(Donald Trump, the United States). The identified relations can be used for enhancing machine reading by building knowledge bases in Resource Description Framework (RDF) or ontology forms. Most IE systems (Bunescu et al., 2005; Zhou et al., 2007; Zhou et al., 2010) focus on extracting tuples from domain-specific corpora and rely on some form of pattern-matching technique. Therefore, the performance of these systems is heavily dependent on considerable domain specific knowledge. Several methods employ advanced pattern matching techniques in order to extract relation tuples by learning patterns based on labeled training examples that serve as initial seeds.

Banko and Etzioni (2007) have introduced one of the pioneering Open Information Extraction (Open IE) systems called TextRunner (Banko et al., 2007) which is a tool for the extraction of relations and their arguments and is agnostic to the target domain. More recently, there has been a move towards next generation IE systems that can be highly scalable on large Web corpora. Such systems need to tackle an unbounded number of relations, and scale linearly. Open IE is currently being developed in its second generation in systems such as ReVerb (Fader et al., 2011), OLLIE (Mausam et al., 2012), and ClausIE (Corro et al., 2013), which extend from previous Open IE systems such as TextRunner (Banko and Etzioni, 2007), StatSnowBall (Zhu et al., 2009), and WOE (Wu et al., 2010).

The ultimate objective of both IE and Open IE systems is to enable the extraction of knowledge that can be represented in structured form and in human readable format. The extracted knowledge can be then used to answer questions (Reddy et al., 2014; Yao et al., 2014; Ryu et al., 2015;). For instance, TextRunner can support user input queries such as “(?, kill, bacteria)” or “(Barrack Obama, ?, U.S)” similar to Question Answering systems. By replacing the question mark in the triple, questions such as “what kills bacteria” and “what was the relationship types between Barack Obama and U.S” will be developed and can be answered. Furthermore, IE systems can be integrated and applied in many higher levels of NLP tasks such as text similarity or text summarization (Christensen et al., 2013; Christensen et al., 2014; Levy et al., 2014). Relation tuples from IE systems can be used to infer or measure the redundancy between sentences based on the facts extracted from the input corpora. Finally, IE systems can enable the automated learning and population of an upper level ontology due to their ability in the scalable extraction of information across domains (Soderland et al., 2013). For instance, Open IE systems can enable the learning of a new biomedical ontology for gene-disease relationships by automatically reading and processing published papers in the literature on this topic.

However, most IE systems suffer from problems such as extracting incoherent and uninformative relations. For example, TextRunner and WOE would extract a triple such as (Peter, thought, his career as a scientist) from the sentence “Peter thought that John began his career as a scientist”, which is clearly incoherent because “Peter” could not be taken as the first argument for relation “began” with the second argument “his career as a scientist”. The second problem, uninformative extractions, occurs when Open IE systems miss critical information of a relation. Uninformative extraction is a type of error relating to light verb construction (Allerton et al., 2002; Stevenson et al., 2004, Fader et al., 2011; Mausam et al., 2012) due to multi-word predicates being composed of a verb and an additional noun. For example, given the sentence "Al-Qaeda claimed responsibility for the 9/11 attacks", Open IE systems such as TextRunner return the uninformative relation (Al-Qaeda, claimed, responsibility) instead of (Al-Qaeda, claimed responsibility for, the 9/11 attack).

In this thesis, we will explore approaches for relation extraction by taking advantage of clause-based patterns from open information extraction. Our work will provide a grammatically refined structure when using English grammar clauses for open information extraction. This is aimed at making the patterns increasingly more effective for the purpose of deriving a set of coherent constituents directed toward generating relation clauses to serve for bootstrapping, matrix factorization and event networks for relation extraction.



The research proposes ***refinements*** to the grammatical structure of syntactic and dependency parsing on ***clause based*** Open IE that offer an unique advantage in that OIE system have improved performance on discovering hidden relations as well as reducing incoherent/uninformative extractions.

-The research proposes clause-based pattern extraction with self-training for ***unsupervised*** relation extraction.

- A proposed ***self-training algorithm*** relies on the clause based approach to extract seed instances in order to derive new patterns.

-This research exploits advanced features from relation characteristics, namely ***clause types*** and ***semantic topics*** to enrich the cells in the matrix of a ***matrix factorization*** model.

- This study uses the surface pattern extracted from Open IE and predicts relations via matrix completion.

-This research proposes a method to extract temporal and causal event relations based on ***event network***, which is built based on the patterns for Open IE.

- A proposed algorithm traverses the event network to detect ***event flows*** to identify the relations between ***indirectly connected events***.

Figure. 1.1. Four research directions.

1.2. Motivation

Open information extraction extracts relations and their arguments automatically from textual documents that require no supervision for performing highly scalable extractions and are often portable across domains. Taking advantage of clause-based patterns from Open IE systems for information extraction, we follow four research directions shown in Figure 1.1.

1.2.1. Grammatical Structure Reformation for Open Information Extraction

As mentioned earlier, many state-of-the-art Open IE systems are faced with the problem of extracting incoherent relations in circumstances when the system extracts relation phrases that present a meaningless interpretation of the content that are mediated by verbs or verbal phrases based on dependency parsing (Fader et al., 2011; Mausam et al., 2012; Corro et al., 2013). Despite key advantages to dependency parsing-based approaches, their failure to extract all potential relations beyond a pre-defined set of relation including syntactic entities such as nouns and adjectives along with a whole range of verbal structures can be problematic. Given segmentation "*...in the southeast Sumatra area of Indonesia...* ", the relation between “southeast Sumatra area” and “Indonesia” cannot be determined by any type of verbs or verbal phrases through either syntactic or dependency parsing. Our work will offer a unique advantage in that it is designed to address some of the more pressing limitations inherent in previous OIE systems through the reformation of the grammatical structure obtained from SP and DP.

1.2.2. Self-training for Information Extraction

Semi-supervised learning (Xu et al., 2007; Xu et al., 2010) is often employed with predefined rules to train relations based on a dependency tree. However, such an approach results in low performance when used on unobserved new domains due to the high likelihood of extracting incorrect rules from the dependency tree during the bootstrapping process. Researchers use dependency trees as the input for pattern extraction and work with trees or sub-trees that contain seed arguments. Despite, their eagerness to maintain high accuracy, it is difficult to claim with certainty that the identified patterns are indeed accurate. In lieu of this, there is a probability that faulty seeds could potentially be injected into the bootstrapping process. To address such limitations, we propose a new bootstrapping approach for relation extraction that automatically learns the required initial seed set without the need for manual input.

1.2.3. Matrix Factorization for Information Extraction

Distant supervision (Angeli et al., 2014; Mintz et al., 2009; Riedel et al., 2013; Min et al., 2013) aims to exploit information from knowledge bases such as Freebase in order to learn large-scale relations from text. Heuristic methods (Mintz et al., 2009; Surdeanu et al., 2012) have been employed to generate training relations by mapping phrases to their corresponding entities in KBs. Dependence on pre-existing datasets in distant supervision approaches can be avoided by using language itself as the source for the universal schema. To this end, Riedel et al. (2013) and Yao (2015) have already presented a model based on matrix factorization with universal schemas for predicting relations. These authors presented a series of models that learn lower dimensional manifolds for tuple of entities and relations with a set of weights in order to capture direct correlations between relations. While these approaches have shown reasonable performance, their limitation is in that they train cells only for tuple of entities, and therefore, are limited when an insufficient number of evidences are present for the entities present in the relations. In this thesis, we exploit advanced features from relation characteristics, namely *clause types* and *semantic topics* to enrich the cells in the matrix of a matrix factorization model for predicting new relation instances. Particularly, we exploit clause types and topic models to predict relations regardless of whether they were seen at training time with direct or indirect access.

1.2.4. Event Network for Information Extraction

Temporal and causal event relations specify how different events expressed within the context of a textual passage relate to each other in terms of time sequence. They are a natural way of identifying the relationship between major events such as news stories. There have already been impactful work (Laokulrat et al., 2013; Mirza et al., 2014b; Laokulrat et al., 2015; Mirza et al., 2016a) in the area of temporal event relation extraction; however, they are mostly supervised methods that rely on sentence-level textual, syntactic and grammatical structure patterns to identify temporal relations. In this thesis, we propose an unsupervised method that operates at the document level. More specifically, we benefit from existing Open IE systems to generate a set of triple relations that are then used to build an event network. The event network is bootstrapped by labeling the temporal and causal disposition of events that are directly linked to each other. We then systematically traverse the event network to identify the temporal and causal relations between indirectly connected events.

1.3. Contributions

The contributions of this thesis research are listed as follows:

- We have demonstrated that a clause-based approach with grammatical structure reformation can be a suitable method for open information extraction to address the following limitations: (1) Identifying relations that previous OIE systems have been oblivious to or overlooked altogether and (2) Reducing the number of erroneous relation extractions.

- We have presented an unsupervised method for relation extraction by combining clause based pattern extraction and bootstrapping. Particularly, we will present a self-training algorithm for the purpose of labeling relations during the bootstrapping process.
- We have proposed numerous matrix models with fully enriched features such as word context, selectional preference, clause types and statistical topic models and employ matrix factorization with direct/indirect references for predicting specific relations between entities.
- We have proposed an event network representation for extracting causal and temporal relations. The event network is bootstrapped by labeling the temporal disposition of events that are directly linked to each other. We then systematically traverse the event network to identify the temporal and causal relations between indirectly connected events.
- We have evaluated and compared our work against the state-of-the-art systems both on Open IE and IE tasks.

1.4. Structure of Thesis

The thesis is composed of seven chapters. This first chapter covers the research motivation and contributions of the work. The rest of the proposal is structured as follows:

Chapter 2 presents the background on information extraction. The core steps of approaches classified as supervised learning, semi-supervised learning, distant supervision, matrix factorization, among others, will be described.

Chapter 3 presents our approach on grammatical structure reformation for open information extraction. It includes the depiction of the general clause-based framework, discussion about syntactic and dependency parsing, and the presentation on the new methods for grammatical structure reformation.

In Chapter 4, we present a clause-based pattern extraction approach with self-training for unsupervised relation extraction. Initially, we propose an approach for extracting patterns from Open IE that might contain relations. Then we propose a self-training algorithm that extracts more relations based on the initial seed set based on coherence scores from the extracted clause-based patterns.

Chapter 5 presents a framework for predicting potential relation instances based on feature enrichments applied to matrix models that are used in a matrix completion process. We exploit universal schemas that are formed as a collection of patterns from Open IE systems and relation schemas from pre-existing datasets to build a matrix model in order to use matrix factorization and collaborative filtering to predict relations.

Chapter 6 presents an event network representation for temporal and causal relation extraction that operates at the document level. More specifically, we benefit from existing Open IE systems to generate a set of triple relations that are then used to build an event network. The event network is bootstrapped by labeling the temporal disposition of events that are directly linked to each other. We then systematically traverse the event network to identify the temporal and causal relations between indirectly connected events.

In Chapter 7, concluding remarks and possible fruitful directions for future research are presented.

Chapter 2

Literature review

Information extraction (IE) in general has become an active research topic in Natural Language Processing during the past decade. This chapter gives an overview of IE including IE history, Open IE, Relation Extraction and Event Extraction and covers some of the more important methods and techniques in each of these areas.

2.1. Brief history of Information Extraction

The idea to extract structured information from textual documents can be traced back to 1987 when Sager et al. (1987) proposed a system to extract information from medical documents. Information Extraction (IE) including Open IE aims at automatically identifying and classifying the instances of user-specified types of entities, relations and events in order to discovery various semantic relations in natural language text. IE outputs are presented in structured form, e.g., a database, ontology, which can be readily interpreted by other applications such as search engines, question answering systems or expert systems. Two common evaluation corpora have been an integral component of most areas of IE namely, (1) Message Understanding Conference (MUC) (Grishman & Sundheim 1996), which focuses on named entity recognition and relation extraction and; (2) Automatic Content Extraction (ACE) which offers annotated training data for entities, relations and events that fosters the development of supervised methods (Doddington et al. 2004). In the last two decades, IE has grown into a major subfield of natural language processing. In this section, we present a brief history of three main tasks of IE including Named Entity Recognition, Relation Extraction, and Event Extraction based on a chronological order.

2.1.1. Brief History of Named Entity Recognition

The goal of NER is to identify entity names such as: person, organization, place names, temporal expressions, and certain types of numerical expressions. Most of the major works in NER have been presented in the last two decades shown in Figure 2.1.

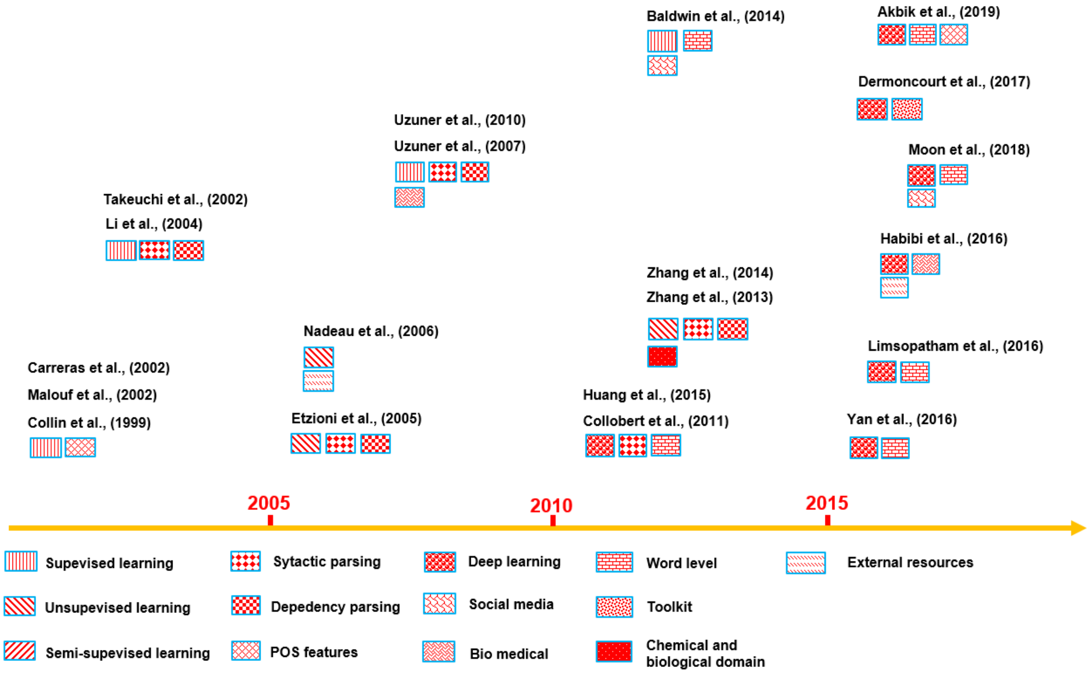


Figure. 2.1. Brief history of NER.

Before 2005. Most of research before 2005 have employed supervised learning methods, e.g., Hidden Markov Models (HMM), Conditional Random Fields (CRF), and Support Vector Machines (SVM), to solve the sequence labeling problem for NER. Collins et al. (1999) used only labeled seeds, and entity features for classifying and extracting named entities. Malouf (2002) and Zhou et al. (2002) exploited the Hidden Markov Model approach to train a model to identify named entities. Carreras et al., 2002) used binary AdaBoost classifiers in combination with decision trees for NER. Besides that, several approaches for NER initially employed grammatical features such as syntactic and dependency and then train them with SVM for entity recognition (Takeuchi and Collier, 2002; Li et al., 2005).

From 2005 to 2010. The problem with supervised learning approaches with NER happens on large text with a high number of categories. Unsupervised learning methods have been proposed to address the limitations of supervised learning, e.g., small hand-labeled annotations and closed domain datasets. Etzioni et al. (2005) proposed an unsupervised learning method by applying eight generic pattern extractors from web text for NER. Nadeau et al. (2006) presented an unsupervised learning approach to identify and classify entities in a huge corpus based on a constructed gazetteer. In this period, the new BioMed NER was introduced. Uzuner et al. (2007) presented a method biomedical NER identification on clinical notes. Later, Uzuner et al. (2011) presented the 2010 I2B2 NER tasks which focused on clinical data such as test and treatment entity types.

From 2011 to 2015. The first RNN deep learning model for NER was presented by Collobert et al. (2011). Following that, a myriad of deep learning techniques came to the fore. Huang et al. (2015) presented a long short-term memory model for NER. Yao et al. (2015) presented word representation based on deep neural networks to train a model for BioMed NER. Besides that, research on NER expanded to several new domain, e.g., chemical and biological NER. Eltyeb and Salim (2014) developed a dataset for chemical NER. Zhang and Elhadad (2013) exploited syntactic parsing and for an unsupervised NER system for the biology domain. In this period, a new trend on NER for social media content was also observed. Baldwin et al. (2015) introduced NER tasks on social media data (person, company, facility, band, sports team, movie, TV show, among others). The authors leveraged sources of information such as POS tags, or token substitutions to address NER in Tweets.

From 2016 to 2019. Deep learning techniques have been rapidly growing in NER. Yan et al. (2016) have presented word level feed forward NN, bi-directional LSTM (bi-LSTM) and window bi-LSTM models for NER. Limsopatham and Collier (2016) presented a method based on convolutional neural networks for Twitter NER. Habibi et al. (2017) presented a neural network system to pre-train word embeddings in order to improve biomedical named entity recognition. Moon et al. (2018) presented a deep learning model which considers short text snippets accompanying images in tweets or snapchats for NER. Moreover, A deep learning based NER toolkit, named NeuroNER (Dernoncourt et al., 2017), has been presented recently. Currently, the work by Akbik et al. (2019) show the best performance over NER by combining a dual token/character model with contextualized word embeddings.

2.1.2. Brief History of Relation Extraction

The goal of Relation Extraction (RE) is to identify relationship that have been expressed between entities within a textual document. We present research on relation extraction based on four main time periods shown in Figure 2.2.

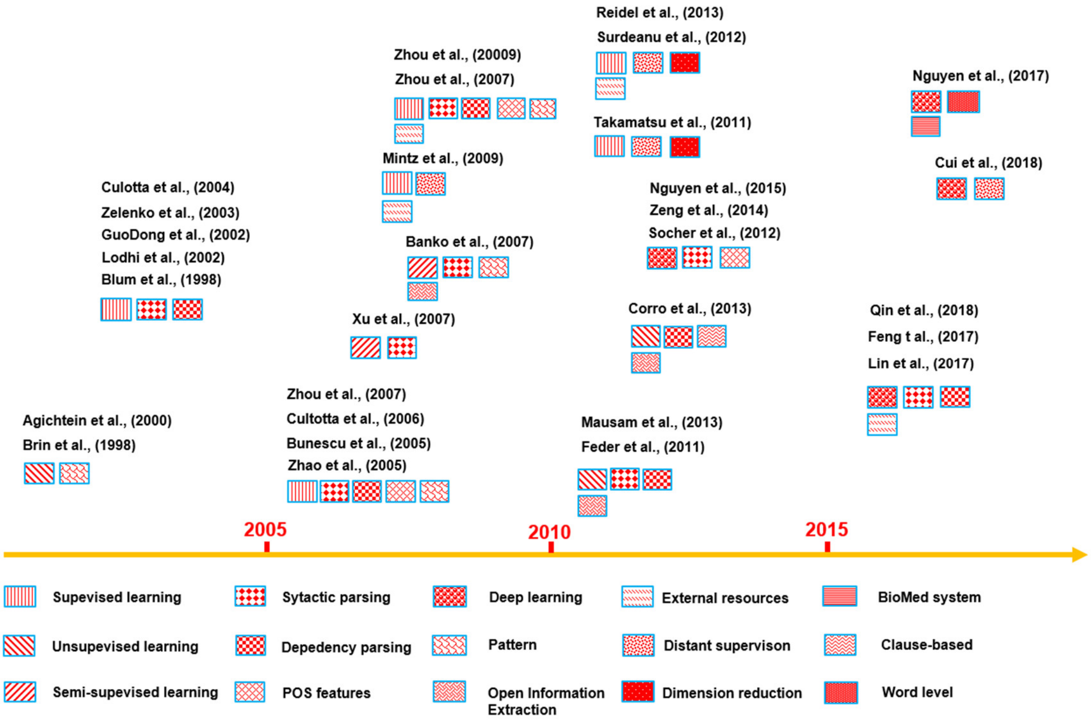


Figure. 2.2. Brief history of Relation Extraction.

Before 2005. DIPRE (Brin, 1998) and Snowball (Agichtein et al., 2020), two pioneering RE systems, require a small set of instances or a few hand-crafted patterns for relation extraction. Most of the early research (Blum et al., 1998; Lodhi et al., 2002; Zelenko et al., 2003; and Culotta et al., 2004) exploited grammatical features, e.g., syntactic and dependency features with supervised learning for relation extraction. GuoDong et al. (2002) used SVMs to train syntactic and dependency features for relation extraction. Lodhi et al., (2002), Zelenko et al., (2003), and Culotta et al. (2004) measured tree kernels based on shallow-parse trees for relation classification.

From 2005 to 2010. More recent research (Zhao & Grishman, 2005; Bunescu et al., 2005; Culotta et al., 2006) focus on training deeply grammatical features with machine learning for relation extraction. Zhao & Grishman, (2005) employ SVMs trained on such features using polynomial and linear kernels for classifying relation types. Bunescu et al. (2005) define tree sub-kernels in the form of before, middle and after portions of entities for relation extraction. Culotta et al. (2006) use Wikipedia as a rich resource which contains hyperlinked entities in their pages for relation extraction. Zhou et al. (2007) employ lexical, syntactic and semantic knowledge in their feature-based relation extraction approach. Moreover, several research work (Zhou et al., 2007; Zhou et al, 2010) have employed external resources, e.g., WordNet and Wikipedia to perform relation classification. Semi-supervised learning based techniques (Xu et al., 2007), open information extraction systems (Banko et al., 2007), and distant supervision techniques (Mintz et al., 2009) started to appear in this period.

From 2011 to 2015. Open information extraction systems were further developed into their so-called second generation (Feder et al., 2011, Mausam et al., 2012; Corro et al., 2013). Fader et al. (2011) presented ReVerb, which effectively applies syntactic and lexical constraints in verbal relation sequences for Open IE. Corro et al. (2013) proposed the clause-based approach for Open IE. In this period, several researchers began employing deep learning models for relation extraction. Socher et al. (2012) presented the RNN model to learn compositional vector representations for phrases and sentences of arbitrary syntactic type and length in order to classify semantic relations between nouns using syntactic paths. Zeng et al. (2014) presented a CNN model for relation classification where sentence-level features are learned through a CNN. Nguyen et al. (2015) applied CNN models in ACE relations. Regarding distant supervision, external resources, e.g., Freebase, DBpedia, were employed with techniques of dimensionality reduction to classify large relation types. Takamatsu et al. (2011) used probabilistic matrix factorization with Singular Value Decomposition to reduce dimensions to discover relations. Surdeanu et al., (2012) used the Freebase knowledge graph by considering each pair of entities that are related to each other within the set of Freebase relations. Reidel et al. (2013) used matrix factorization by combining surface patterns extracted from Open IE and knowledge bases such as Freebase to train latent relations.

From 2016 to 2019. Deep learning has been gaining even more momentum. Lin et al. (2017) employed deep learning techniques for RE in multiple languages. Feng et al. (2018) presented a model that fed ranked sentences to the relation classifier using reinforcement learning. Qin et al. (2018) combined deep learning models and distant supervision for RE. Cui et al. (2018) proposed a neural approach with an encoder-decoder framework which is the first research employing deep learning techniques specifically in Open IE. Besides that, a deep learning BioMed system, Biomedical relation extraction, has been presented by Nguyen et al. (2019) using different pooling mechanisms.

2.1.3. Brief History of Event Extraction

Event extraction aims at identifying knowledge about incidents of events found in text. We review a brief history of research related to event extraction shown in Figure 2.3.

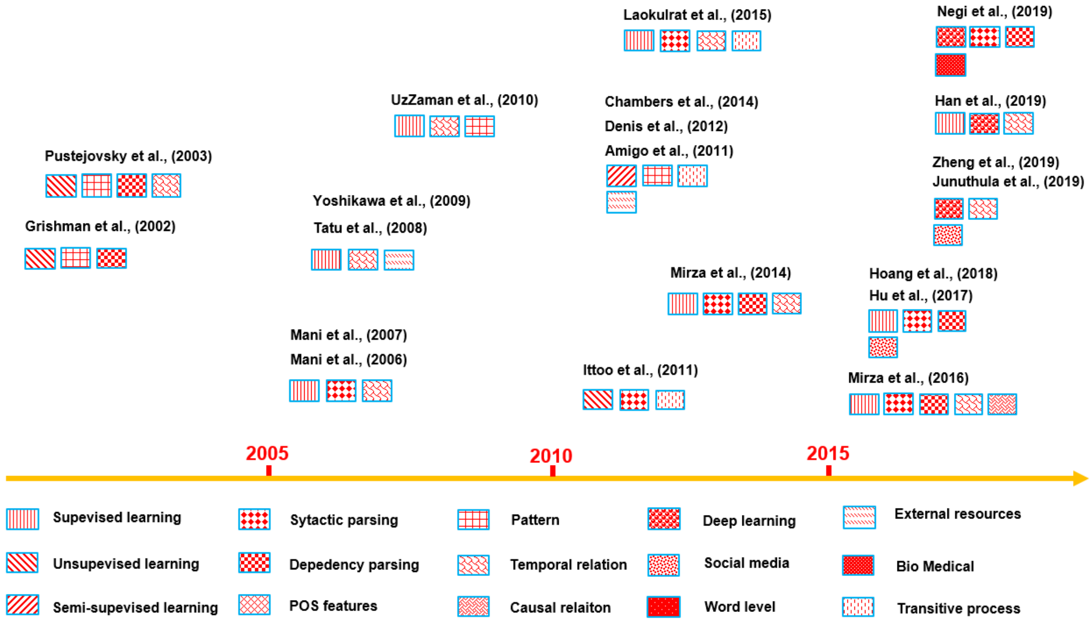


Figure. 2.3. Brief history of Event Extraction.

Before 2005. There was only a few work on event extraction prior to 2005. Grishman et al. (2002) presented a pattern-based method to extract and update an event database of infectious disease outbreaks. Pustejovsky et al. (2003) presented the TimeML corpus with temporal link annotations between events and temporal expressions.

From 2005 to 2010. In this period, supervised learning techniques were the predominant approach for event relation extraction. Mani et al. (2006) used supervised leaning methods to classify temporal relations based on local information. Mani et al. (2007) focused on formulating temporal feature ordering around event pairs for classification. Tatu et al. (2008) exploited external temporal resources to improve supervised learning for event relation extraction. Moreover, some research (Yoshikawa et al., 2009; UzZaman et al., 2010) exploited external resources to enhance event features for classification. Yoshikawa et al. (2009) used global information to improve event relation classification. UzZaman et al. (2010) took advantage of external information to enhance temporal logic transitivity constraints for pairwise event classification.

From 2011 to 2015. The main focus of research work was on exploiting deep linguistic features, e.g., dependency paths, with supervised learning or handcrafted rules for event relation extraction. Ittoo at al. (2011) had proposed a rule-based method with handcrafted linguistic patterns consisting of syntactic and semantic analysis in order to identify event relations from textual documents based on pattern matching. Chambers et al. (2014) combined rule-based and a data-driven method for temporal ordering in order to identify event relation. Laokulrat et al. (2013) trained features based on syntactic parsing including phrase structures for relation classification. TRelPro (Mirza et al., 2014b) employed an SVM classifier based on event linguistic features. Amigo et al, (2011) and Denis and Muller, (2011) presented inference models for the identification of temporal relations. Laokulrat et al. (2015) used time graphs and stacked learning to perform event relation classification.

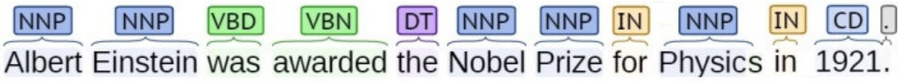
From 2016 to 2019. Mirza et al. (2016a) extended TRelPro system for causal relation extraction. Han et al. (2019) proposed a deep structured learning framework to learn scoring functions for pair-wise relations and employed a structured support vector machine for temporal relation extraction. Moreover, problems of event relations in social media or medical domains have been introduced in this period. Hu et al. (2017) focused on causal relation extraction from blogs and films. Hoang et al. (2018) worked on tweets to predict location and time events on social data. Junuthula et al. (2019) end Zheng et al. (2019) analyzed temporal relations between events in on-line social networks using a block point process model (BPPM). Negi et al. (2019) employed syntactic and dependency features in order to classify causal relationships between drugs.

2.2. Information Extraction

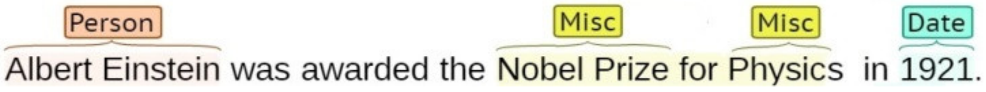
2.2.1. Open Information Extraction

Open Information Extraction systems aim to obtain relation tuples with highly scalable extraction by identifying a variety of relation phrases and their arguments in arbitrary sentences. The first generation of Open IE systems learn linear chain models based on unlexicalized features such as Part-of-Speech (POS) or shallow tags to label the intermediate words between pair of potential arguments for identifying extractable relations. Open IE currently is developed in the second generation that is able to extract instances of the most frequently observed relation types such as Verb, Noun and Prep, Verb and Prep, and Infinitive with deep linguistic analysis. They expose simple yet principled ways in which verbs express relationships in linguistics such as verb phrase-based extraction or clause-based extraction. They obtain a significantly higher performance over previous systems in the first generation.

Part-of-Speech


NNP NNP VBD VBN DT NNP NNP IN NNP IN CD .
Albert Einstein was awarded the Nobel Prize for Physics in 1921.

Named Entity Recognition


Person
Misc
Misc
Date
Albert Einstein was awarded the Nobel Prize for Physics in 1921.

Named Entity Recognition

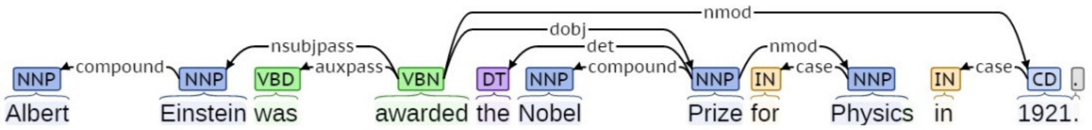

nmod
dobj
nsubjpass
det
nmod
compound
auxpass
compound
case
case
NNP NNP VBD VBN DT NNP NNP IN NNP IN CD .
Albert Einstein was awarded the Nobel Prize for Physics in 1921.

Figure. 2.4. POS, NER and DP analysis in the sentence “Albert Einstein was awarded the Nobel Prize for Physics in 1921”.

2.2.1.1. The first generation

In the first generation, Open IE systems aimed at constructing a general model that could express a relation based on unlexicalized features such as Part-of-Speech (POS) or shallow tags e.g., a description of a verb in its surrounding context or the presence of capitalization and punctuation. While traditional IE requires relations to be specified in their input, Open IE systems use their relation-independent model as self-training to learn relations and entities in the corpora. TextRunner is one of the first Open IE systems. It applied a Naive Bayes model with POS and Chunking features that trained tuples using examples heuristically generated from the Penn Treebank. Subsequent work showed that a linear-chain Conditional Random Field (CRF) (Banko et al., 2007; Etzioni et al., 2008) or Markov Logic Network (Zhu et al., 2009) can be used for identifying extractable relations. Several Open IE systems have been proposed in the first generation, including TextRunner (Banko et al., 2007), WOE (Xu et al., 2010), and StatSnowBall (Zhu et al., 2009) that typically consist of the following three stages: 1)

Intermediate levels of analysis and 2) Learning models and 3) Presentation, which we elaborate in the following:

Intermediate levels of analysis: In this stage, NLP techniques such as Named Entity Recognition (NER), POS and Phrase-chunking are used. The input sequence of words is taken as input and each word in the sequence is labeled with its part of speech, e.g., noun, verb, adjective by a POS tagger. A set of non-overlapping phrases in the sentence is divided based on POS tags by a phrase chunked. Named entities in the sentence are located and categorized by NER. Some systems such as TextRunner and WOE^{pos} defined a method to identify useful proposition components from parse trees as shown in Figure 2.4. The structure and annotations from a parse tree will be essential for determining the relationship between entities for learning models of the next stage.

Learning models: An Open IE would learn a general model that depicts how a relation could be expressed in a particular language. A linear-chain model such as CRF can then be applied to a sequence which is labeled with POS tags, word segments, semantic roles, named entities, and traditional forms of relation extraction from the first stage. The system will train a learning model given a set of input observations to maximize the conditional probability of a finite set of labels. TextRunner and WOE^{pos} use CRFs to learn whether sequences of tokens are part of a relation. When identifying entities, the system determines a maximum number of words and their surrounding pair of entities which could be considered as possible evidence of a relation. Figure 2.5 shows entity pairs "*Albert Einstein*" and "*the Nobel Prize*" with the relationship "was awarded" serving to anchor the entities.

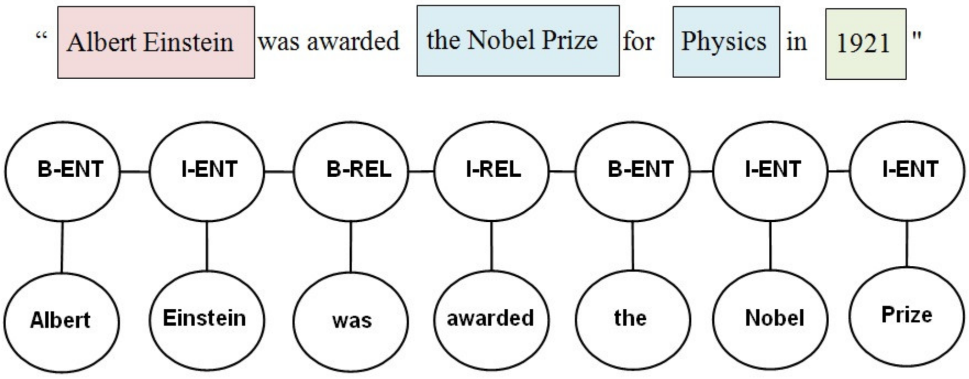


Figure. 2.5. A CRF is used to identify the relationship "was awarded" between "Albert Einstein" and "the Nobel Prize".

Presentation: In this stage, Open IE systems provide a presentation of the extracted relation triples. The sentences of the input will be presented in the form of instances of a set of relations after being labeled by the learning models. TextRunner and WOE^{pos} take sentences in a corpus and quickly extract textual triples that are present in each sentence. The form of relation triples contains three textual components where the first and third denote pairs of entity arguments and the second denotes the relationship between them as (Arg1, Rel, Arg2). Figure 2.6 shows the differences of presentations between traditional IE and Open IE. Additionally, with large scale and heterogeneous corpora such as the Web, Open IE systems also need to address the disambiguation of entities e.g., the same entity may be referred to by a variety of names "*Obama*" or "*Barack Obama*" or "*B. H. Obama*" or the same string "*Micheal*" may refer to different entities. Open IE systems try to compute the probability that two strings denote synonymous pairs of entities based on scalable and unsupervised analysis of tuples. TextRunner applies the Resolver system (Yates et al., 2007) while WOE^{pos} uses the infoboxes from Wikipedia for classifying entities in the relation triples.

Sentence: "Apple Inc. is headquartered in California"
Traditional IE: Headquarters(Apple Inc., California)
Open IE: (Apple Inc., is headquartered in, California)

Figure. 2.6. Traditional IE and Open IE extractions.

2.2.1.2. The second generation

In the second generation, Open IE systems focus on addressing the problem of incoherent and uninformative relations. In some cases, TextRunner and WOE^{pos} do not extract the full relation between two noun phrases, and only extract a portion of the relation which is ambiguous. For instance, where it should extract the relation "*is author of*", it only extracts "*is*" as the relation in the sentence "William Shakespeare is author of Romeo and Juliet". Similar to first generation systems, Open IE systems in the second generation have also apply NLP techniques in the intermediate level analysis of the input and the output is processed in a similar vein to the first generation. They take a sentence as input and perform POS tagging, syntactic chunking and dependency parsing and then return a set of relation triples. However, in the intermediate level analysis process, Open IE systems in the second generation focus deeply on a thorough linguistic analysis of sentences. They expose simple yet principled ways in which verbs express relationships in linguistics. Based on these linguistic relations, they obtain a significantly higher performance over previous systems in the first generation. Several Open IE systems have been proposed after TextRunner and WOE^{pos}, including ReVerb, OLLIE, Christensen et al., (2011), ClausIE (Corro et al., 2013) with two extraction paradigms, namely verb-based relation extraction and clause-based relation extraction.

Verb phrase-based relation extraction

ReVerb is one of the first systems that extracts verb phrase-based relations. This system builds a set of syntactic and lexical constraints to identify relations based on verb phrases then finds a pair of arguments for each identified relation phrase. ReVerb extracts relations by giving first priority to verbs. Then the system extracts all arguments around verb phrases that help the system to avoid common errors such as incoherent or uninformative extractions made by previous systems in the first generation. ReVerb considers three grammatical structures mediated by verbs for identifying extractable relations. In each sentence, if the phrase matches one of the three grammatical structures, it will be considered as a relation. Give a sentence “Albert Einstein was awarded the Nobel Prize.” for each verb V (awarded) in sentence S, it will find the longest sequence of words (V | VP | VW*P) such that (1) it starts with V, (2) it satisfies the syntactic constraint, and (3) it satisfies the lexical constraint. As a result, (V | VP | VW*P) identifies “was awarded” as a relation. For each identified relation phrase R, it will find the nearest noun phrase X to the left of R, which is “Albert Einstein” in this case. Then it will find the nearest noun phrase Y to the right of R, which is “the Nobel Prize” in S.

1. “*Microsoft co-founder Bill Gates spoke at ...*”
   OLLIE: (“Bill Gates",“be co-founder of”, “Microsoft”)
2. “*Early astronomers believed that the earth is the center of the universe.*”
   ReVerb: (“the earth”,“be the center of”, “the universe”)
   OLLIE: (“the earth”,“be the center of”, “the universe”)
   AttributeTo believe; Early astronomers
3. “*If he wins five key states, Romney will be elected President.*”
   ReVerb:(“Romney”, “will be elected”, “President”)
   OLLIE: (“Romney”, “will be elected”, “President”)
   ClausalModifier if; he wins five key states

Figure 2.7. ReVerb extraction vs. OLLIE extraction (Mausam et al., 2012).

OLLIE has addressed the problem in ReVerb by adding two new elements namely “AttributedTo" and “ClauseModifier” to relation tuples when extracting all relations mediated by noun, adjective, and others. “AttributeTo” is used for determining additional information and “ClauseModifier” is used for adding conditional information as seen in sentences 2 and 3 in Figure 2.7. OLLIE produces high yield by extracting relations not only mediated by verbs but also mediated by nouns, and adjectives. OLLIE follows ReVerb to identify potential relations based on verb-mediated relations. This system applies bootstrapping to learn other relation patterns using its similarity relations found by ReVerb. In each pattern, the system uses dependency paths to connect a relation and its corresponding arguments for extracting relations mediated by nouns, adjectives and others. After identifying the general patterns, the system applies them to the corpus to obtain new tuples. Therefore, OLLIE extracts a higher number of relations from the same sentence compared to ReVerb.

Clause-based relation extraction

A more recent Open IE system named ClausIE presented by Corro & Gemulla (2013) uses clause structures to extract relations and their arguments from natural language text. Different from verb-phrase based relation extraction, this work applies clause types in sentences to identify useful pieces of information. ClausIE uses dependency parsing and a set of rules for domain-independent lexica to detect clauses without any requirement for training data. ClausIE exploits grammar clause structure of the English language for detecting clauses and all of its constituents in a sentence. As a result, ClausIE obtains high-precision extraction of relations and also it can be flexibly customized to adapt to the underlying application domain. Particularly, in the ClausIE system, a clause can consist of different components such as subject (S), verb (V), indirect object (O), direct object (O), complement (C), and/or one or more adverbials (A). This system obtains and exploits clauses for relation extraction in the following three steps: 1) it seeks to identify the clauses in the input sentence by obtaining the head words of all the constituents of every clause based on dependency parsing; 2) When a clause is obtained, it needs to be associated with one of the main clause types; this step uses a decision tree to identify the different clause types. In this process, the system marks all optional adverbials after the clause types have been identified; 3) the system extracts relations from a clause based on the patterns of the clause type. Assuming that a pattern consists of a subject, a relation and one or more arguments, it is reasonable to presume that the most reasonable choice is to generate n-ary propositions that consist of all the constituents of the clause along with some arguments. The system identifies the subject of each clause and then uses it to construct the proposition. To accomplish this, they map the subject of the clause to the subject of a proposition relation.

As previously indicated, the use of syntactic and dependency parsing has the potency to reduce precision at higher points due to incoherent information extractions after parsing. In our work, we address such limitations by refining the grammatical structure of SP and DP. To achieve this objective, we use English grammar clauses (Corro et al., 2013; Quirk et al., 1985) to identify the set of "clauses" as relations within each sentence, for each clause, we identify the corresponding clause type according to the grammatical function of its coherent constituent. By taking these factors into account and based on the obtained information, we succeeded in extracting relations with a higher precision than existing state-of-the-art approaches.

2.2.2. Relation Extraction

Relation extraction discoveries the relevant segment of information in relation triples that will be useful for structuring data. Relation extraction has been applied in many natural language processing task such as question answering, biography extraction, and textual entailment. Several learning approaches have mainly applied, named supervised methods (Zhou et al., 2010; Choi et al., 2013; Batista et al., 2015; Singhal et al., 2016), semi-supervised methods (Gupta et al., 2014; Xu et al., 2014, Zhang et al., 2015a) and unsupervised methods (Turney, 2008; Rosenfeld et al., 2007; Akbik et al., 2012), to determine semantic relations. Supervised methods exploit language analysis to generate features based on syntactic, dependency, or shallow semantic structures of the text. Based on these features, the models are then trained to identify pairs of entities that are related through relations and classify them based on pre-defined relation types. Semi-supervised methods perform the concept of information redundancy and hypothesize that similar relations tend to appear in similar contexts. Zhang et al.(2015a) and Batista et al. (2015) present a semi-supervised method for relation extraction using word embeddings to find similar relationships using NER on Freebase. Unsupervised methods (Turney, 2008; Akbik et al., 2012) are usually based on rules or some clustering techniques over a large unlabeled corpus for relation discovery and extractions. They use dependency trees as the input for pattern extraction and work with trees or sub-trees that contain seed arguments by clustering entity pairs.

2.2.3. Event Extraction

Grishman et al. (2002), Tanev et al., (2008) and Fukui et al., (2019) present pattern-based approaches for event extraction that identify events in free text such as news, emails and health reports and derive detailed information about who did what to whom, where and when. Grishman et al. (2002) also built a complete system that automatically extracts and updates an event database of infectious disease outbreaks. The system has an extraction engine that is based on text zones including headline, date, text body to extract events using predefined event patterns. Tanev et al. (2008) identified syntactic patterns to extract events on violent and natural disaster news. Their method used several components in a pipeline to learn patterns through a semi-automatic pattern acquisition technique consisting of a small corpus that was manually annotated with event information. Regarding supervised learning, several approaches (Hardy et al., 2006; Ahn, 2006; Bethard and Martin, 2006; Ji and Grishman, 2008; Abebe et al., 2019; Fan et al., 2019; Zhao et al., 2018; Ye et al., 2019; Zhang et al., 2019a) train one or more classifiers that can extract events from textual documents. Ahn (2006) proposed a series of classification sub-tasks such as event anchor identifier, argument identifier, attribute assigner and event core reference module for event extraction. Bethard and Martin, (2006) designed an event extraction method using sequence tagging with the BIO (Begin-Inside-Outside) schema. The schema is used to trained features based on Support Vector Machines (SVM) over the TimeBank corpus (Pustejovsky et al., 2003). Furthermore, Ji et al. (2008) and Wang et al. (2017) train event relation classifiers based on the words and phrases of relation arguments that evoke events in the text. Li et al. (2011) used topic modeling to improve the process of event extraction. Their method operates by recognizing topic clusters for event extraction. Ye et al. (2019) proposed a general ranking based multi-label learning framework combined with convolutional neural networks to seek latent connection between relation types for event extraction. Zhang et al. (2019a) presented a hybrid approach by combining semantic role labelling frames and lexicon of event nouns and Levin's verb classes for event extraction over 20 event types related to socio-economic phenomena.

2.2.4. Temporal Event Extraction

Temporal event relations specify how different events expressed within the context of a textual passage relate to each other in terms of time sequence. Allen (1983) first introduced a transitivity table of temporal intervals for reasoning about events, e.g., the table determines if event e_1 happens before event e_2 and e_2 happens before e_3 that e_1 happens before e_3. Supervised methods (Chambers et al., 2008; Mani et al., 2007; Katsumasa et al., 2009) have focused on formulating temporal features ordering around event pairs to formulate a classification task. Mani et al., (2007) as well as Tatu and Srikanth (2008) have curated training data by exploiting external temporal resources to improve supervised learning for temporal relation extraction. Chambers and Jurafsky (2008), UzZaman and Allen (2010) and Yoshikawa et al. (2009) take advantage of external information to enhance temporal logic transitivity constraints for pariwise classification. These methods face limitations related to the sparseness of event relations in a corpus (Chambers and Jurafsky, 2008).

Furthermore, Amigo et al., (2011) and Denis and Muller, (2011) presented an inference method over time graph to address limitations of constraint space to improve the identification of temporal relations. The time graph is built for representing temporal entities with edges as relations to predict the relationship between entity nodes. Moreover, UTTime (Laokulrat et al., 2013) and NavyTime (Chambers, 2013) exploit data-driven analysis using syntactic information and lexical semantic information for classifying temporal relation types. These systems only exploited features at the sentence-level such as predicate-argument structure of relations in a sentence. Mirza et al. (2014b) and Mirza et al. (2016a) have proposed hybrid classification models (Chambers et al., 2014; D'Souza and Ng, 2013; Navarro-Colorado et al., 2016; Ozdikis et al., 2016) for temporal event classification. They integrate rule-based and data-driven classifiers in a sieve-based architecture for temporal ordering. The classifiers are ordered by their individual precision. After each classifier proposes a label, the architecture infers transitive links from the new labels, adds them to the temporal label graph and informs the next classifier about this decision. Hoang et al. (2018) have presented a method to predict a location and time in a tweet by enhancing tweet representation with prepositions before a proper noun. Regarding deep learning methods, Han et al. (2019) have proposed a deep structured learning framework to learn scoring functions for pair-wise relations and employed a structured support vector machine for temporal relation extraction.

A main challenge with earlier work is that temporal and causal relations might be present across sentences. In this thesis, we address such challenge by proposing an event network representation for temporal and causal relation extraction that operates at the document level. More specifically, we construct event network based on triple relations generated from Open IE systems. The event network is made possible due to the linking of different events in the proposed event network whose linking transcends individual sentences and forms a representation of events in the document.

2.2.5. Causal Event Extraction

Recognizing causal relations between any two events is as challenging as detecting their temporal relation. In this context, Bethard et al. (2007) have collected conjoined event pairs from the Wall Street Journal corpus which described cause-effect relations for causal classification. Rink et al. (2010) improved Bethard et al.'s work by identifying relation types as a feature and proposed textual graph classification for causal event relation classification. Do et al. (2011) and Peng et al., (2016) have presented a method to measure causality on event triggers, e.g., relations on verb-verb, verb-noun and noun-noun between two events by using pointwise mutual information. They also used discourse information from trained corpus, e.g., connective types on Penn Discourse TreeBank (Do et al., 2011), as discourse connectives towards identifying causality between events. Ittoo and Bouma (2011) proposed causal patterns based on syntactic structure analysis in order to extract explicit and implicit causal relations. On other hand, Riaz et al. (2010) and Riaz et al. (2013) extracted causal relations based on analyzing verbs between events. Recently, Mirza et al. (2014b) and Mirza et al. (2016a) presented a data-driven approach with rules to extract (explicit) causal relations between events from a text. The rule-based system relies on an algorithm that, given a term *w* belonging to affect, link, causative verbs or causal signals, looks for specific dependency where term *w* is connected to the two observed events. If such dependencies are found, a causal link (CLINK) is automatically set between the two events. Negi et al. (2019) have trained syntactic and dependency features in order to classify the causal relationship between drugs and medical conditions such as suspect drug or non-suspect drug. Li et al. (2019) have used a Knowledge-oriented Convolutional Neural Network (K-CNN) model that incorporates human knowledge to capture the causal relationship, and a causal-oriented direction in order to learn cause-effect features for causal relation extraction. Fei et al. (2019) have proposed a method to use a recursive neural network that automatically learns syntactic features from dependency trees in order to represent global dependencies for biomedical event detection.

2.3. Machine learning based models

The task of information extraction was first introduced in the Message Understanding Conference (MUC). Since then, a number of techniques have been proposed for this task such as supervised learning (Bunescu et al., 2005; Kambhatla et al., 2004; Ravichandran et al., 2002; Zhou et al., 2010; Abacha et al., 2016; Singhal et al., 2016), distant supervision (Angile et al., 2014; Mintz et al., 2009, Riedel et al., 2013; Surdeanu et al., 2012), deep learning (Socher et al., 2012; Xu et al., 2015; Santos et al., 2015; Zeng et al., 2014; Zeng et al., 2015; Zheng et al., 2017), unsupervised learning (Etzioni et al., 2005; Turney, 2008; Rosenfeld et al., 2007; Akbik et al, 2012; Yao et al., 2012; Vlachidis et al., 2016; Oramasa et al., 2016), bootstrapping methods (Agichtein et al., 2000; Pantel et al., 2006; Xu et al., 2007; Batista et al., 2015) and matrix factorization (Kemp et al., 2006; Kolda et al., 2009; Kang et al., 2012; Bordes et al., 2013; Weston et al., 2013; Riedel et al., 2013; Yao, 2015). In this section, we present several studies that are relevant to IE.

2.3.1. Supervised learning

In supervised learning approaches (Bunescu et al., 2005; Kambhatla et al., 2004; Ravichandran et al., 2002; Zhou et al., 2007; Zhou et al., 2010; Choi et al., 2013; Song et al., 2015; Hashimoto et al., 2015; Abacha et al., 2016; Singhal et al., 2016), there is a heavy reliance on hand-crafted datasets for training the extractor with manually pre-labeled training data. The advantage of these approaches is the use of linguistic patterns for learning information from different surface expressions. These approaches rely on pre-specification of desired relations or patterns by performing hand coding. The common strategy of these approaches is to generate linguistic features based on the analysis of the syntactic features, dependency features, or shallow semantic structure of text. These systems are trained to identify pairs of entities, and to classify them based on the pre-defined relations. Kambhatla et al. (2004) used textual features such as POS, parsing, and NER to define features which include entities, types of entities (person, location), number of entities, number of words separating the two entities, and paths between the entities in a parse tree. Zhou et al. (2007) employed lexical, syntactic and semantic knowledge in feature-based relation extraction using support vector machines. The work by Zhou et al. (2010) illustrated how features can be constructed based on syntactic and semantic information from WordNet. Suchanek et al. (2007) built an ontology by extracting relations from Wikipedia categories using WordNet and heuristic rules. Choi et al. (2013) proposed a dependency trigram kernel based on Support Vector Machines (SVM) to classify the relationship between two persons' names in order to extract social relations. Reidel et al. (2013) used matrix factorization based on combining surface patterns extracted from OIE and knowledge bases such as Freebase to train latent relations. Abacha et al. (2016) trained an SVM classifier on the i2b2 2010 challenge's corpus. They used a set of lexical, morpho-syntactic and semantic features for each pair of medical entities (E1, E2) in order to be able to classify relations. Singhal et al. (2016) used supervised classifiers such as C4.5, Multilayer Perceptron, and Bayesian logistic regression on various types of features to identify relations of disease-mutation in biomedical text. While such approaches offer high precision and recall, most of them are laborious and expensive in training and face problems when handling large-scale text documents.

2.3.2. Unsupervised learning

Unsupervised approaches are usually based on rules or some clustering techniques over a large unlabeled corpus for relations. Several approaches have been built based on latent relation hypothesis (Turney, 2008; Rosenfeld et al., 2007; Akbik et al., 2012), latent topic assumption (Yao et al., 2012; Yao et al., 2011), low rank assumption (Takamatsu et al., 2011; Kok et al., 2008) and rule-based methods (Ryu et al., 2014; Vlachidis et al., 2016; Oramasa et al., 2016). Turney (2008), Akbik et al. (2012) and Yao et al., (2012) exploit features from the dependency tree for discovering relations by clustering entity pairs. The cluster vector space model is often applied by using the k-mean algorithm and cosine similarity is used to measure distance. In rule-based approaches, Ryu et al. (2014) have defined a set of relationships on named entities such as Person, Location, and Data to support question answering in the Korean language. Similarly, Vlachidis et al. (2016) have defined a set of rules based on syntactic analysis for extracting relation patterns within the archaeology domain. Further, Oramasa et al. (2016) have defined rules based on syntactic and semantic information to extract potential relations between entities, which have been discovered by traversing the dependency tree in the music domain.

2.3.3. Weakly supervised and bootstrapping-based learning

The minimally supervised learning systems (Brin et al., 1998; Greenwood et al., 2006; Sudo et al., 2003; Yangarber et al., 2000) engender a context within which the concept of information redundancy is used in conjunction with bootstrapping. Through the adoption of different methods, minimally supervised learning systems attempt to estimate the confidence of the learned patterns for relation extraction. Yangarber (2000), Sudo et al. (2003) and Greenwood et al. (2006) are among a myriad of scholars who have opted to calculate domain relevance by relating the frequency of a term in domain relevant documents. Based on bootstrapping, the central goal of prevalent minimally supervised learning systems is to identify relation patterns that can lead to the identification of newer seeds and patterns. Here, the domain relevance of documents is used to discover patterns along with the distribution frequency of said patterns in relevant documents as an indicator of good patterns. Some other bootstrapping approaches (Agichtein et al., 2000; Gupta et al., 2014; Xu et al., 2014, Zhang et al., 2015a; Zhang et al., 2015b) have proven to be effective methods to generate high-precision relation patterns when the set of labeled instances are limited. These works aim to expand an initial "seed" set of instances with new relationship instances.

Xu et al. (2007) presented Domain Adaptive Relation Extraction system (DARE), which is comprised of four major components including linguistic annotation, classifier, rule learning, and relation extraction. The second component, rule learning, is used to identify and extract relations of varying complexities through a seed-driven bottom-up process. Given the fact that DARE might include faulty seeds in the bootstrapping process, its performance demonstrates weaker results when used against unobserved new domains. This is due to the high probability that DARE extracts incorrect rules from the dependency tree during the bootstrapping process. Xu et al. (2010) extend DARE using supervised learning to build seeds by observing learning rules. Despite the immense advances brought on by such an approach for improving precision/recall, there is still need for manual semantic annotation in these approaches. In this thesis, we also present a bootstrapping approach, but in contrast, our work largely avoids such errors by exploiting an initial seed set without the need for manual input for bootstrapping. We automatically build an initial seed set for later iterations based on high confidence patterns from clause patterns extracted from Open IE. To ensure that a seed has high confidence, it is essential for it to be generated by multiple high-confidence patterns.

2.3.4. Distant supervision

The core idea of distant supervision is to learn a classifier based on a set of weakly labeled instances that are often annotated using some heuristics. In the area of relation extraction, the work by Mintz et al. (2009) is among the pioneering works that consider the application of distant supervision techniques. In their work as well as other closely related work such as Surdeanu et al., (2012) and in order to curate the weakly labeled corpus, they use the Freebase knowledge graph whereby for each pair of entities that are related to each other using some Freebase relation, they will identify sentences in their corpus where these entities have been seen together. This way they are able to extract features that can help them train a classifier for relation extraction. One of the challenges of distant supervision methods is the noisy labels, which are generated by the heuristics that will eventually lead to poor relation extraction performance. There have been works by Takamatsu et al. (2012) and Min et al. (2013) among others that propose methods to identify low confidence labels that can be removed or ignored. From a different perspective and in order to augment the work in distant supervision, Riedel et al. (2010) argue that many of the errors produced by relation extraction techniques are due to the generous interpretation of sentence relevance. In other words, if two entities were related to each other through a Freebase relation, any sentences containing these two entities would be considered related and labeled as such. The authors argue that this might not necessarily be the case, especially for cases when the knowledge base is not fully aligned with the corpus. For this reason, they propose the idea of expressed-at-least-once assumption and use constraint-driven semi-supervision without worrying about exactly which sentence expresses the relation.

2.3.5. Matrix Factorization-based Methods

The objective of matrix factorization and collaborative filtering methods in relation extraction is to predict hidden relations that might not have been explicitly observed. Kemp et al. (2006) used Infinite Relational Model (IRM) in order to build a framework to discover latent relations jointly from an n-dimensional matrix. In this matrix, each dimension has a latent structure through which relations can be found. Bollegala et al. (2010) try to explore clusters of entity pairs and patterns jointly as latent relations by employing co-clustering. Takamatsu et al. (2011) use probabilistic matrix factorization with Singular Value Decomposition to reduce dimensions to discover relations. Kolda et al. (2009) and Kang et al. (2012) employ the tensor product of three vectors, i.e., the vectors for two entities and the vector for the relation for decomposing an entity-entity-relation matrix. These authors employed vector of user-query-page for web recommendation and page-page-anchor for web links. Bordes et al. (2013) and Weston et al. (2013) propose a method to determine the first entity vector and its relation vector which leads to the creation of a link to the second entity vector. The goal is to optimize the distance between the second entity vector and the association of the first entity and relation vectors. Riedel et al. (2013) and Yao (2015) use matrix factorization and collaborative filtering by including surface patterns in a universal schema and a ranking objective function to learn latent vectors for tuples of entities and relations. These authors represent each relation as a vector instead of a matrix. Representing each entity as a vector breaks the interaction between two entities. In their work, the authors use surface patterns extracted from existing Open IE systems and predict the hidden relations through matrix completion. Similar to Riedel et al. (2013) and Yao (2015), in this thesis, we employ the notion of universal schemas that is in the form of a collection of patterns derived from OpenIE systems as well as from relation schemas of pre-existing knowledge bases. While previous systems have trained relations only for entities, we exploit advanced features from relation characteristics such as clause types and topic models for predicting new relation instances. Our work can naturally predict any tuple of entities and relations regardless of whether they were explicitly observed at training time with direct or indirect access in their provenance.

2.3.6. Deep learning

In deep learning, several approaches address the task of extracting relations through the use of two major architectures of neural networks, namely Recursive Neural Network (RNN) (Socher et al., 2012; Xu et al., 2015; Yadav et al, 2018) and Convolutional Neural Networks (CNN) (Santos et al., 2015; Zeng et al., 2014; Zeng et al., 2015). These approaches learn the hidden and continuous structures of relations on both internal features such as POS, Chunking, and Syntactic and/or external features such as word embeddings. Socher et al. (2012) presented the RNN model to learn compositional vector representations for phrases and sentences of arbitrary syntactic type and length in order to classify semantic relations between nouns using syntactic paths. Xu et al., (2015) proposed an RNN model by exploiting long short-term memory units (LSTM) and shortest dependency path (SDP) to classify the relation between two entities in a sentence. In their architecture, SDPs are used to retain most relevant information of the sentence while LSTMs are used as multichannel networks that can effectively integrate information from heterogeneous sources over the dependency paths. Zeng et al. (2014) presented a CNN model for relation classification where sentence-level features are learned through a CNN. In their CNN architecture, they extract lexical and sentence level features without complicated NLP preprocessing and assign pairs of words to targeted relations by encode the distances of the features relative to the position of the target noun pairs. Santos et al. (2015) proposed a Ranking CNN model that learns a distributed vector representation for relation classification. The network generates a distributed vector representation for the relations by using a ranking function in order to produce a score for each relation type. Habibi et al. (2017) presented a deep learning NER system to pre-train word embeddings in order to improve biomedical named entity recognition. In this study, the authors used long short-term memory network-conditional random field (LSTM-CRF) based on neural network and word embeddings on 33 datasets with five different entity classes for classification. Moon et al. (2018) presented a deep learning model which works with short text with accompanying images in tweets or snapchat for NER. They constructed NER models based on Bi-LSTM word/character consisting of a deep image network corresponding with visual context information and attention module with adaptive of samples and tokens. Qin et al. (2018) combined deep learning models and distant supervision for RE. The authors exploited reinforcement learning to indicate false-positive instances when inferring distant supervision for each relation type to improve performance on RE. Cui et al. (2018) proposed a neural approach with an encoder-decoder framework to learn highly coherent arguments and relation tuples bootstrapped from Open IE system which is the first research employing deep learning technique to Open IE. Nguyen et al. (2016) proposed neural networks to capture memory matrices for the network training and then used for event trigger and argument prediction. Han et al. (2019) have proposed a deep structured learning framework to learn scoring functions for pair-wise relations based on RNN model and employ a structured support vector machine for temporal relation extraction. The RNN model learns representation of pair-wise relation based on long-term contexts while a structured support vector machine uses domain knowledge to proceed transitive closure of temporal relations. Zhang et al. (2019b) have introduced a method to incorporate word and character level information by using features extracted from a deep neural model for event relation extraction.

Chapter 3

Clause-based OIE with grammatical structure reformation

Within the context of Open Information Extraction (OIE), relation extraction is oriented toward identifying a variety of relation phrases and their arguments in arbitrary sentences. In the plethora of research that focus on the use of syntactic and dependency parsing for the purposes of detecting relations, there has been increasing evidence of incoherent and uninformative extractions. The extracted relations have even been erroneous at times and failed to provide a meaningful interpretation. In this chapter, we propose refinements to the grammatical structure of syntactic and dependency parsing. In lieu of this, we use the English clause structure and clause types in an effort to generate propositions that can be deemed as extractable relations.

3.1. Introduction

Relation extraction (RE) is one of the challenging tasks of natural language processing. Accordingly, relation extraction engines set out to obtain structured relations from unstructured text. In this milieu, the presence of Open Information Extraction (OIE) (Banko et al., 2007; Etzioni et al., 2012; Fader et al., 2011; Nebot et al., 2014; Yahya et al., 2014) offers a more nuanced approach that relies minimally on background knowledge and manually labeled training data. In this respect, various types of relations are taken into consideration without the need to restrict the search to pre-specified semantic relations.

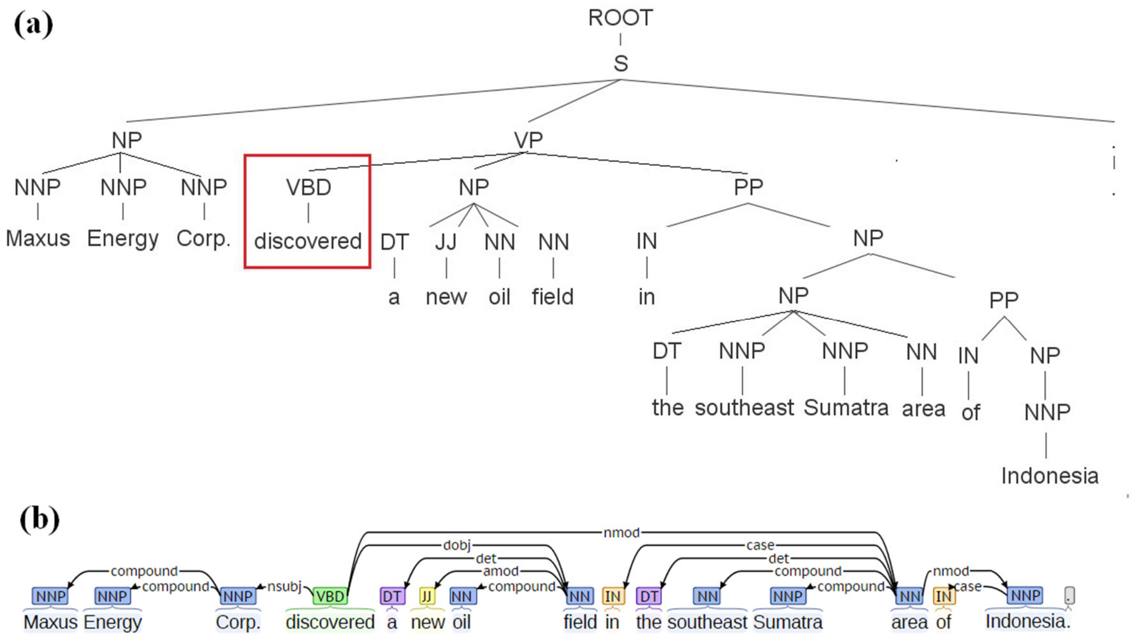


Figure. 3.1. (a) Syntactic and (b) dependency parsing.

In the effort to exploit corpus level statistics, different OIE approaches have incorporated advanced techniques and opted to use handcrafted extraction heuristics or automatically constructed training data. This is aimed at effectively addressing the need to create robust and accurate natural language processing system through learning extractors or estimating the confidence of propositions from a text corpus. Banko et al. (2007), Wu et al. (2010), and Fader et al. (2011) propose a shallow syntactic representation of natural language text in the form of verbs or verbal phrases and their arguments. There has also been a more intense interest in approaches (Corro et al., 2013; Mausam et al., 2012; Garcia et al., 2011; Gamallo et al., 2011) that employ robust and efficient dependency parsing for relation extraction. Various heuristics is utilized to determine relevant segments of information based on dependency parsing analysis by identifying factors that draw attention to whether two chunks of the original sentence exhibit connection, disconnection, or dependence on one another. While shallow semantic representation of natural language text is attained in the form of verbs or verbal phrases and their arguments, there is a generation of research extraction that uses dependency parsing to achieve the same goal. Nonetheless, one of the serious drawbacks of techniques that are restricted to shallow syntactic and dependency analysis is detecting relations that display no connection between the verb or verbal phrases in the sentence.

For instance, consider the following sentence such as "*Maxus Energy Corp. discovered a new oil field in the southeast Sumatra area of Indonesia.*", the syntactic parsing (SP) and dependency parsing (DP) of the sentence displaced in Figure 3.1. Existing state-of-the-art OIE systems the like of OLLIE (Mausam et al., 2012) and ClausIE (Corro et al., 2013) extract relations that are mediated by verbs or verbal phrases based on dependency parsing. Despite key advantages to this approach, the failure to extract all potential relations beyond a pre-defined set of relation including syntactic entities such as nouns and adjectives along with a whole range of verbal structures can be problematic. In the example provide here, the relation between "*southeast Sumatra area*" and "*Indonesia*" cannot be determined by any type of verbs or verbal phrases through either syntactic or dependency parsing. Our work offers a unique advantage in that it is designed to address some of the more pressing limitations inherent in previous OIE systems through the reformation of the grammatical structure obtained from SP and DP. The anticipated outcome of the proposed reformulation of the syntactic and dependency parsing trees is as follows:

- Identifying relations that previous OIE systems have been oblivious to or overlooked altogether, e.g., the relation between "*southeast Sumatra area*" and "*Indonesia*".
- Reducing the number of erroneous relation extractions, e.g., ClausIE erroneously identifies 'there' as a subject of a relation in the following sentence: "In today's meeting, there were four CEOs". This is while, the paradigm proposed in our work allows for the extraction of more precise assertions that allows it to perform complex tasks, which inevitably prevent it from producing such incorrect relations.

In this study, we build on the work presented in Corro et al. (2013) once generating relations based on English grammar clauses (Corro et al., 2013; Quirk, 1985; Thenmozhi, et al., 2015). Essentially, our approach focuses on identifying the set of "clauses" in each sentence. For each clause, the corresponding clause type is determined in accordance to the grammatical function of its coherent constituent. The novel grammatical structure reformation proposed in the current study uses patterns for the determined clause type to generate high-precision relations. Our experiments reveal that the approach utilized in this study essentially improves upon the performance of leading OIE systems such as ClausIE, OLLIE and ReVerb.

3.2. Proposed approach

An OIE system extracts triples in the form of (arg1, rel, arg2), representing basic propositions or assertions from text. In this context, propositions are defined as coherent and non-over-specified pieces of basic information. In our work and inspired by Corro et al. (2013), we focus on the English grammar clause structure in the effort to extract every relation of interest in text. The Oxford dictionary defines clause as "A unit of grammatical organization next below the sentence in rank and in traditional grammar said to consist of a *subject* and *predicate*". While the literature (Corro et al., 2013; Quirk, 1985; Thenmozhi, et al., 2015) is replete with definitions of clause as a part of a sentence that expresses some coherent piece of information, our proposed approach moves beyond this and refines the tree structure produced from syntactic and dependency parsing. We embark on this process by adding necessary relation nodes and removing noise nodes in order to derive a set of coherent constituents for generating propositions that can produce correct extractable relations. An overview of the proposed approach is illustrated in Figure 3.2.

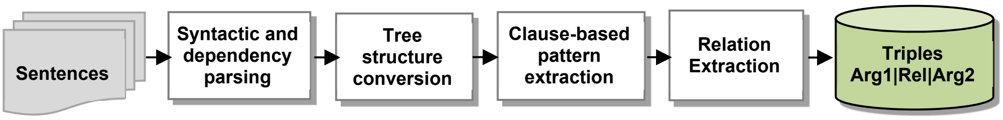

Sentences
Syntactic and dependency parsing
Tree structure conversion
Clause-based pattern extraction
Relation Extraction
Triples Arg1|Rel|Arg2

Figure 3.2. The proposed approach.

3.2.1. Grammatical structure reformation

Considering the fact that a relation candidate is surrounded by words before, between, or after the relation pair, as well as or the combination of two consecutive positions, the clause structure can be posed as a suitable grammatical structure for identifying relations in a sentence (Corro et al., 2013; Quirk, 1985)**.** It should be noted that a clause can consist of different components including subject (S), verb (V), indirect object (O), direct object (O), complement (C), and/or one or more adverbials (A). In our effort to refine the grammatical structure to improve clauses-based relation extraction techniques, the following section offers the details of our proposed approach. The central issues addressed here are related to how the parsing structure should be refined in general and how extra components are added, or the manner in which noise components are removed from the parsing structure.

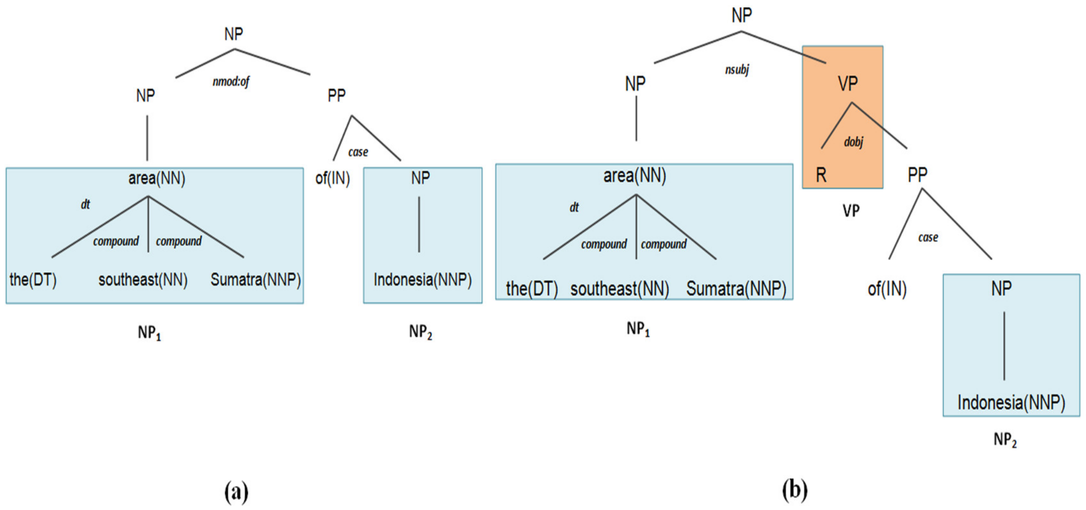


Figure 3.3. (a) Shortest path between "*the southeast Sumatra*" and "*Indonesia*"; (b) the refined tree in sentence "*Maxus Energy Corp. discovered a new oil field in the southeast Sumatra area of Indonesia.*"

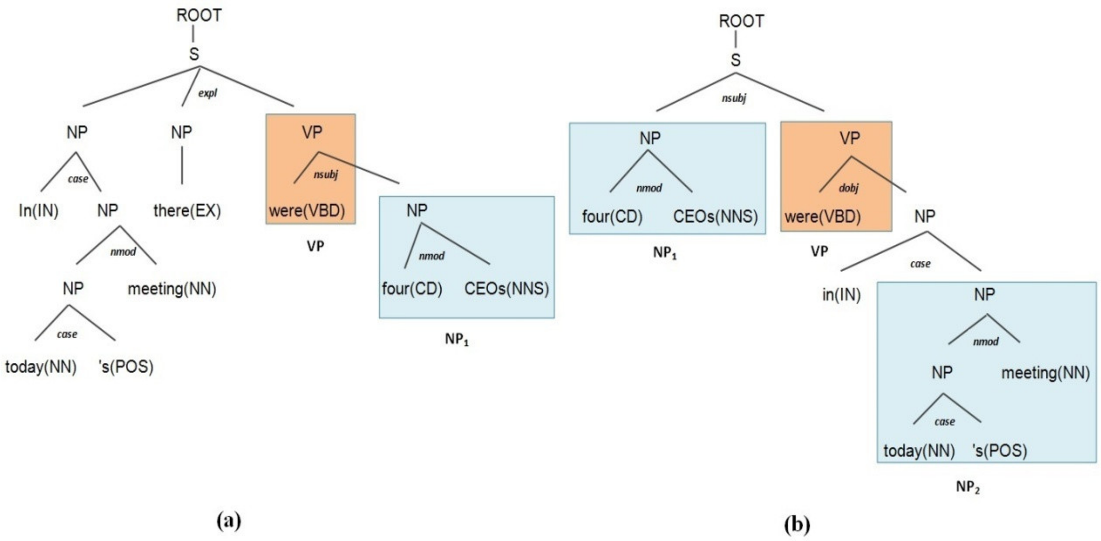


Figure 3.4. (a) Shortest path tree between "*In today's meeting*" and "*four CEOs*" and (b) refined tree in sentence "*In today's meeting, there were four CEOs.*"

Normally relying merely on syntactic or dependency parsing based on verb or verbal phrases to determine relations have a tendency to engender certain problems. This is particularly true in dealing with sentences that do not exhibit sufficient information in order to create a connection between the subject, verb, and object of a relation. For instance, the relation between "*southeast Sumatra area*" and "*Indonesia*" in Figure 1 cannot be determined in the absence of a verb or verbal phrase that is required to describe the relation. Hence, the grammatical tree structure of "*southeast Sumatra area*" and "*Indonesia*" essentially needs to be refined by adding new relation nodes (e.g. a dummy R relation) with associated links between "*southeast Sumatra area*" and "*Indonesia*". This is of significance due to the fact the relation between these structures can come to be explicitly observed and subsequently extracted.

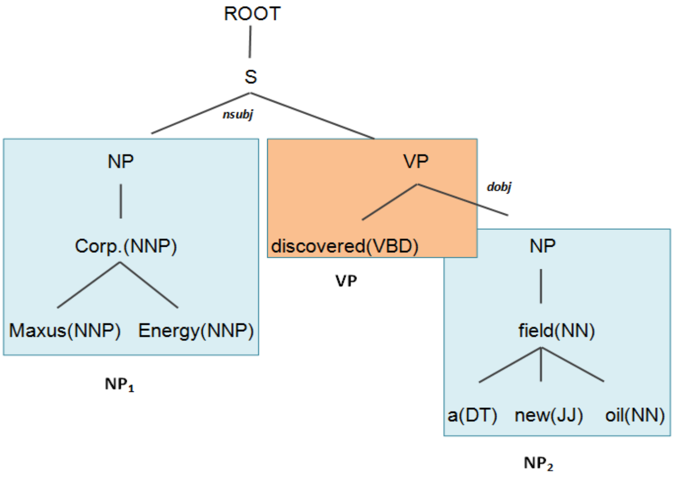


Figure 3.5. Shortest path between "*Maxus Energy Corp.*" and "*a new oil field*" the verb "*discovered*" in "*Maxus Energy Corp. discovered a new old field in the southeast Sumatra area of Indonesia.*"

To this end, we extract the shortest path (Bunescu et al., 2005; Croce et al., 2011) between two potential entity heads detected with SP and DP and use them for grammatical structure reformation. Potential entities are determined by analyzing noun phrases and verb phrases in SP or *nsubj* or *dobj* components in DP. The tree structure was refined in the following manner:

- For cases where the shortest path does not consist of either the "*Subject-Verb-Object*" or "*Subject-Verb-Complement*" structures as shown in Figure 3.3(a), an *R* (relation) node will be added as a central relation in the tree and R will be associated with the other phrases. We also order the nodes of the dependency tree in a way to place the dominant nodes on top and the dependent nodes at the bottom. Figure 3.3(b) depicts the new *R* node that has been added in the tree structure between the *NP* and *PP* phrases, hence creating a relation between "*the southeast Sumatra area*" and "*Indonesia*".

- The next case is when the shortest path NP_1-VP-NP_2 does not have a structure in the form of "*Subject-Verb-Object*" or "*Subject-Verb-Complement*". This is while the structure "*Subject-Verb*" is observed. For instance, in Figure 3.4(a) the two phrases "four CEOs" and "were", are indicative of the relationship "*Subject-Verb*" in DP. In these instances, we propose refining the structure by reversing NP_2 to NP_1. We then add the remaining NP_2 to the structure by identifying the *PP* that has the closest connection with the *VP.* The refined structure of "*four CEOs*", "*were*" and "*in today's meeting*" is demonstrated in Figure 3.4(b). In this context, the intended extraction of the wrong main subject, which in this case is "*There*" is replaced by "*four CEOs*".
- In situations where the shortest path consists of either a *"Subject-Verb-Object*" or *"Subject-Verb-Complement*" structure, any change to the structure of the tree is not required. Our proposition is that the nodes of the dependency tree be ordered in a way so as to place the dominant node at the top and the dependent at the bottom. We recommend keeping the main phrases containing the associated links around the verbal phrase. For example, Figure 3.5 displays a structure where the VP phrase that contains the verb "*discovered*", has associated links with the NPs containing "*Maxus Energy Corp.*" and "*a new oil field*".

On account of these three refinements on the trees, new forms of logical relationships become visible that in turn expedite the extraction of increasingly more accurate relations.

3.2.2. Clause-based relation extraction

As previously pointed out, a clause can consist of different components such as subject (S), verb (V), indirect object (O), direct object (O), complement (C), and/or one or more adverbials (A). As illustrated in Table 3.1, a clause can be categorized into different types base on its constituent components. For instance, the clause type for "*Albert Einstein remained in Princeton*" is SVA with Subject: "*Albert Einstein*", Verb: "*remained in*" and Adverbial: "*in Princeton*". For each clause, we determine the set of coherent derived-clauses based on the syntactic and dependency tree after refining the tree structure. Following Corro et al. (2013), we obtain and exploit clauses for the purpose of relation extraction in the following manner:

Step 1. Determining the set of clauses. According to Corro et al. (2013), we exploit an algorithm for generating clause types shown in Algorithm 3.1. The algorithm will start when finding the subjects (S) and the governor of the verb (V) from DP. This step seeks to identify the clauses in the input sentence by obtaining the head words of all the constituents of every clause. The mapping of syntactic and dependency parsing are utilized to identify various clause constituents. Subsequently, a clause is constructed for every subject dependency, dependent constitutes of the subject, and the governor of the verb. An example of this is the construction of clause relation (subject: "*Maxus Energy Corp.*", verb: "*discovered*", object: "*a new oil field*") in Figure 5 via *nsubj, compound* for the subject, *nsubj* and *dojb* for the verb, and *dojb, dt, amod, compound* for the object. Moreover, subjects like relative pronouns that have been obtained through the *rcmod* dependency (e.g., which or who) and reference a word in the DP or correspond to an artificially created verb are replaced by their antecedent. For instance, in the sentence "*Obama, who is the president of the U.S, came to Canada.*" '*who*' is replaced by '*Obama*' in the extracted patterns.

Table 3.1. Sample clause types (Corro et al., 2013; Quirk et al., 1985); S: Subject, V: Verb, A: Adverbial, C: Complement, O: Object.

Clause types	**Sentences**	**Patterns**	**Derived clauses**
SV	Albert Einstein died in Princeton in 1955.	SV SVA SVA SVAA	(Albert Einstein, died) (Albert Einstein, died in, Princeton) (Albert Einstein, died in, 1955) (Albert Einstein, died in, 1955, [in] Princeton)
SVA	Albert Einstein remained in Princeton until his death.	SVA SVAA	(Albert Einstein, remained in, Princeton) (Albert Einstein, remained in, Princeton, until his death)
SVC	Albert Einstein is a scientist in the 20th century.	SVC SVCA	(Albert Einstein, is, a scientist) (Albert Einstein, is, a scientist, in the 20 the century)
SVO	Albert Einstein has won the Nobel Prize in 1921.	SVO SVOA	(Albert Einstein, has won, the Nobel Prize) (Albert Einstein, has won, the Nobel Prize, in 1921)
SVOO	RSAS gave Albert Einstein the Nobel Prize.	SVOO	(RSAS, gave, Albert Einstein, the Nobel Prize)
SVOA	The doorman showed Albert Einstein to his office.	SVOA	(The doorman, showed, Albert Einstein, to his office)
SVOC	Albert Einstein declared the meeting open.	SVOC	(Albert Einstein, declared, the meeting, open)

Step 2. Identifying clause types. When a clause is obtained, it needs to be associated with one of the main clause types as shown in Table 3.1. In lieu of the previous assertions, we use a decision tree from the ClausIE system to identify the different clause types. In this process, the system marks all optional adverbials after the clause types have been identified. For example, the derived clauses in Table 1 include: (S: "*Albert Einstein*", V: "*has won*", O: "*the Nobel Prize*"), (S: "*Albert Einstein*", V: "*remained in*", A!: "*Princeton*"), and (S: "*Albert Einstein*", V: "*is*", A!: "*smart*"). Optional adverbials "A!" and "A?" indicate essential adverbials and optional adverbials, respectively. Clause types SVC, SVOO, and SVOC are identified solely based on the structure of the clause; all adverbials are optional for these types.

```
Input:   DP analysis
Output:  list of clause patterns

if found(S,V) do
|   if seek(O)
|   |   if seek(direct O) and seek(indirect O) do
|   |   |   generate SVOO
|   |   else if seek(C) do
|   |       |   generate SVOC
|   |       else if seek(A) and seek(direct O) do
|   |           |   generate SVOA
|   |           else generate SVO
|   else if seek(C) do
|       |   generate SVC
|       else if seek(A) do
|           |   generate SVA
|           else generate SV
end if
```

Algorithm 3.1. Clause pattern generation.

Step 3. Extracting relations. We extract relations from a clause based on the patterns of the clause type as illustrated in Table 1. Assuming that a pattern consists of a subject, a relation and one or more arguments, it is reasonable to presume that the most reasonable choice is to generate n-ary propositions that consist of all the constituents of the clause along with some arguments. To generate a proposition as a triple relation (arg1, rel, arg2), it is essential to determine which part of each constituent would be considered as the subject, the relation and the remaining arguments. We initially identify the subject of each clause and then use it to construct the proposition. To accomplish this, we map the subject of the clause to the subject of a proposition relation. This is followed by applying the patterns of the clause types in an effort to generate propositions on this basis. For instance, for the clause type SV in Table 1, the subject presentation "Albert Einstein" of the clause is used to construct the proposition with the following potential patterns: SV, SVA, and SVAA. We then recommend using DP to forge a connection between the different parts of the pattern. As a final step, n-ary facts are extracted by placing the subject first followed by the verb or the verb with its constituents. This is followed by the extraction of all the constituents following the verb in the order in which they appear. As a result, we link all arguments in the propositions in order to extract triple relations. For example, for the sentence "*Albert Einstein died in Princeton in 1955*", we obtain four coherent relation propositions as follows:

- (S: "*Albert Einstein*", V: "*died*")
- (S: "*Albert Einstein*", V: "*died in*", A: "*Princeton*")
- (S: "*Albert Einstein*", V: "*died in*", A: "*1955*")
- (S: "*Albert Einstein*", V: "*died in*", A: "*Princeton*, A: [*in*] *1955*")

3.3. Experimentation

3.3.1. Experimental setup

In this study, we adopt the evaluation strategy proposed in Corro et al. (2013) and use three different benchmark datasets, namely ReVerb, Wikipedia, and the New York Times datasets. In our experiments, we used the Stanford parser to perform SP and DP on the sentences derived from the three standard benchmark datasets. In the first dataset, the ReVerb dataset consisting of 500 sentences have been extracted using Yahoo's random link service with manually labeled extractions from the Web. The sentences may have irrelevant phrases due to noise in the Web texts. The second dataset is comprised of 200 random sentences extracted from Wikipedia. These sentences have a tendency to be shorter and simpler than the ones extracted from the ReVerb dataset. Bearing in mind that a considerable majority of information in Wikipedia is generally edited by non-native speakers, at times the sentences display incorrect grammatical structures, but they are less noisy than the information from the first dataset. The third dataset is comprised of 200 random sentences extracted from the New York Times collections (NYT). While these sentences are generally very clean, they have a propensity to be long and complex. Our experiments were carried out on these benchmark datasets in an effort to analyze the reliability and reproducibility of our work. Additionally, we compared our proposed approach with other OIE systems including ClausIE, OLLIE, ReVerb, WOE and TextRunner. It should be noted that TextRunner and WOE have not publicly disclosed their code base confining us to use the results they reported in their publication for the ReVerb dataset and not for Wikipedia and the NYT datasets, marked as n/a in Table 3.2.

We manually label and verify all of the extraction by ReVerb, OLLIE and ClausIE systems from three datasets. To guarantee consistency among the labels, the ReVerb dataset is relabeled following the original labels from TextRunner, ReVerb, and ClausIE because it covers the entire set of these systems. As for the Wikipedia and the NYT datasets, each extraction is relabeled in accordance to the same output result from ClausIE. Each extraction was labeled by two independent experts, which came to be considered as the gold standard. The experts were instructed to treat an extraction as correct if it was both informative and devoid of extraneous information. The correct extract had to be approved and labeled as correct by both experts. In contrast, the extractions that lacked meaning were labeled as incorrect. The experts' ruling was measured using Cohen's Kappawith 0.57 on the ReVerb dataset, 0.68 on the Wikipedia dataset, and 0.63 on the NYT dataset. Regardless, some incorrect extractions were labeled as correct in ClausIE, e.g., (“*his*”, “*has*”, “*a car*”), (“*John*”, “*works*”, “*at IBM*”), (“*John*”, “*works*”, “*for IBM*”). These relations were subsequently labeled to be incorrect relations in our work.

Table 3.2. Overview of the precision of the six systems.

	ReVerb	Wikipedia	NYT
TextRunner	35.84% (286/798)	n/a	n/a
WOE	43.48% (447/1028)	n/a	n/a
ReVerb	53.37% (388/727)	66.26% (165/249)	54.98% (149/271)
OLLIE	44.04% (547/1242)	41.41% (234/565)	42.46% (211/497)
ClausIE	50.37% (1182/2348)	49.56% (397/797)	52.67% (493/936)
LS3RyIE	67.77% (1642/2425)	68% (614/903)	70.19% (690/983)

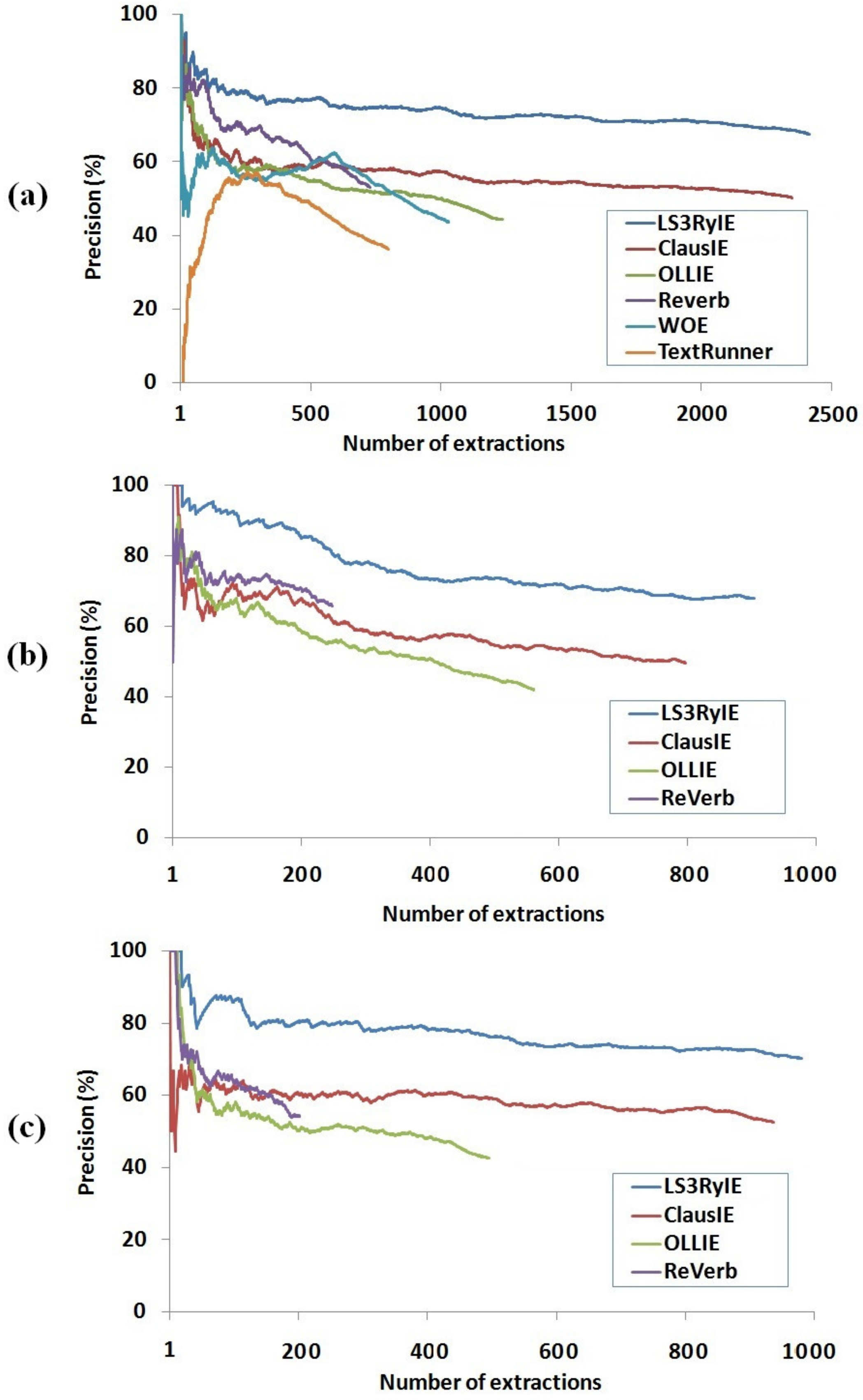


Figure 3.6. Comparative results on (a) ReVerb; (b) Wikipedia; and(c) NYT datasets.

3.3.2. Experimental results

The results of our proposed approach and the comparison to the other state-of-the-art OIE systems are presented in Table 3.2 and Figure 3.6 on the three standard benchmark datasets. Figure 6 plots the precision of each OIE system ordering them in decreasing confidence as a function of the number of extractions. It can be observed the performance of our proposed approach has raised some interesting points as it outperforms ClausIE, OLLIE, and ReVerb. The relative quality differences between our proposed approach and the state-of-the-art OIE systems employed in this study were essentially improved in all three datasets. The results reveal that we obtained 67.77% precision on the ReVerb dataset, 68% precision on the Wikipedia dataset, and 70.19% precision on the NYT dataset. The increase in precision is obtained through the discovery of hidden relations in addition to the removal of unrecognized relations are removed within our proposed grammatical structure. Our system identified 2,425 extractions in the ReVerb dataset, 903 extractions in Wikipedia and 983 extractions in the NYT dataset. These extractions were higher in number compared to the other OIE systems. The precision of TextRunner was significantly lower than the other systems on the ReVerb dataset. The other systems obtain high precision on high-confidence extractions; the precision drops based on low confidence values in each extraction except for the ClausIE system on the NYT dataset (Figure 3.6.c). ClausIE identifies numerous incorrect extractions in possessive clauses, e.g., ("*his*", "*has*", "*a computer*"), with high confidence values, something that is prevented in our work due to the grammatical structure refinements.

3.3.3. Extraction samples and discussion

Several sample relations extracted from a similar sentence using each of the OIE system are demonstrated in Table 3.3. In light of the fact that our proposed approach and ClausIE explore the clause structure of the sentence, the two systems have demonstrated an ability to extract the highest number of relations, which are 12 and 10, respectively. In the process of exploring the clause structure of a sentence, the adverbials in a clause are considered in addition to the verb or verbal phrases adverbials. However, the refinement of the tree structure in our proposed approach has led to improved performance. This helps us discover hidden relations and reduce noise in the identified relations. As seen in the table, relations r_8, r_9, r_{10}, r_{11}, and r_{12} are correct relations based on the results obtained from our system. These relations, which would have otherwise not been identified, have specifically been detected because of the structure refinements proposed in our approach. While ClausIE is capable of obtaining those relations, they have mostly been with other form types. The limitations imposed by DP also leads to extractions by ClausIE that are not correct. As a consequence, ClausIE extracts incorrect relations in r_{13}, r_{21}, and r_{22}. The relations r_9 and r_{10} have only been identified and extracted using our approach due to the fact that the other systems rely on DP to identity the subject in the sentence. Consequently, the afore mentioned systems were unable to recognize "*bombing*" as a subject. As illustrated in Table 3.3, together with ClausIE, our proposed approach was able to correctly identify the subject and henceforth extract a number of correct relations such as r_2, r_3, r_4, r_{15}, r_{16}, and r_{17}.

ReVerb obtained r_{25} and r_{26} which are both incorrect; these relations have been obtained because ReVerb restricts subjects to noun phrases without prepositions and a result incorrectly omits "*...the stabbing was unrelated to the escape plan and stemmed instead from...*" for r_{25} and "*with his lawyers*" for r_{26}. OLLIE makes use of DP and obtains r_{23} and r_{24}, but r_{24} is incorrect because OLLIE fails to correctly identify the subject and object in the structure. WOE, meanwhile, fails to identify verbal phrases because in using DP, a non-informative connection is made between "*said*" and "*stemmed*". Hence, WOE obtains the incorrect r_{28} and r_{29} relations. TextRunner obtains incorrect relations r_{32}, r_{33}, r_{34}, and r_{35}, because it uses POS tagging and chunking for data training. Problems arise for TextRunner when faced with identifying connection words for a relation in a long sentence.

The utilization of three different datasets in our experiments is indicative of the fact that our proposed approach is not overfitted for a specific dataset. The presence of some incorrect extractions in our proposed approach may be due to the incorrect tree obtained from SP and DP. There have been instances where the incorrect DP resulted from noise in the input sentences, including incorrect grammatical structures or the presence of spurious words. For instance, for the incorrect relation r_1, DP incorrectly determines "*Continuing*" to be an adverbial, which in turn leads to a connection being made with the subject "*Salim*". As future work, we will focus on identifying specific cases that can be problematic in DP and SP and further refine the grammatical structure to address these shortcomings.

Table 3.3. Extraction samples with Correct: 1 and Incorrect: 0.

OIE	Triple	Label
Sentence: "*Continuing to maintain his innocence of terrorism charges about a bombing in Iraq, Salim said the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers.*"		
LS3RyIE	r_1: ("Salim", "be continuing to maintain", "his innocence of terrorism charges") r2: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers Continuing to maintain his innocence of terrorism charges") r_3: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers") r_4: ("the stabbing", "was unrelated to", "the escape plan") r_5: ("the stabbing", "was", "unrelated") r_6: ("the stabbing", "stemmed", "instead") r_7: **("the stabbing", "stemmed instead from", "his dissatisfaction with his lawyers")** r_8: **("he", "has", "a innocence of terrorism charges")** r_9: **("a bombing", "was in", "Iraq")** r_{10}: **("his dissatisfaction", "is with", "his lawyers")** r_{11}: **("he", "has", "a dissatisfaction")** r_{12}: **("he", "has", "lawyers")**	0 1 1 1 0 0 1 1 1 1 1 1
ClausIE	r_{13}: ("his", "has", "innocence of terrorism charges") r_{14}: ("Salim", "be Continuing", "to maintain his innocence of terrorism charges") r_{15}: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers Continuing to maintain his innocence of terrorism charges") r_{16}: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers") r_{17}: ("the stabbing", "was", "unrelated to the escape plan") r_{18}: ("the stabbing", "was", "unrelated") r_{19}: ("the stabbing", "stemmed", "instead from his dissatisfaction with his lawyers") r_{20}: ("the stabbing", "stemmed", "instead") r_{21}: ("his", "has", "dissatisfaction with his lawyers") r_{22}: ("his", "has", "lawyers")	0 0 1 1 1 0 1 0 0 0
OLLIE	r_{23}: ("the stabbing", "was unrelated to", "the escape plan") r_{24}: ("the stabbing", "was", "unrelated")	1 0
ReVerb	r_{25}: ("Salim", "said", "the stabbing") r_{26}: ("the stabbing", "stemmed instead from", "his dissatisfaction") r_{27}: ("the stabbing", "was unrelated to", "the escape plan")	0 0 1
WOE	r_{28}: ("Salim", "said stemmed from", "his dissatisfaction") r_{29}: ("Salim", "said unrelated to", "the escape plan") r_{30}: ("the stabbing", "stemmed from", "his dissatisfaction") r_{31}: ("the stabbing", "was unrelated to", "the escape plan")	0 0 1 1
TextRunner	r_{32}: ("Continuing", "to maintain", "Salim") r_{33}: ("Continuing", "to maintain", "his innocence of terrorism charges") r_{34}: ("his innocence of terrorism charges", "said", "the stabbing") r_{35}: ("the stabbing", "was to stemmed from", "his dissatisfaction")	0 0 0 0

3.3.4. System scalability

We have measured several factors in our system that can impact execution time such as the parsing process, clause generation and grammatical structure reformation when the system deals with a large number of sentences. Given the fact that the execution time of the system can depend on sentence type, we have performed our experiments on 3 different sentence types based on their structure, namely short sentences (simple), medium sentence (borderline complex) and long sentence (complex). In short sentences, the numbers of extracted patterns are in the range of 1 to 2 patterns. Medium sentences can produce 3 to 5 patterns while more than 5 patterns are extracted from long sentence. We ran our system on a desktop computer with Intel Core i5 3.1 ghz, 8GB RAM and 1TB hard disk. For time complexity, the parsing process has $O(n)$ to proceed left-to-right over n transition words in the sentence. The process of clause generation can generate i clauses after parsing. This process takes $O(n+i)$. LS3RyIE also performs search for k potential patterns. The system will take $O(i+k)$ for clause generation with i denoting the original clauses and k denoting the new clauses. Thus, the time complexity for LS3RyIE is $O(n+i+2k)$ as shown in Table 3.4. In case when the number of original generated clauses and new generated clauses are equal with the number of transition words, the system has a linear time complexity of $O(n)$.

Table 3.4. Time complexity of LS3RyIE.

Time complexity	Cost
Parsing process	$O(n)$
Generating clauses	$O(n + i)$
Generating clauses with structure reformation	$O(n+i+2k) \sim O(n)$

Table 3.5 shows the detailed execution time of our system. It takes on average 0.447s for the system to process sentences when clause generation is only used, while it takes 0.745s on average when the system includes clause generation as well as grammatical structure reformation. This could indicate that LS3RyIE has no limitation to deal with large numbers of sentences. However, it should be noted that in our work, the results of the extracted patterns are considered at the sentence-level and are therefore independent from the results of the other sentences. Therefore, LS3RyIE can run in parallel on different segments of an input dataset. Hence, if our work is executed on a powerful server, which supports for many more cores than the desktop that we had access to, the execution time will be significantly reduced in LS3RyIE. For instance,

execution time of LS3RyIE can be reduced by 10 folds when the system is run in parallel by ten concurrent threads shown in Figure 3.7. On the same note and for the same reason, due to the fact that our approach performs at a sentence level, it will have limitations in performing co-reference resolution on sentence elements such as pronouns.

Table 3.5. The execution time breakdown for LS3RyIE.

	Parsing time (s)	Generating clause time (s)	Generating clause with structure reformation time (s)
Short sentence	0.016	0.018	0.060
Medium sentence	0.216	0.271	0.319
Long sentence	1.003	1.054	1.857
Mean	0.411	0.447	0.745

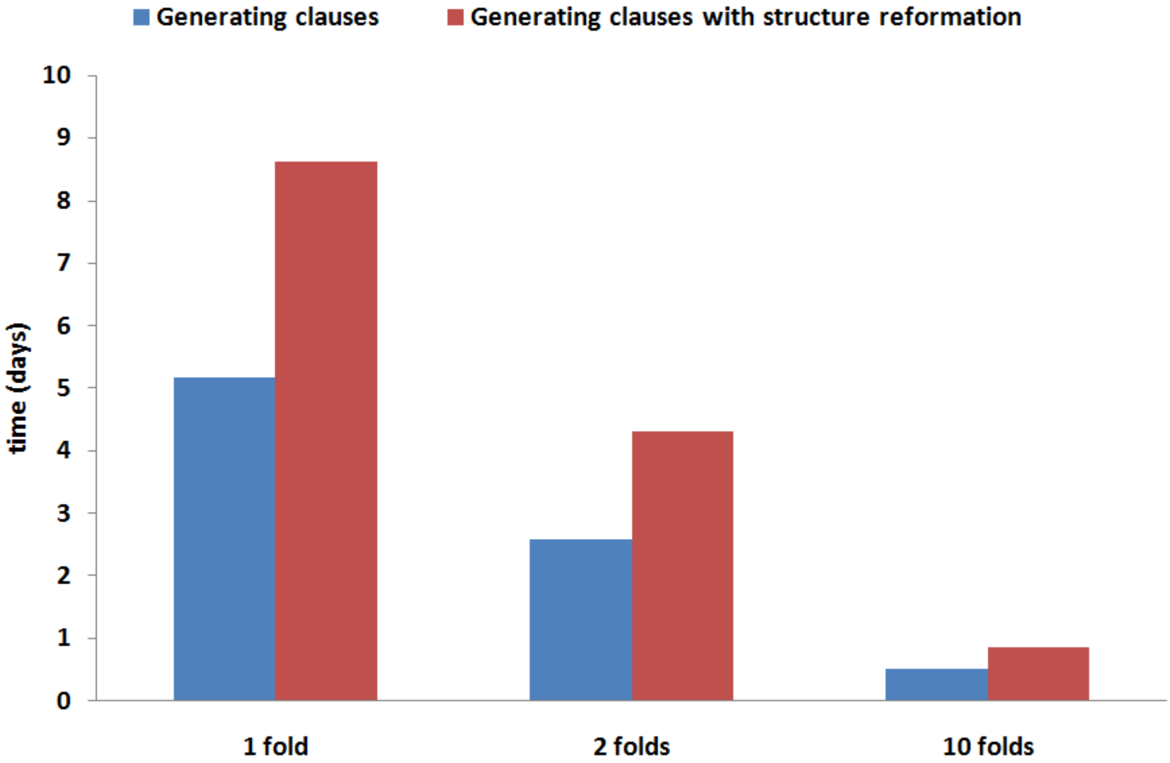


Figure 3.7. Estimated execution time in one million sentences.

3.4. Concluding remarks

In this chapter, we presented a method to address some limitations faced by previous OIE systems through the reformation of the grammatical structure obtained from SP and DP. We used English grammar clauses to identify the set of clauses in each sentence from the refined grammatical structure. In each clause, the corresponding clause type was determined as an extractable relation according to the grammatical function of its coherent constituent. In our extensive experiments, we compared our method with other OIE systems using three standard benchmark datasets. The results reveal that our method obtained not only higher precision extractions but also had more flexible generation of relations over other state-of-the-art OIE systems.

Chapter 4

Self-training on refined clause patterns for relation extraction

Bootstrapping techniques utilized for relation extraction have shown to be effective in terms of interactively expanding a set of initial relations. Such tasks are primarily carried out through semi-supervised classification approaches. Considering that choosing the most efficient seeds are pivotal to the success of the bootstrapping process, these methods depend on a reliable set of seeds or rules that incorporate domain knowledge. In this study, we propose clause-based pattern extraction with self-training for unsupervised relation extraction. Accordingly, we extract patterns based on a clause-based approach that strives to consider all possible clause types that may contain a relation. The proposed self-training algorithm relies on the clause-based approach to extract a small set of seed instances in order to identify and derive new patterns.

4.1. Introduction

Essentially, relation extraction seeks to effectively organize and identify the relevant segments of unstructured text in relation triples. As part of an effort to infer more complex relational structures, relation extraction techniques aim to steer the extraction process away from the ambiguous extractions of semantic relations. This is aimed at representing a particular set of relationships between two or more entities in text that can be used for querying and automated reasoning. To infer more complex relations, several approaches have been proposed, involving supervised, semi-supervised, and unsupervised methods. Among the supervised methods Bunescu et al. (2005), Kambhatla et al. (2004), Ravichandran et al. (2002), and Zhou et al. (2010) have focused on performing language analysis for semantic relation extraction. A running theme among these techniques is the capacity to generate linguistic features based on syntactic, dependency, or shallow semantic structures of the text. Espousing these features, the models are subsequently trained to identify instances of entities that are related through relations. Once the identification process is underway, the extractions are classified based on pre-defined relation types. This is a laborious and time-consuming undertaking on the part of these approaches, involving the analysis of vast quantities of sample data. In contrast, a bootstrapping technique displays a clear advantage by working with unannotated documents and a small initial seed set. Choosing the most efficient seeds is imperative to the success of the bootstrapping process as it is these very seeds that are utilized by the afore mentioned techniques to extract patterns that can represent extractable relations. The "best" learned relations are selected and identified as new instances, which are in turn used to update the seed set. This process is repeated until an explicit and adaptive stopping criterion has been reached.

Bootstrapping based pattern matching approaches have been employed by various researchers (Agichtein et al., 2000; Greenwood et al., 2006; Pantel et al., 2006; Brin, 1998) to extract patterns from seed relations. These approaches exploit the concept of information redundancy and hypothesize that similar relations tend to appear in uniform contexts. A study conducted by Batista et al. (2015) established that semi-supervised bootstrapping techniques used to extract semantic relations from text iteratively expand a set of initial seed relationships. In an effort to find similar relationships, the researchers investigated the effectiveness of bootstrapping for relationship extraction using word embedding. Their evaluation model involved the Named Entity Recognition (NER) module along with weak entity linking by matching entity names with Freebase concepts. In Xu et al. (2007) and Xu et al. (2010), the authors' goal of extracting relations of various complexities and optimizing the learned rules through a seed-driven process is accomplished through bootstrapping with the ability to automatically learn pattern rules from parsed data. The researchers use dependency trees as the input for pattern extraction and work with trees or sub-trees that contain seed arguments. Despite, their eagerness to maintain high accuracy, it is difficult to claim with certainty that the identified patterns are indeed accurate. In lieu of this, there is a probability that faulty seeds could potentially be injected into the bootstrapping process.

To address such limitations, we initially extract patterns organized with a triple structure (Arg1, Relation, Arg2) based on a clause-based Open IE. We automatically build an initial seed set for later iterations based on high confidence patterns. We present a self-training algorithm for relation extraction that automatically learns the required initial seed set without the need for manual input. To ensure that a seed has high confidence, it is essential for it to be generated by multiple high-confidence patterns. Through the iterative expansion of the original seed set, bootstrapping allows for an increasing number of seeds to be identified that can ultimately lead to higher confidence relation extraction patterns. In the section below, we have identified the most significant contributions of our work:

- We present an unsupervised method for relation extraction by combining clause-based pattern extraction and bootstrapping.
- We present a self-training algorithm for the purpose of labeling relations during the bootstrapping process.
- Following in the footsteps of a number of research (Stevenson et al., 2007; Swampillai et al., 2010; Xu et al., 2007; Xu et al., 2010), we apply our proposed method on the standard and widely used Nobel and MUC-6 corpora.

In the sections that follow, we begin by a detailed description of our proposed method in Section 4.2 where we put forth methods used for RE with self-training. Section 4.3 offers an in-depth analysis of our experimental results as compared to the DARE system. In the last section, we draw conclusions about the merits of our work and offer ways to advance the literature.

4.2. Proposed approach

As illustrated in Figure 4.1, our approach consists of two primary tasks constituting pattern identification and self-training through unsupervised bootstrapping. As a first step, we use English grammar clauses to identify the set of "clauses" in each sentence for the purpose of pattern extraction. With respect to each clause, the corresponding clause type will be determined pursuant to the grammatical function of its coherent constituent. The emergent patterns for the determined clause type will be used to extract relations. Subsequently, we propose a new self-training method based on bootstrapping that uses the patterns identified in the first step to automatically derive the required seeds. We learn context clues from the learned seeds and use the clues to identify the category of a particular relation. The approach proposed here eliminates the need for a manually prepared seed set at the onset and instead opts to automatically extract the required seeds from high confidence patterns extracted in the first step.

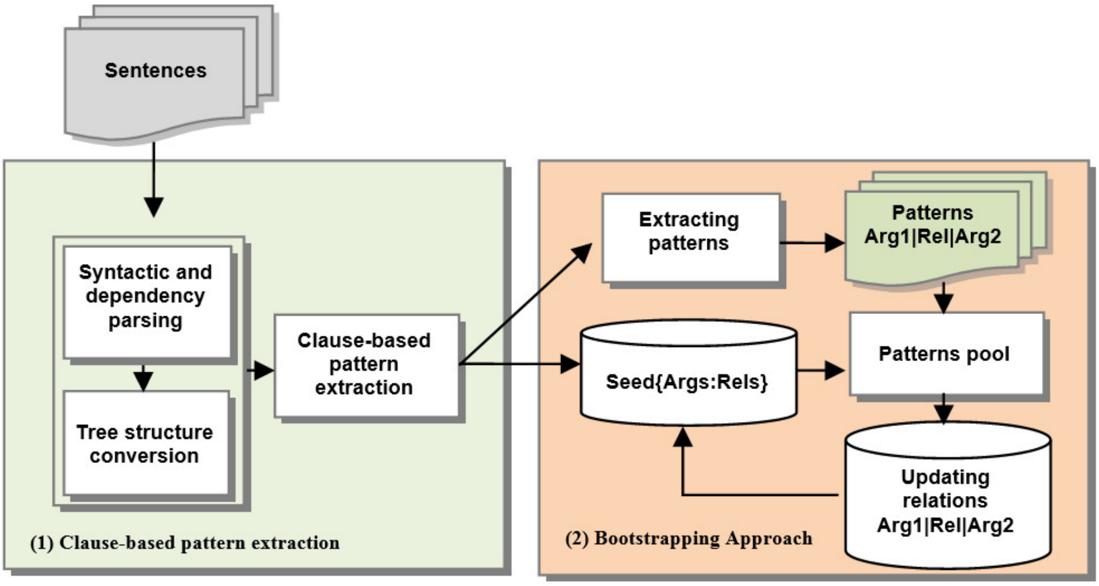

Sentences
Syntactic and dependency parsing
Tree structure conversion
Clause-based pattern extraction
(1) Clause-based pattern extraction
Extracting patterns
Patterns Arg1|Rel|Arg2
Seed{Args:Rels}
Patterns pool
Updating relations Arg1|Rel|Arg2
(2) Bootstrapping Approach

Figure 4.1. The proposed approach.

4.2.1. Clause-based pattern extraction

Considering that a relation candidate is surrounded by words before, between, or after the relation pair, or the combination of two consecutive positions, the clause structure can be deemed as a suitable grammatical form used to identify relations in a sentence (Corro et al., 2013; Thenmozhi et al., 2015; Xu et al., 2013). A clause is part and parcel of a sentence as it embodies pieces of information that are coherent. While not all combinations of a clause's different constituents materialize in the English language, a clause is comprised of a subject (S), verb (V), and optionally a direct object or an indirect object (O), complement (C), and one or more adverbials (A). The extracted patterns will be refined within a new structure through the addition of necessary relation nodes and the removal of noise nodes. This is aimed at making the patterns increasingly more effective for the purpose of deriving a set of coherent constituents directed toward generating relation clauses to serve for bootstrapping. The detail of the processes how to refine grammatical structure and extract clause-based patterns has been described in Chapter 3, Section 3.2.

```
Input: Set of Pattern $P = p_1, p_1, ...p_n$ with $p\{Arg_1, Rel, Arg_2\}$
       $Seed\{Arg, Rel\}$ with $Arg\{arg_1, arg_1,... arg_n\}$ and $Rel\{rel_1, rel_2,... rel_m\}$
Output: selected patterns $P'$

$P' \leftarrow null$
loop $N$ iterations || $P$ != $null$ do
    for all $p \in P$ do
        $Score(p, Seed\{Arg,Rel\})$
    end for
    $pattern\text{-}pool \leftarrow$ top $k$ patterns $p$
    for all $p\{Arg_1, Rel, Arg_2\} \in pattern\text{-}pool$ do
        $Score(arg\in \{Arg1, Arg2)\}, Seed\{Arg, Rel\}, P)$
        $Score(rel\in \{Rel\}, Seed\{Arg, Rel\}, P)$
    end for
    $Seed\{Arg, Rel\} \leftarrow$ top $m$ argument $arg$
    $Seed\{Arg, Rel\} \leftarrow$ top $m$ relation $rel$
    $P' \leftarrow$ top $k$ patterns $p$
    $Update (P)$
end loop
```

Algorithm 4.1. Self-training algorithm.

4.2.2. Self-training algorithm

Broadly speaking, bootstrapping methods begin with an un-annotated corpus and a small set of hand-tagged seed words. In contrast to bootstrapping approaches that require an input seed set, we propose a new self-training method based on bootstrapping that benefits from the patterns extracted from the previous step (Section 5) to identify and extract relations from the corpus. The emergent patterns for the determined clause types will be used to extract specified relations. The method proposed here eliminates the need for a manually prepared seed set at the onset and instead opts to automatically extract the required seeds from high confidence extracted patterns. We learn context clues from the learned seeds and use the clues to identify the category of a particular relation. The words in these relations are assigned to a seed set in order to incrementally complete a lexicon that can be used for further bootstrapping. Our method retrains using the new updated seeds and the process is repeated iteratively. In light of the fact that the extracted patterns are organized with the (a_1, r, a_2) structure, we organize the seeds in the form of $S(E, R)$ where $E = \{a_1, a_2, \ldots, a_n\}$ and $R = \{r_1, r_2, \ldots, r_m\}$.

Our self-training algorithm for bootstrapping is demonstrated in Algorithm 4.1. The basic idea behind our self-training algorithm is that the system takes a few initial selected seeds and a set of patterns T from Open IE as input and learns further patterns based on the initial selected seeds. The algorithm begins by scoring each pattern t of set of patterns T and selecting the top-k scored patterns, which will then be inserted into a *pattern-pool*. In order to score the patterns, we utilize the scoring function introduced in (Thelen et al., 2002; Patwardhan et al., 2007), known as the *RlogF* metric, which has already been used for learning lexicons in previous studies. The *RlogF* metric scores each extracted pattern by calculating the occurrences of the arguments and the relation of a given pattern within the seed set. Equation 4.1 showcases the approach proposed in our work regarding the implementation of the *RlogF* metric.

$$RlogF(t_k) = \frac{\left(F_k^a \times log_2(F_k^a)\right) + \left(F_k^r \times log_2(F_k^r)\right)}{N_k} \tag{4.1}$$

Where F_k^a is the number of argument seeds extracted by pattern t_k, F_k^r represents the number of relation seeds extracted by pattern t_k, and N_k stands for the total number of words extracted by pattern t_k. Immediately after all the extracted patterns are ranked using the *RlogF* metric, the top-k patterns with the highest score are selected and added to the *pattern-pool*. The ensuing step involves scoring the candidate seeds in the top-k selected patterns within the *pattern-pool*. Candidate seeds comprise of nouns, compound nouns, and verbs observed in the arguments and relations of the extracted patterns. For each candidate seed, the algorithm collects all the patterns used to produce the candidate seed in question. Our algorithm scores the candidate seed by computing the average number of patterns that are extracted by that seed. Equations (4.2) and (4.3) detail the scoring method:

$$AvgLog(a_i) = \frac{\sum_{j=1}^{P_i} log_2\left(F_j^a + F_j^r + 1\right)}{P_i} \tag{4.2}$$

$$AvgLog(r_i) = \frac{\sum_{j=1}^{P_i} log_2\left(F_j^a + F_j^r + 1\right)}{P_i} \tag{4.3}$$

where P_i is the number of patterns that extract e_i/r_i. Also, F_j^a and F_j^r denote the number of entity seeds and relation seeds extracted by pattern j, respectively. Candidate a_i and r with the high score in the *pattern-pool* will be added to S. In each iteration, top-k

patterns from the *pattern-pool* are selected and removed until the *pattern-pool* is depleted of any patterns. It is through this process that we obtain a list of updated patterns allowing us to categorize other patterns as they are extracted. The algorithm is repeated until no more patterns emerge.

4.3. Experimentation

4.3.1. Experimental setup

For benchmarking our approach in this task, we conducted experiments on two widely used datasets, namely the Nobel Prize and MUC-6 corpora shown in Table 4.1. The content of the Nobel Prize corpus is comprised of reports from the New York Times, BBC Online, and CNN News. The data, proposed by Xu et al. (2007), is available[1] for evaluation purposes which were extracted from the Nobel Prize website[2]. The corpus comes in two parts consisting of NobelPrize A (1999-2005) and NobelPrize B (1981-1998), based on the timestamp of the content as proposed by Xu et al. (2007). Nobel Prize A are records of newspapers extracted from 1981 to 1998 and Noble Prize B are records of online news extracted from 1999 to 2005. The targeted relations for the experiments are binary to quaternary relation such as <Recipient, Prize, Area, Year>, e.g., ("*Albert Einstein*", "*was awarded*", "*Nobel Prize for Physics*", "*1921*").

The MUC-6 corpus is smaller than the Nobel Prize corpus and describes events related to '*the person who obtained a position*' and '*the person who left a position*'. The targeted relations are defined with several factors such as: (1) PersonIn: The person who is currently in a position or the person who obtains a new position; (2) PersonOut: The person who left a position; (3) Position: The position which a person has worked or the position which a person has left; (4) Organization: the company where the person has worked or has left. The gold standard of the relations in MUC-6 is available for evaluation purposes. The gold standard contains 200 documents separated into test and training sets. The training dataset (MUC-6a) consists of 256 events in an additional 100 documents; the test dataset (MUC-6b), meanwhile, presents 227 events in 100 documents. We adopt the evaluation strategy proposed in Xu et al. (2007) and use the abovementioned datasets for evaluation. It should be noted that when using the MUC-

---

[1] http://dare.dfki.de

[2] http://nobelprize.org

6 corpus, Xu et al. only evaluated their work based on the training dataset, which consisted of 256 events.

We separately extract patterns in two datasets with all OIE systems such as ReVerb, OLLIE, ClausIE and LS3RyIE. The output of these systems will be used for the bootstrapping process (BT) for identifying relations. Given the fact the approach is a self-training-based bootstrapping model, which does not require the manual specification of the initial seed set, it is worth mentioning what initial seed sets were determined in the first step of our approach for each of the two corpora. For setting seeds of the bootstrapping process, we automatically extract the patterns with the highest confidence value in each OIE systems to build the seed set and use the pronouns and compound nouns observed in these patterns for the argument seed set, together with the verbs in the patterns for the relation seed set. As a result, we obtained the following argument seed set for the Nobel Prize corpus: {*Peace, Nobel, Medicine, Literature, Laureate*} as well as the relation seed set: {*won, awarded*}, and the argument seed set of {*President, Chief, Officer*} along with the relation seed set of {*appointed, named, succeeded, retired*} for MUC-6.

Table 4.1. The employed corpora.

Corpus	Number of Documents	References
Nobel Prize:		
Nobel Prize A (1999-2005)	2,296	Xu et al. (2007)
Nobel Prize B (1981-1998)	1,032	Xu et al. (2010)
MUC-6:		Xu et al. (2007)
MUC-6a (training)	256	Stevenson (2007)
MUC-6b (testing)	227	Swampillai et al. (2010)

In the bootstrapping process, the outcome of each iteration is updated and used in the training for subsequent iterations. The number of candidate relations and seeds should be determined for selecting in each iteration. Normally, the candidate relations will not be selected if they have a low score. A higher score for a candidate relation will show that the candidate has a higher significance. To proceed between iterations, the algorithm needs to define how many suitable candidate relations need to be added in each iteration. If the number of selections is not enough, the algorithm will stop when no more candidate relations are found. To this end, we have selected two different values for each of the two configurable parameters, namely the number of extracted patterns (#p) and the number of added seeds (#s) in our experiments as shown later in this section.

4.3.2. Experimental results

We first extract patterns from Nobel Prize and MUC corpuses with all OIE systems such as ReVerb, OLLIE, ClausIE and LS3RyIE. Table 4.2 summarizes the output of these systems. In the bootstrapping process, the number of iterations is set based on the number of relevant output patterns. We ran the algorithm with a number of iterations according to the number of relevant relations in each OIE systems. For instance, in case of (#p=10, #s=5) for Nobel Prize A, the algorithm has been ran with 130 iterations in the ReVerb+BT, 300 iterations in OLLIE+BT, 300 iterations in ClausIE+BT, and 420 iterations in LS3RyIE+BT. Experiments are applied in two cases with (#p=5, #s=3) and (#p=10, #s=5).

Table 4.2. Extractions by OIE systems.

Corpus	# Relevant _relations	# patterns
Nobel Prize A		
ReVerb	1052	3925
OLLIE	2832	8179
ClausIE	2924	12309
LS3RyIE	4238	14606
Nobel Prize B		
ReVerb	478	1718
OLLIE	1365	3749
ClausIE	1345	5665
LS3RyIE	1857	6545
MUC-a		
ReVerb	103	252
OLLIE	327	511
ClausIE	402	781
LS3RyIE	438	863
MUC-b		
ReVerb	131	269
OLLIE	294	497
ClausIE	364	700
LS3RyIE	384	788

Tables 4.3-4.6 show the bootstrapping results on each OIE extractor on the Nobel Prize and MUC corpuses. In every case, recall increased and precision decreased until the F-measures produced significant results in a reasonable number of iterations. Regarding Nobel Prize corpus, OLLIE+BT and ClausIE+BT produced lower scores

than ReVerb+BT while the best results were obtained by LS3RyIE+BT with F-measure of 67.01% (#p=5, #s=3) for Noble Prize A, and F-measure of 73.19% (#p=5, #s=3) for Noble Prize B. In terms of MUC corpus, Tables 4.5-4.6 indicate that LS3RyIE+BT and OLLIE+BT perform better than ReVerb+BT and ClausIE+BT. More specifically, OLLIE+BT obtained its best value with the F-measure of 67.4.3% (#p=5, #s=3) for MUC-6a, and its best value with the F-measure of 68.94% (#p=5, #s=3) for MUC-6b while LS3RyIE+BT obtained the best results with 68.23% of F-measure (#p=5, #s=3) for MUC-6a, and 70.38 % of F-measure (#p=5, #s=3) for MUC-6b.

Table 4.3. Performance on Nobel Prize A.

	p=10 and s=5				p=5 and s=3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	110	61.81	64.67	63.19	180	66.33	56.74	61.16
	120	60.00	68.44	63.94	200	65.60	62.35	63.93
	130	58.00	71.67	64.11	220	64.09	67.01	65.52
OLLIE	260	66.65	61.19	63.80	520	64.46	59.18	61.71
	280	65.14	64.41	64.72	560	62.85	62.15	62.50
	300	63.13	66.87	64.98	600	61.00	64.61	62.76
ClausIE	240	70.00	57.45	63.11	520	65.57	58.31	61.73
	260	68.61	61.01	64.59	560	64.53	61.79	63.13
	280	66.85	64.01	65.40	590	63.59	64.16	63.87
LS3RyIE	340	74.99	60.09	66.69	720	69.91	59.39	64.23
	360	73.11	62.10	67.19	800	67.00	63.23	65.06
	380	71.68	64.28	67.78	880	64.59	67.06	65.80

Table 4.4. Performance on Nobel Prize B.

	p=10 and s=5				p=5 and s=3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	40	74.75	60.77	67.04	80	75.00	60.97	67.26
	45	70.00	64.02	66.87	90	73.55	67.27	70.27
	50	66.20	67.27	66.73	100	70.80	71.95	71.37
OLLIE	120	71.33	62.71	66.74	240	68.25	60.00	63.86
	130	69.76	66.45	68.06	260	66.54	63.37	64.91
	140	67.07	68.79	67.92	270	66.07	65.34	65.71
ClausIE	100	72.5	53.90	61.83	220	70.18	57.39	63.15
	110	70.09	57.32	63.06	240	67.58	60.29	63.73
	120	67.58	60.29	63.73	260	65.69	63.49	64.57
LS3RyIE	120	83.00	53.63	65.16	340	74.64	68.36	71.35
	140	80.85	60.69	69.51	360	73.39	71.13	72.24
	190	72.37	74.04	73.19	380	72.26	73.93	73.09

4.3.3. Comparison

We compared the performance of all OIE systems with bootstrapping (best cases) with DARE (Xu et al., 2007) as a baseline on both Nobel Prize and MUC-6 corpora. Note that we did not compare our method with (Xu et al., 2010) because the authors used a supervised learning method to build their seeds. Results on ReVerb+BT are only used to compare its bootstrapping performance with other OIE systems due to its limitation on extracting patterns. Table 4.7 demonstrates the performance of the best cases from our proposed method and the best results reported in Xu et al. (2007). The DARE system obtained 59.36% and 46.31% of F-measure in Nobel Prize A and Nobel Prize B and 54.1% of F-measure in MUC-6a. The authors used two different sets of seeds for the two corpora of Nobel Prize. The results from bootstrapping with OIE systems confirm that we succeeded in improving upon Xu et al.'s work. In Nobel Prize, OLLIE+BT, ClausIE+BT and LS3RyIE+BT obtain 64.95 %, 65.41% and 67.78% in F-measures for Nobel Prize A, and 67.92%, 64.58%, 73.19% in F-measures for Nobel Prize B, respectively. LS3RyIE+BT obtained better results compared with other baselines where the system improved 8.42% and 26.88%, respectively, over the Nobel Prize A and Nobel Prize B corpora in F-measures compared to the baseline.

Table 4.5. Performance on MUC-6a.

	p=10 and s=5				p=5 and s=3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	9	59.34	52.42	55.67	20	56.00	54.36	55.17
	12	53.71	63.10	58.04	25	49.60	60.19	54.39
	15	52.08	72.81	60.72	30	49.32	69.90	57.83
OLLIE	25	69.20	52.91	59.97	60	60.00	55.05	57.41
	30	66.67	61.16	63.79	70	61.43	65.74	63.51
	35	64.29	68.81	66.49	80	61.25	74.92	67.40
ClausIE	35	66.57	57.96	61.96	70	66.57	57.96	61.96
	40	64.00	63.68	63.84	80	65.75	65.42	65.58
	45	62.00	69.40	65.49	90	63.11	70.64	66.67
LS3RyIE	40	68.50	62.55	65.39	80	69.25	63.24	66.10
	45	65.77	67.57	66.67	90	66.00	67.80	66.89
	50	63.60	72.06	67.80	100	64.00	73.05	68.23

Table 4.6. Performance on MUC-6b.

	p=10 and s=5				p=5 and s=3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	9	65.68	51.14	57.51	20	58.00	44.27	50.21
	12	56.25	48.09	51.85	25	59.20	56.49	57.81
	15	58.59	57.25	57.91	30	58.67	67.17	62.63
OLLIE	25	64.4	54.76	59.19	50	66.00	56.12	60.66
	30	61.67	62.92	62.28	60	64.67	65.98	65.31
	35	62.29	74.15	67.70	70	63.43	75.51	68.94
ClausIE	35	50.85	48.90	49.86	70	49.71	47.80	48.74
	40	53.75	59.06	56.28	80	52.00	57.14	54.45
	45	56.00	62.23	61.92	90	53.11	65.66	58.72
LS3RyIE	40	65.5	68.23	66.83	80	70.4	68.75	69.56
	45	62.22	72.92	67.15	85	68.82	73.95	70.21
	50	60.60	78.90	68.52	88	65.91	75.52	70.38

As for the MUC-6 corpus shown in Table 4.8, the baseline (DARE system) succeeded in producing 54.1% of F-measure in MUC-6a. OLLIE+BT, ClausIE+BT and LS3RyIE+BT obtain 67.4%, 66.66%, 68.23% in F-measure for MUC-6a, and 68.94%, 61.92%, 70.38% in F-measure for MUC-6b, respectively. Performance wise, LS3RyIE+BT has also established its superiority to DARE with a margin of 14.13% in

F-measure on the MUC-6a corpus. It should be pointed out that Xu et al. only conducted experiments on the MUC-6a with 256 events with a large number of seeds (55 seeds) and did not perform experiments or report results on MUC-6b. It is also worth noting that in DARE high precision is obtained at the cost of recall. Our system has succeeded in addressing such limitation and overcoming it.

Table 4.7. Overall comparison on Noble Prize domain.

Methods	Precision (%)	Recall (%)	F-measure (%)
Nobel A			
DARE (baseline)	71.60	50.70	59.36
ReVerb+BT (#p=5, #s=3)	64.09	67.02	65.52
OLLIE+BT (#p=10, #s=5)	63.13	66.87	64.95
ClausIE+BT (#p=10, #s=5)	66.85	64.02	65.41
LS3RyIE+BT (#p=10, #s=5)	71.68	64.28	67.78
Nobel B			
DARE (baseline)	83.80	32.00	46.31
ReVerb+BT (#p=5, #s=3)	70.80	71.95	71.37
OLLIE+BT (#p=10, #s=5)	67.07	68.79	67.92
ClausIE+BT (#p=5, #s=3)	65.69	63.49	64.58
LS3RyIE+BT (#p=10, #s=5)	72.36	74.04	73.19

Table 4.8. Overall comparison on MUC-6 domain

Methods	Precision (%)	Recall (%)	F-measure (%)
MUC-6a			
DARE (baseline)	62.00	48.00	54.10
ReVerb+BT (#p=10, #s=5)	52.08	72.81	60.73
OLLIE+BT (#p=5, #s=3)	61.25	74.92	67.40
ClausIE+BT (#p=5, #s=3)	63.11	70.65	66.66
LS3RyIE+BT (#p=5, #s=3)	64.00	73.06	68.23
MUC-6b			
DARE (baseline)	n/a	n/a	n/a
ReVerb+BT (#p=5, #s=3)	58.67	67.18	62.63
OLLIE+BT (#p=5, #s=3)	63.43	75.51	68.94
ClausIE+BT (#p=10, #s=5)	56.00	69.23	61.92
LS3RyIE+BT (#p=5, #s=3)	65.91	75.52	70.38

4.3.4. Errors analysis and discussion

One of the important considerations that demand an in-depth analysis is the required number of iterations for extracting patterns. As discussed in the literature (Thelen et al., 2002; Patwardhan et al., 2007; Xu et al., 2010), there are no standard techniques for determining the right or exact number of iterations; therefore, in our work, we terminate the process after so many iterations until the best F-measure is obtained. Given the fact that LS3RyIE extracts a higher number of patterns compared to the other systems, on average, it requires a higher number of iterations to reach its best F-measure. Therefore, our observation is that the higher the number of extracted patterns is, the more iterations it would take on average to reach the best F-measure. We ran our system on a desktop computer with Intel Core i5 3.1 ghz, 8GB RAM and 1TB hard disk. For time complexity, in Self-training algorithm the system scores all m patterns in P with k iterations ($1<k<m$) (Algorithm 4.1), which will take O(mk). In each iteration, the system will collect i seeds of arguments and j seeds of predicates that will cost $O(ik + jk)$. For updating the set of patterns P and P' (Lines 13-14 in Algorithm 4.1), the process takes O($2k$). Thus, the time complexity of the system would be $O(mk+ ik + jk +2k) \sim O(k(m+ i + j))$ shown in Table 4.9. In case when seeds of arguments and relations are equal ($i, j = m/k$), the self-training process has a time complexity of O($m(k+1/k)$) ~ O(mk).

Table 4.9. Time complexity of LS3RyIE+BT.

Time complexity	Cost
Open IE	$O(n)$
Self-training	O($m(k+1/k)$) ~ $O(km)$

Now from a performance perspective, despite the better performance of our method compared to DARE, we are aware of potential drawbacks that our work suffers from. While we explored the results produced by our method on a case-by-case basis, we have become aware of errors in the pattern extraction step. Specifically, some errors stemmed from incorrect parsing of the input sentences in the clause-based pattern extraction method. In certain cases, the incorrect parsing resulted from the noise in the input sentences, including erroneous grammatical forms or spurious words. Table 4.10 shows the total output errors (t_errors) encountered including the grammatical errors (g_errors). We report both the number of errors and the percentage of errors in the table. These errors include patterns that contain incoherent information or have wrong grammatical structure. Most of the g_errors occur when the parser fails to correctly disambiguate a noun, verb or adjective in a sentence. For example, the verb "award" is often detected to be a noun. Errors in ClausIE and LS3RyIE are higher than OLLIE due to a higher number of extracted patterns. As discussed in Chapter 3 Section 3.3.3, the number of errors in LS3RyIE is less than the errors in ClausIE primarily because of the structure reformation process. Even though the number of errors in our proposed method is not significant, we are aware of these deficiencies and their impact on the F-measure. We anticipate that the use of increasingly more robust SP and DP methods in the future will translate into improved performance for pattern extraction.

4.4. Concluding remarks

In this chapter, we presented a clause-based pattern extraction approach with self-training for unsupervised relation extraction. Initially, we proposed an approach for extracting patterns that might contain relations. We used English grammar clauses to identify a set of clauses in each sentence from a refined grammatical structure. In each clause, the corresponding clause type was determined as an extractable relation in accordance to the grammatical function of its coherent constituent. The identified relations were then used to construct a seed set. Based on the identified seeds, we proposed a self-training algorithm that extracts more relations based on the initial seed set. We conclude that our proposed method has produced competitive results on both Nobel Prize and MUC-6 corpora, which is indicative of its efficiency.

Table 4.10. Errors analysis.

Cases	ReVerb+BT		OLLIE+BT		ClausIE+BT		LS3RyIE+BT	
	#g_errors /#t_errors	g_rate /t_rate(%)	#g_errors /#t_errors	g_rate /t_rate (%)	#g_errors /#t_errors	g_rate /t_rate (%)	#g_errors /#t_errors	g_rate /t_rate(%)
Nobel Prize A								
(#p=10,#s=5)	24/48	1.85/3.70	56/103	1.86/3.43	115/259	4.11/9.25	95/202	2.50/5.32
(#p=5,#s=3)	27/46	1.64/2.80	50/101	1.65/3.35	124/273	4.18/9.22	102/225	2.32/5.11
Nobel Prize B								
(#p=10,#s=5)	15/28	3.00/5.60	33/61	2.35/4.35	53/105	4.41/8.75	61/150	3.21/7.89
(#p=5,#s=3)	13/23	2.47/4.38	27/55	1.99/4.07	61/110	4.69/8.46	58/135	3.05/7.11
MUC-6a								
(#p=10,#s=5)	14/24	9.68/16.60	19/46	5.44/13.16	26/52	5.77/11.55	26/46	5.2/9.20
(#p=5,#s=3)	16/25	10.94/17.10	31/63	7.75/15.75	19/49	4.29/10.88	22/47	4.4/9.40
MUC-6b								
(#p=10,#s=5)	23/27	14.56/21.09	22/38	6.28/10.85	39/52	8.66/11.55	13/19	2.60/3.80
(#p=5,#s=3)	29/35	19.30/23.30	26/39	7.22/10.83	38/51	8.44/11.33	10/12	2.67/2.72

Chapter 5

Feature-based matrix factorization for relation extraction

Relation extraction aims at finding meaningful relationships between two named entities from within unstructured textual content. In this chapter, we define the problem of information extraction as a matrix completion problem where we employ the notion of universal schemas formed as a collection of patterns derived from open information extraction systems as well as additional features derived from grammatical clause patterns and statistical topic models.

5.1. Introduction

In relation extraction, most supervised (Kambhatla et al., 2004; Zhou et al., 2007; Zhou et al., 2010; Won et al., 2015; Barrio et al., 2017) and semi-supervised (Xu et al., 2007; Xu et al., 2010; Zhang et al., 2015a) extraction methods use a predefined, finite and fixed schema of relation types (such as located-in or founded-by). A running theme among these techniques is the capacity to generate linguistic features based on syntactic, dependency, or shallow semantic structures of the text. The main strategy used in supervised methods is to generate linguistic features based on syntactic, dependency, or shallow semantic structures of text. Based on these features, the models are then trained to identify pairs of entities that might be related through some relation, and then to classify them based on a predefined set of relation types. In contrast, semi-supervised techniques employ an initial seed set of, often manually labeled, relations, which are used to extract patterns that can extract additional relations from text. The newly extracted relations based on the learnt patterns are then iteratively used to update the initial seed set and the process is repeated until certain stopping criterion is met. These approaches require corpora that include sufficient example relation instances that might be time consuming to prepare.

Unsupervised techniques, often referred to as Open Information Extraction (Open IE), (Etzioni et al., 2008; Banko et al., 2007; Wu et al., 2010, Mausam et al., 2012; Corro et al., 2013) focus on extracting relations with minimal domain-dependent background knowledge and the least amount of annotated training data. In this context, distant supervision (Angeli et al., 2014; Riedel et al., 2013; Surdeanu et al., 2012) techniques exploit information from external knowledge bases, such as Freebase, in order to perform large-scale relation extraction from text. Distant supervision approaches often avoid dependence on training samples by using natural language grammatical structures or semantic word senses to define and model universal schemas. To this end, Riedel et al. (2013) have presented a matrix factorization model based on universal schemas for predicting relations. These authors presented a set of models that learn lower dimensional manifolds for tuples of entities and relations with a set of weights in order to capture and model relationships between relation types. In this context, the output of the first generation of Open IE systems, such as TextRunner (Banko et al., 2007) and WOE (Wu et al., 2010), are used for building the universal schemas.

While approaches based on universal schemas have shown reasonable performance, their limitation is in that they are trained for relationships between specific entity tuples and relation types, and therefore, are limited when an insufficient number of explicit evidences is present for each relation type for specific entity tuples. For instance, the relation (“*Hawking*”, “*professor-of*”, “*Cambridge Univ.*”) could not help us infer a similar yet unobserved relation (“*Wiles*”, “*professor-of*”, “*Oxford Univ.*”) due to the differences of the entity tuples, i.e., (“*Hawking*”, “*Cambridge Univ.*”) and (“*Wiles*”, “*Oxford Univ.*”). Furthermore, such systems also fall short in predicting other relation types between the same entity tuples, e.g., (“*Obama*”, “*is-president-of*”, “*U.S.*”) and (“*Obama*”, “*has-returned-to*”, “*U.S.*”), which are between the same entity tuples but with different relation types. In addition, these approaches learn linear chain models based on unlexicalized features such as part of speech or shallow tags to label the intermediate words between pairs of potential arguments for identifying extractable relations. However, they do not employ deeper linguistic analysis on the grammatical structure of a sentence such as clause level analysis and therefore, they might suffer from problems such as extracting incoherent and uninformative relations. Therefore, it is possible to summarize the limitations of the current work on universal schemas as follows:

- The matrix models built based on the universal schemas are trained to predict relation types between specifically observed entity tuples; hence, same relation types cannot be predicted for other different yet semantically related entity tuples;
- Similarly, the relation type between entity tuples is predicted based on a matrix factorization model where the participation of entities in other already observed relations determines an unobserved relation and as such other semantically relevant yet unobserved relations cannot be determined;
- Universal schemas are primarily built based on part of speech tags and shallow analysis of the textual content; however, deeper linguistic analysis such as considering entity and relation context within the sentence structure, e.g., sentence clause structure, is not yet considered.

5.2. Research Objectives and Contributions

In light of the above limitations, while existing work based on universal schemas and matrix models have a reasonable precision in retrieving correct relations, they do not perform as well on the recall measure. In other words, given matrix models, such as matrix factorization, require substantial amount of evidence to draw conclusions, and also the fact that there are often very limited set of explicit evidence supporting relation instances, these models fall short in showing good recall performance. In order to address this challenge and improve the recall of existing work, we propose that additional features that can serve as implicit indicators of relation instances need to be introduced. To this end, we introduce and exploit new features based on grammatical clause types and statistical topic models to enrich universal schemas used in the matrix factorization model for predicting new relation instances. Particularly, we exploit clause types and topic models to predict relations regardless of whether they were explicitly observed at training time with direct or indirect access. This allows us to make predictions on relation types that have not been explicitly observed in the training corpus that can hence lead to improved recall performance.

Our work uses the concept of universal schemas from (Riedel et al., 2013) in order to convert the knowledge base information combined with Open IE patterns into a binary matrix representation where entity tuples form its rows and relations are represented as columns. More concretely, the contributions of our work can be enumerated as follows:

- We propose numerous matrix models with fully enriched features such as word context, selectional preference, clause types and statistical topic models and employ matrix factorization with direct/indirect references for predicting specific relations between entities. We show that the consideration of sentence clause types as well as information from statistical topic models enables us to identify and determine semantically relevant yet explicitly unobserved relations between entity tuples, as mentioned above;
- We employ and evaluate the impact of four state-of-the-art Open IE systems used for constructing and populating the initial matrix models that represent the relations between entity tuples and relation types and show how the characteristics and performance of these systems impact the outcome of our proposed approach.
- We systematically evaluate our proposed features in isolation and in tandem within the matrix factorization model and study their impact for identifying relations between entity tuples. We compare our work with the state of the art based on the widely used gold standard by Angeli et al. (2014), which consists of 40 different relation types and over 22,000 relation instances. We also use two silver standard corpora collected from Wikipedia and New York Times to perform additional experiments.

The rest of this chapter is organized as follows. An overview of the proposed approach is presented in Section 5.3 where the description of the features employed for relation extraction is shown. Section 5.4 presents the formal description for computing and incorporating these features into a matrix-based model. This is followed by an in-depth discussion of experimental results in Section 5.5 where the results are compared to the state-of-the-art and the impacts of different Open IE systems on performance are studied. Section 5.6 finalizes the chapter with conclusions and future work.

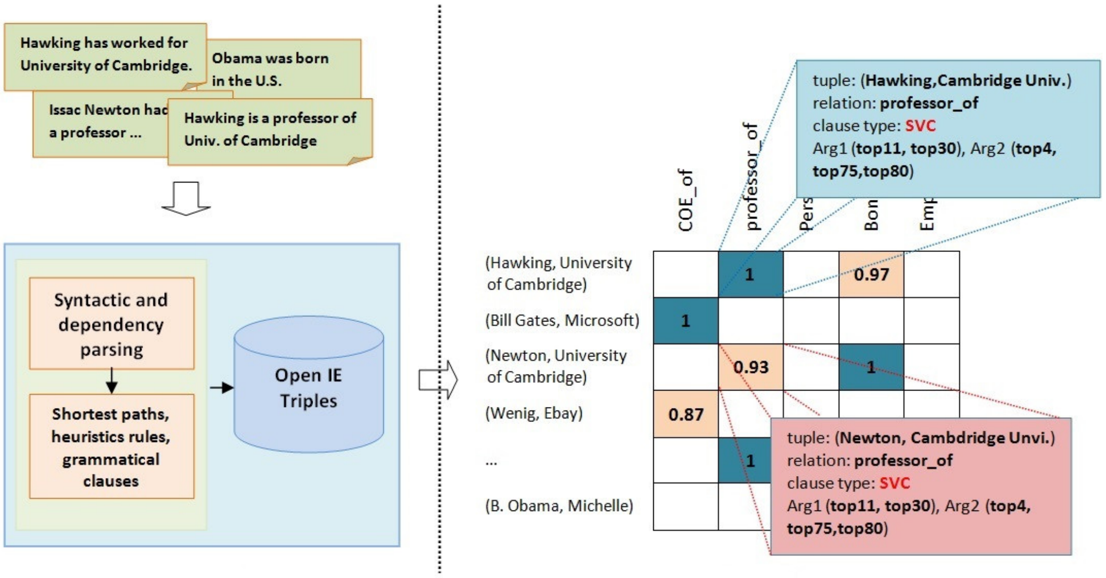
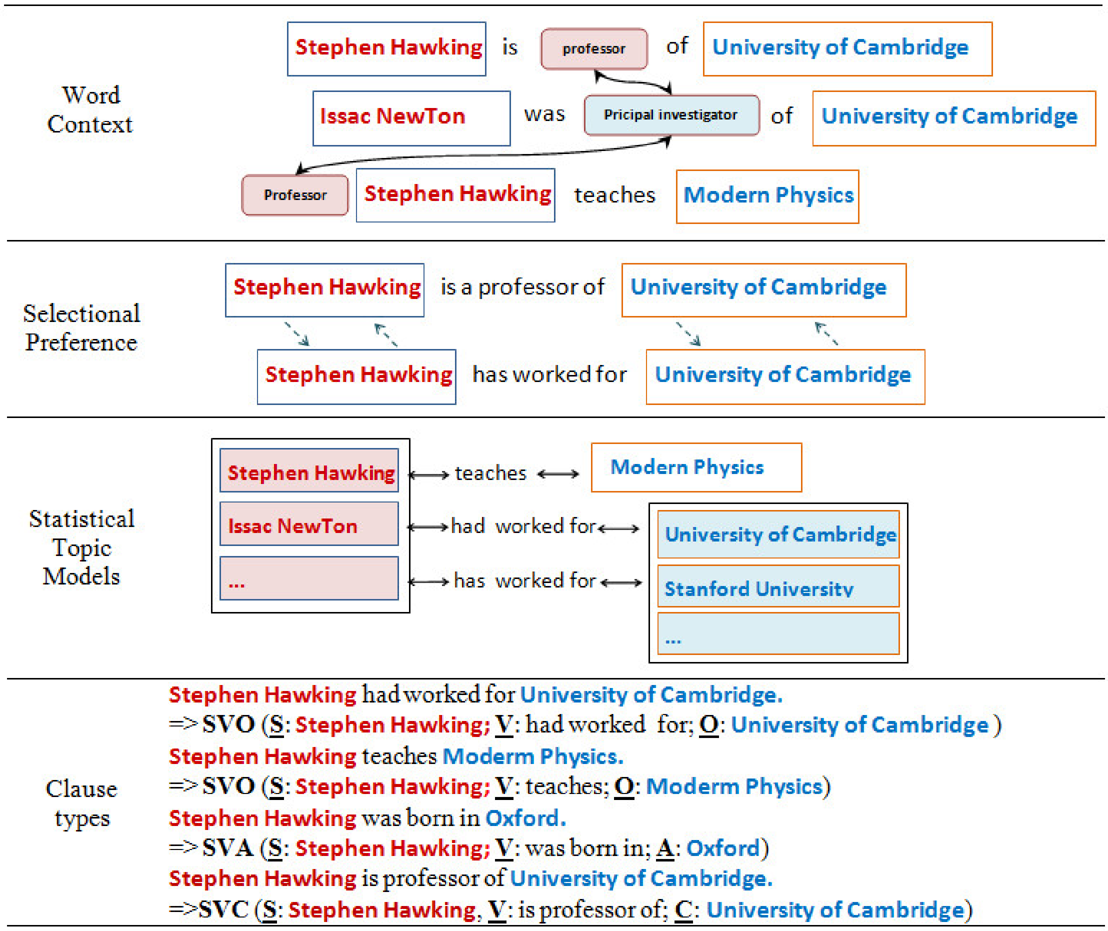

Hawking has worked for University of Cambridge.
Obama was born in the U.S.
Issac Newton had a professor ...
Hawking is a professor of Univ. of Cambridge
Syntactic and dependency parsing
Shortest paths, heuristics rules, grammatical clauses
Open IE Triples
COE_of
professor_of
Pers
Bon
Emp
tuple: (Hawking,Cambridge Univ.)
relation: professor_of
clause type: SVC
Arg1 (top11, top30), Arg2 (top4, top75,top80)
(Hawking, University of Cambridge)
(Bill Gates, Microsoft)
(Newton, University of Cambridge)
(Wenig, Ebay)
...
(B. Obama, Michelle)
1
0.97
1
0.93
1
0.87
1
tuple: (Newton, Cambdridge Unvi.)
relation: professor_of
clause type: SVC
Arg1 (top11, top30), Arg2 (top4, top75,top80)

Figure 5.1. Overview of the Proposed Approach.

5.3. Overview of our Proposed Approach

The main premise of the work in this chapter is that existing Open IE systems are able to automatically extract a set of relatively stable relations from a textual corpus based on some heuristic patterns such as analysis of grammatical structure of the sentence in the form of dependency or syntactic parsing or part of speech tagging. While Open IE systems have shown acceptable performance, they may not be able to extract all possible relations from the text especially in cases when the arguments of the relation have not been explicitly observed within the contexts or forms expected by the Open IE system. However, it could be possible to capitalize on the relations that have been extracted by Open IE systems with specific relation types to explore the possibility of inferring other unobserved relations between entities. One of the systematic ways of achieving this objective would be to view this task as a matrix completion process whereby the rows of the matrix are entity pairs and the columns are relation types. Such a matrix could be partially filled based on the relations derived by Open IE systems and other potential relations between other entity pairs and relation types could be identified by the matrix completion process. Figure 5.1 shows an overview of this process where a collection of documents are fed into Open IE systems whose output relations are then used to build the matrix representation in which the explicit relations extracted by the Open IE system would be denoted by cells consisting of 1 between pairs of entities and relation types.

In our work and inspired by Riedel et al. (2013), we find unobserved yet valid relations by using probabilistic matrix factorization to efficiently estimate vector embeddings for both entities and relation types through stochastic gradient descent optimization. The probability of assigning a relation type to an entity pair is determined by the dot-product of the corresponding embeddings, mapped through a logistic function. While matrix factorization will project the relation matrix into two matrices that exhibit some latent structure in the identified relations, which we refer to as the latent feature, we additionally define four other types of features to further enhance the performance of the matrix factorization method, namely word context feature, selectional preference feature, statistical topic model-based feature and clause-type feature. It is possible to perform the matrix completion task based on each individual feature as well as the integration (interpolation) of these features. The overview of each of the features is provided in the following.



Stephen Hawking had worked for University of Cambridge.
=> SVO (S: Stephen Hawking; V: had worked for; O: University of Cambridge)
Stephen Hawking teaches Moderm Physics.
=> SVO (S: Stephen Hawking; V: teaches; O: Moderm Physics)
Stephen Hawking was born in Oxford.
=> SVA (S: Stephen Hawking; V: was born in; A: Oxford)
Stephen Hawking is professor of University of Cambridge.
=>SVC (S: Stephen Hawking, V: is professor of; C: University of Cambridge)

Figure 5.2. The four feature types employed in our work.

Word Context Feature: The first type of feature that we consider is the context in which a pair of entities or the relation type of that relation happen. We define context to be the set of words observed before or after an item of interest such as the entities in a relation instance or the relation type. The reason it is important to consider word context as a feature in our work is because the objective of our work is to identify unobserved relations and hence if an entity pair is frequently observed within the same context as another entity pair for which we already explicitly observed some relation type, then it would be possible to deduce that such relation might also exist between the entities of the first pair. Poon & Domingos (2009) and Koren (2009) have argued that those words, which occur in similar contexts tend to have similar meanings, which allows us to argue that it would be possible to assume that the entities of two similar entity pairs would be related to each other with similar relation types. For instance, "*Professor*" and "*Principal Investigator*" are often seen in similar relation instances shown in Figure 5.2. Therefore, it would be possible to probabilistically infer that two entities that are related to each other through the "*Professor*" relation could also be related to each other through the "*Principal Investigator*" relationship as well. In the matrix model, the word context feature could be used to define a neighborhood relationship between relations based on the similarity of their contexts.

Selectional Preference Feature: The motivation behind this feature rests on the understanding that only specific types of entities can be used to fill in the relation argument roles for any given relation type. In the context of relations, selectional preference can refer to the constraints that relation types impose on their arguments. The application of selectional preference allows one to not only refine incorrectly identified relation instances, but also identify entity pairs that satisfy the constraints of a relation type and hence can act as a candidate for serving as a relation instance. As an example, consider the relation type "*Professor*" in Figure 5.2 that requires its arguments to be entities of type scholar and academic institution. Therefore, any pair of entities of type scholar and academic institution would be intuitively a candidate for a relation instance of type "*Professor*". It is clear that this feature will result in the extraction of many false positive relations; however, it can be envisioned that when used in conjunction with other features such as the word context feature, many irrelevant relation candidates will be ruled out. Therefore, the selectional preference feature can be seen as an effective tool for increasing the recall of the relation extraction model.

Statistical Topic Model-based Feature: The third type of feature that we consider is in essence a semantic extension of the word context feature. While neighborhood is defined based on the similarity of entity pair and relation type contexts in the word context feature, in the statistical topic model-based feature, neighborhood is defined based on the membership of entity pairs and relation types to the same topics derived by a topic model. In other words, two entities can be considered to be semantically related to each other if they belong to the same topics. As such, it is possible to probabilistically assume that entities within the same topic as the entities participating in a relation instance could participate in a similar relation type. The underlying reason for this is that topic modeling methods provide a probabilistic framework based on term co-occurrences within the documents of a given corpus. Topic models produce probability distributions over words in topics and documents in topics, which can assist in identifying highly similar terms based on their co-occurrence in similar topics. In other words, topics derived by topic models could be seen as the semantic clustering of terms based on which a neighborhood model could be defined. For instance, as shown in Figure 5.2, the neighborhood model would deduce that ("*Isaac Newton*", "*teaches*", "*Physics*") given an explicitly observed relation in the corpus ("*Stephen Hawking*", "*teaches*", "*Physics*") and the fact that "*Isaac Newton*" and "*Stephen Hawking*" were observed in the same topic derived by the statistical topic model on the document collection. In the matrix model, the topic model-based information is used to define a neighborhood model.

Clause-Type Feature: The previous three features are primarily included to benefit from some form of context similarity to infer a neighborhood model for determining unobserved yet reasonable new relations. This feature, however, defines context based on the grammatical role that entity pairs or relation types play within a given sentence. Considering the fact that sentence clause structure has already been shown to be a suitable grammatical structure for identifying relations within a sentence (Corro & Gemulla, 2013; Quirk et al, 1985), we employ clause types and clause components in this feature. Technically speaking, a clause can consist of different components including subject (S), verb (V), indirect object (O), direct object (O), complement (C), and/or one or more adverbials (A). As demonstrated in Corro & Gemulla, (2013) and Quirk et al. (1985), a clause can be categorized into different types based on its constituent components. Given a clause in a relation, it is possible to determine its type of relation via relation presentation of S, V, O and C such as SVO, SVC, SVOO depicted in Table 3.1 Section 3. Given the clause type of the relation instance, one possibility for generating new relation instances is to use entity pairs that have appeared in similar clause patterns within the corpus to form a new relation instance of that relation type. Similar to the selectional preference feature, the clause-type feature can also lead to increased recall at the expense of a higher false positive rate, which can be mitigated when integrated with the other features.

5.4. Formalization of the Proposed Approach

The objective of our work is to predict the hidden relations by completing the schema in the matrix built from surface patterns and fixed relations. Using the same notation as (Riedel et al., 2013; Yao, 2015), we use T and R to correspond to entity tuples and relations. Given a relation $a_r \in R$ and a tuple $e_t \in T$, the objective of our work is to derive a fact about a relation $a_r$and a tuple of two entities e_t. A matrix is constructed with size $|T| \times |R|$ for relation instances. Each matrix cell presents a fact as $x_{r,t}$ and is a binary variable. The variable in each cell of the matrix is 1 when relation a_r is true for the tuple e_t, and 0 when relation a_r is false for e_t. We aim at predicting new relations that could potentially hold for tuple of entities, which are missing in the matrix. We present several models based on the features introduced in the previous section to address the task as follows.

5.4.1. Model-based on the Latent Feature (F model)

The model based on the latent feature derives the latent relations based on the matrix factorization approach, hence we refer to it as the *F model*, where we denote each relation by a_r and each tuple of entities as e_t. We measure compatibility between relation a_r and tuple e_t as the dot product of two latent feature representations of size k. Thus, we have:

$$\theta_{r,t}^{F} = \sum_{k} a_{r,k} e_{t,k} \tag{5.1}$$

The formula is factorizing a matrix into a multiplication of two matrices $\Theta = AE$, A denoting the lower dimension matrix of a_r, and E representing the lower dimension matrix of e_t based on PCA (Collins et al., 2001). Thus, a model with the matrix $\Theta = (\theta_{r,t}^{F})$ of natural parameters is defined as the low rank factorization AE. To estimate the values in PCA, we have:

$$\sigma\left(\theta_{r,t}^{F}\right) = \sigma\left(\sum_{k} a_{r,k} e_{t,k}\right) \tag{5.2}$$

Here, we are applying a logistic function $\sigma(\theta^F_{r,t})$ = 1/(1 + exp($-\theta^F_{r,t}$)) (Collins et al., 2001; Koren et al., 2009; Riedel et al., 2013) to model a binary cell in the matrix. Each cell is drawn from a Bernoulli distribution with natural parameter $\theta^F_{r,t}$. Following Yao et al. (2015), adding a prior for the parameters, the gradient with respect to $\theta^F_{r,t}$ is as follows:

$$\left(1 - (\theta^F_{r,t})\right)\frac{\partial}{\partial_{\theta^F_{r,t}}}\theta^F_{r,t} - \lambda_{\theta^F_{r,t}}\theta^F_{r,t} \tag{5.3}$$

Applying gradients of $\theta^{\mathrm{F}}_{\mathrm{r,t}}$ with regards to the parameters $a_{r,k}$ and $e_{t,k}$, we have:

$$\frac{\partial}{\partial_{e_{t,k}}}\theta^F_{r,t} = a_{r,k} \tag{5.4}$$

$$\frac{\partial}{\partial_{a_{r,k}}}\theta^F_{r,t} = e_{t,k} \tag{5.5}$$

We maximize the log-likelihood of the observed cells under a probabilistic model to learn low dimensional representations as:

$$\max \sum_{r}\sum_{\neg r} \log\left[\sigma(\theta^F_{r,t} - \theta^F_{\neg r,t})\right] \tag{5.6}$$

The representations a_r and e_t can be found by maximizing the log-likelihood using stochastic gradient descent. In this model, the existence of a certain relation type between an entity pair is estimated based on the value determined for the corresponding matrix cell through the optimization problem. As explained later in our experiments, cells with scores above a predefined threshold are considered as true.

5.4.2. Model-based on the Word Context Feature (N model)

Based on the word context feature and within our matrix formalization, a relation in a column could be neighbor to some other co-occurring relation (Koren, 2009) (hence called the *N model*). For example, the relations "*Professor-of*" and "*Principal-Investigator-of*" are often seen in similar contexts. Therefore, the word context feature is essential to capture the localized correlation of the cells in the matrix to incorporate this information. We implement a neighborhood model N via a set of weights *w* of features based on co-occurrence of information around tuples of entities, e.g., headword "*Professor*" often appears in tuples of entities in relations such as "*Dr.*" and "*Principal*

Investigator". In this model, each cell is scored based on the set of weights between this cell and its associated neighbors. This leads to the following formulation:

$$\theta_{r,t}^{N} = \sum_{k} w_k f_k \left(a_{r\prime}, a_r\right) \tag{5.7}$$

where w_{k} is the weight of association between $a_{r\prime}$and a_r; $f_{\mathrm{k}}(a_{r\prime}, a_r)$ defines a conjunctive feature between relation a_r and neighbor relation $a_{\mathrm{r}\prime}$ and k is the number of relations $a_{r\prime}$ that have the exact same tuples as $a_{r\prime}$.

In this model, we additionally employ clause-based feature and integrate it with the word context feature. For instance, a relation ("*Hawking*", "*professor-of*", "*Cambridge*") or ("*Hawking*", "*theoretical-physicist-at*", "*Cambridge*") could be presented by a clause type "Subject-Verb-Complement", while another relation ("*Hawking*", "*born-in*", "*Cambridge*") is in the form of a "Subject-Verb-Adverb" clause. Therefore, considering only entities will fail to predict relations for the tuple ("*Hawking*", "*Cambridge*"). We have used clause types in Open IE (Corro et al., 2013; Vo & Bagheri, 2018) when extracting surface patterns for the matrix. We can interpolate the confidence for a given tuple and a specific relation based on the trueness of other similar relations for the same tuple. Measuring compatibility of an entity tuple and relation amounts to summing up the compatibilities between each argument slot representation and the corresponding entity representation. We extend the neighborhood model to incorporate clause types, which is presented as follows:

$$\theta_{r,t}^{NC} = \sum_{k} \left[w_k f_k \left(a_{r\prime}, a_r\right) + w_{k,v} f_{k,v} \left(a_{r\prime}, a_r\right) \right] \tag{5.8}$$

where $f_{k,v}(a_{r\prime}, a_r)$ defines a conjunctive clause v between relation a_r and neighbor relation $a_{r\prime}$ and $w_{k,v}$ is the weight of association between $a_{r\prime}$and a_r.

5.4.3. Model-based on Selectional Preference (E model)

Earlier, Riedel et al. (2013) introduced the use of entities in collaborative filtering based on a similar idea to the word context feature but geared specifically for entities. In their method, they employed entities to predict latent relations, hence we refer to it as the *E Model*. The model embeds each entity and relation type into a low dimensional space of size *k*. For binary relations such as a_r between a pair of entities $e_t = (e_t^1, e_t^2)$, the relation a_r and the arguments e_t^1 and e_t^2, are modeled in a low dimensional space of size *k*. The equation below leads to the calculation of the compatibility of tuple of entities and their relations by summing up the presentation of each argument slot. Thus, this leads to:

$$\theta^{E}_{r,t} = \sum_{k} a_{r,k}\, e^{1}_{t,k} + \sum_{k} a_{r,k}\, e^{2}_{t,k} \tag{5.9}$$

Analogous to the Neighbor model, we augment the entity model with clause-based features, which enhances the entity model as follows:

$$\theta^{EC}_{r,t} = \sum_{k} a_{r,k}\, e^{1}_{t,k} v_{t1,r} + \sum_{k} a_{r,k}\, e^{2}_{t,k} v_{t2,r} \tag{5.10}$$

where $v_{t1,r}$ is clause type for argument e^1, and $v_{t2,r}$ is clause type for argument e^2.

5.4.4. Model-based on the Statistical Topic Models (T model)

In the Entity model, selectional preferences are employed based on each argument's slot representation and the corresponding entity representation in order to learn from other relations. However, in addition to this, many relations can be considered to be related to other relations based on the probability of being observed within the same topic. For instance, the relation tuple ("*Hollande*", "*France*") could be learned from the observed relation ("*Obama*", "*U.S.*"), if and when "*Obama*"-"*Hollande*" and "*US*"-"*France*" are observed in the same topics, respectively. Therefore, relations can further be learned by their observations within topics. This helps to determine more relations that are missing when learning from directly observed relations. We use Latent Dirichlet Allocation (Blei et al., 2003, Phan et al., 2011) to generate topics, and then embed this information in the matrix. Let $h = \{t_1, t_2, t_3, ..., t_m\}$ and $t_i = \{e_1, e_2, e_3, ..., e_n\}$ be the vector of the set of topic models and each topic model generated from LDA. The entities in the same topic model could be used to refer to others. E.g., the entity "Obama" presented in topic t_1 and t_3 and "*Hollande*" presented in topic t_1 and t_5 could be referenced due to existing in the same topic t_1. We embed each entity into a low dimensional space if they are mapped together within similar topics. We measure each cell based on the compatibility of the argument representation and their corresponding topic with other cells. This can be more formally represented as:

$$\theta^{T}_{r,t} = \sum_{k} a_{r,k}\, e^{1}_{t,k} h_{e_1} + \sum_{k} a_{r,k}\, e^{2}_{t,k} h_{e_2} \tag{5.11}$$

where h_{e_1} denotes the vector of topics where argument e^1 belonged to, and h_{e_2} denotes vector of topics where argument e^2 belonged to.

Given the fact that using only topics could be noisy for training purposes, we also further augment the topic model with clause-based features. For instance, ("*Hollande*", "*France*") can be learned from ("*Obama*", "*U.S.*") if they are present in similar clause types. This could be formulated as:

$$\theta_{r,t}^{TC} = \sum_k a_{r,k}\, e_{t,k}^1 h_{e_1} v_{h_{e1}} + \sum_k a_{r,k}\, e_{t,k}^2 h_{e_2} v_{h_{e2}} \tag{5.12}$$

where $v_{h_{e1}}$ is the vector of clause type for topic argument e^1, and $v_{h_{e2}}$ is the vector of clause type for topic argument e^2.

It is important to mention that while models *F*, *N* and *E* have been proposed earlier in (Riedel et al., 2013), our work is focused on proposing and investigating the role of *T* and *C* in this process and systematically proposing how these two types of features can be interpolated with other feature types. The objective is to explore whether *T* and *C* features are able to address the recall limitation of the state of the art features.

5.4.5. Model Interpolation and Parameter Estimation

Each of the above models represents a unique and important aspect of the data that needs to be combined with other models to predict potential relations in the matrix. In practice, combining the introduced models can capture different necessary aspects of the data. For instance, the combined model of Entity and Neighbor can take advantage of selectional preference on argument slot presentation from the Entity model and the weight of the related neighbors from the Neighbor model. We linearly interpolate the models, e.g., the combination of *F*, *N*, *E* and *T* models can be shown as follows:

$$\theta_{r,t} = \theta_{t,r}^F + \theta_{t,r}^N + \theta_{t,r}^E + \theta_{t,r}^T \tag{5.13}$$

Similar to the F model, relation cells in the matrix model are parameterized through weights and/or latent component vectors. In each model, we predict a relation with a number between 0 and 1. However, the models require negative training data for the learning process. We train the models by ranking the positive cells (observed true facts) with higher scores than the negative cells (false facts). The log-likelihood setting could be contrasted with this constraint that primarily requires negative facts to be scored below a defined threshold. Thus, it is possible to calculate the gradient for the weights of cells. We also use log-likelihood as the objective function and employ stochastic gradient descent with a logistic function $\sigma(\theta_{r,t}) = 1/(1 + \exp(-\theta_{r,t}))$ to learn the parameters $x_{r,t} = \sigma(\theta_{r,t})$.

5.5. Experiments and Evaluation

In order to benchmark our proposed approach, we perform two sets of extensive experiments. In the first experiment, we use the dataset provided by Angeli et al. (2014). The advantage of this dataset is that it already consists of a gold standard of relations that can be used for evaluation purposes. In the second experiment, we use two corpora from Wikipedia and NYTimes. Unlike the dataset from Angeli et al., these two corpora do not have gold standard relation instances. For this reason, we extract silver standard relation instances from these two corpora based on the relation patterns provided by the PATTY project[3]. Furthermore, given our work requires grammatical clause information, we used the work in (Corro et al., 2013) to extract the clause patterns and then check them with entity tuples annotated in each sentence in order to embed them into the matrix. For embedding clause types into the matrix, we use three fundamental clause types, namely SVO, SVC and SVA. The details of these clauses are presented in Corro et al. (2013). Given we only focus on three clause types, if a tuple of entities was extracted with a different clause type, e.g., "*Bill has worked for IBM since 2010*" that corresponds to the SVOA clause pattern, we check the main entities of the relation's corresponding elements and convert its clause type into one of the three main types of clauses. In this case, SVOA will be converted into SVO because "*Bill*" represents S and "*has worked*" denotes V, and "*IBM*" represents O.

Table 5.1. Details of the gold standard corpus.

Corpus	Relation types	#relations
Angeli et al.'s dataset	40	22,765

Additionally, for extracting topics, we generate and estimate topic models based on LDA through Gibbs Sampling using GibbsLDA++[4]. We optimize three important parameters *a*, *b* and number of topics *T* in the LDA. It is based on the number of topics and the size of the vocabulary in the document collection, which are *a* = 50/*T* and *b* =

---

[3] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/patty/

[4] http://gibbslda.sourceforge.net

0.01, respectively (Phan et al., 2011). Then we vary topic sizes between 100, 150, and 200. We evaluate each group of topics and select topic size 150, which shows the best performance for our experiments.

5.5.1. Relation Extraction based on Gold Standard Dataset

In order to benchmark our approach in the first experiment, we employed the dataset[5] proposed in Angeli et al. (2014) described in Table 5.1. The content of this dataset is comprised of articles from New York Times where each sentence has been annotated with entity tuples and relation types, which are linked to entities from Freebase. This dataset also consists of gold standard relation instances. Note that, we do not use the dataset from (Riedel et al., 2013) given the fact that it does not include the original sentences, which prevents us from being able to identify grammatical clauses or learn the topic models as required in our approach.

5.5.1.1. Evaluation results

In our work, we conducted experiments on both individual models and interpolated models for predicting relations as listed in Tables 5.2 and 5.3. We randomly split the dataset for training and testing and applied 10-fold cross validation for all models. We have applied the threshold 0.5 as suggested in (Yao, 2015) for all models that indicate the confidence value to predict a relation. Table 5.2 shows the detailed performance of each model as well as the combined models in Table 5.3. As observed in Table 5.2, using clause features shows improved performance compared to when models are built without clause information. Using the clause information, we can see the EC model with F-measure of 41.81% is better than the *E* model with F-measure of 38.77%; *N* model obtained only 36% in F-measure while *NC* obtained 39.5% in F-measure. Regarding *T* model where their argument slots are presented in high dimensions, this could take benefits of selectional preference in the training process due to their co-occurrence direct/indirect with other relations. As a result when applying *T* recall and F-measure have greatly improved, e.g., the *E*+*T* model with F-measure of 45.65% is better than the *E* model with F-measure of 38.77%; or *F* model obtained 46.83% in F-measure while *F*+*T* obtained 47.82% in F-measure. We observe that, *N* models are lower than the other models due to weak co-occurrence with other relations. The

---

[5] http://nlp.stanford.edu/software/mimlre-2014-07-17-data.tar.gz

interpolation of *N*, *F*, *E* and *T* models outperforms the non-interpolated models, indicating the synergistic contribution of each of these, e.g., *F*+*E*+*N* (being the baseline presented by Riedel et al. (2013) and *F*+*E*+*N*+*T* models have an F-measure of 51.9% and 52.23%, respectively.

Table 5.2. Effectiveness of *T* and *C* on individual models; *EC*, *NC*, *TC* are individual models with clauses; *N*+*T*, *E*+*T*, *F*+*T* are models with *T*.

Models	Precision (%)	Recall (%)	F-measure (%)
E	48.23	32.41	38.77
EC	51.97	37.02	41.81
E+T	52.20	40.56	45.65
N	44.61	30.18	36.00
NC	48.94	33.11	39.50
N+T	52.47	38.45	44.37
T	46.79	41.70	44.10
TC	54.71	37.02	44.16
F	**58.02**	39.26	46.83
F+T	47.25	**48.45**	**47.82**

Table 5.3. Experimental results for interpolated models.

Models	Precision (%)	Recall (%)	F-measure (%)
Baseline (F+E+N)	**79.58**	38.51	51.90
F+E+N+T	51.16	53.30	52.21
EC+NC	72.29	32.51	43.88
TC+NC	64.12	34.98	47.82
EC+ TC	59.58	39.67	47.62
F+EC	54.65	42.36	47.69
F+NC	56.24	40.14	46.85
F+TC	53.02	46.87	49.75
NC+EC+TC	57.24	42.36	48.69
F+EC+NC	57.31	49.24	52.96
F+NC+TC	55.01	54.80	54.90
F+EC+NC+TC	60.23	**60.00**	**60.11**

The results of interpolated models *EC*+*NC*, *EC*+*TC*, and *EC*+*TC*+*NC* show that each of the models provide advantage in a non-overlapping aspect of the data and hence their interpolation leads to improved performance. *EC*+*NC* achieves an F-measure of 43.88%, *EC*+*TC* has an F-measure of 47.62% and *EC*+*TC*+*NC* produces an F-measure of 48.69%. Therefore, the interpolated models obtain better results compared to the individual *EC*, *NC*, or *TC* models. We note that *TC* employs features based on the presentation of argument slots from entities; and the presentation of argument slots in the *TC* model results in a much higher number of co-occurrences compared to the *EC* model. Therefore, the interpolated models with *TC* achieve better results compared to the interpolated models with EC, e.g., *TC*+*NC* yielded 47.82% while *EC*+*NC* yielded 43.88%.

It is important to point out why the interpolation of *F* and *C* has not been built and included in Table 5.2. The main reason for this is that the *F* model is a matrix factorization model where clause information cannot be directly incorporated into it. On the other hand, *C* is a feature that is not developed in isolation and is only meaningful when used in tandem with other features. Therefore, once a model based on feature *F* has been developed, it is not possible to interpolate it with *C* because *C* does not have a separate standalone model on its own.

The interpolated models with *F* such as *F*+*TC*, *F*+*NC*+*TC* and *F*+*EC*+*NC*+*TC* have features, which are derived based on PCA components (*F* model). Therefore, *F*+*TC*, *F*+*NC*+*TC* and *F*+*EC*+*NC*+*TC* achieve better results compared to the interpolated models without *F* such as *TC*, *NC*+*TC*, and *EC*+*NC*+*TC*. For instance, *NC*+*TC* obtains an F-measure of 47.82% while *F*+*NC*+*TC* obtains 54.90%. Finally, the best interpolated model is *F*+*EC*+*NC*+*TC* which produces the highest result with 60.11% in F-measure when compared to the other models. Our interpolated models, namely *F*+*NC*+*TC*, *F*+*E*+*N*+*T* and *F*+*EC*+*NC*+*TC* outperform the baseline (*F*+*E*+*N*) proposed by Riedel et al. (2013).

Finally, we would like to summarize the impact of our proposed work on performance. As seen in Table 5.3, when employing clause types on the baseline (*F*+*E*+*N* vs. *F*+*EC*+*NC*), we see that recall increases and overall the incorporation of clause type improves F-measure. Also when adding topics to the baseline (*F*+*E*+*N* vs. *F*+*E*+*N*+*T*), we see a similar trend. The important observation is that once clause types and topic models are added simultaneously (*F*+*EC*+*NC*+*TC*) that we achieve a significant improvement on recall and a reasonable precision performance, leading to much higher F-measure. This shows that clause types and semantic topics can help identify a higher number of relevant relations and hence increase retrieval rates and also maintain acceptable precision.

Table 5.4. Top and additional relation samples. Bold values indicate best performing model.

Top relation samples	Org /country_of_headquarter			person/founded			org /city_of_headquarters			Person /country_of_birth			org/member_of		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
F+E+N (Baseline)	**79.87**	63.08	70.49	**76.24**	30.61	43.68	76.78	54.76	63.93	**77.42**	81.96	**79.63**	80.19	77.26	78.70
F+EC+NC	75.58	67.75	71.45	75.31	55.01	63.57	**77.24**	56.36	65.17	70.14	82.12	75.65	**81.55**	77.83	**79.64**
F+TC+NC	62.06	78.10	69.16	69.96	**76.30**	72.99	72.03	**78.23**	75.00	68.56	78.63	72.25	66.00	78.22	71.59
F+EC+TC+NC	75.55	**79.34**	**77.39**	75.74	75.84	**75.84**	74.98	78.13	**76.52**	74.00	**80.02**	76.88	63.07	**79.20**	70.22
Additional relation samples	person/parents			org/shareholders			org_political /religious_affiliation			Person /spouse			person /school_attended		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
F+E+N (Baseline)	50.25	41.75	45.56	**69.61**	30.35	48.46	**64.87**	42.27	55.47	**63.70**	49.22	55.53	60.50	25.75	36.13
F+EC+NC	**56.54**	54.89	55.70	67.76	35.19	46.32	56.94	57.42	**57.18**	61.19	55.20	58.04	**69.40**	31.46	43.29
F+TC+NC	44.67	**75.78**	56.20	62.72	38.79	47.93	44.46	**77.11**	56.40	53.83	**76.78**	63.28	52.08	**42.33**	**46.70**
F+EC+TC+NC	53.12	62.88	**57.58**	55.38	**48.41**	**51.66**	51.16	69.09	56.73	62.63	76.62	**68.92**	53.77	38.14	44.62

Table 5.4 shows several sample relation types, which are broken down into top and sample relation types. The top relations are based on pooled results in the matrix models which provide the basis for meta-analysis (Bravata et al., 2001). We present three models where *T* and *C* show strong influence compared to the baseline model (*F*+*E*+*N*), namely *F*+*EC*+*NC*, *F*+*TC*+*NC* and *F*+*EC*+*TC*+*NC*. In most of the models, recalls and F-measures have greatly improved. While models with E and N have used references for predicting relations on entities, their limitations is in that they are trained for relationships between specific entity tuples and relation types, and therefore, are limited when an insufficient number of explicit evidence is present for each relation type for specific entity tuples. Regarding models with T and C, the relations take advantage of selectional preference in the training process due to their co-occurrence and/or clause type similarity with other relations. These models exploit advanced features from relation characteristics such as clause types and semantic topics for predicting new relation instances. They help in predicting any tuple of entities and relations regardless of whether they were seen at training time with direct or indirect access in their provenance. These results show that the employment of *T* and *C* lead to improved recall and hence better performance over F-measure, which has been the objective of our work.

5.5.1.2. Discussion

Now, in terms of the performance of the individual models, we observe that the *E* and *T* models outperform the *N* model. The *E* and *T* models employ the presentation of argument slots while *N* employs co-occurrence with neighbors. The *N* model might face situations where only a few co-occurrences with other neighbor relations are observed that can cause weak evidence in the training process for learning hidden relations. However, in the *T* and *E* models where their argument slots are presented in high dimensions, this could increase the number of desirable co-occurrences. Most of the models have increased performance when applying clause type features because the clause type information can reduce noise in the training process.

Interpolated models benefit from the advantages of each individual model. Thus, most of the interpolated models achieve better results compared to their constituting separate models. Comparing our best models (*F*+*NC*+*TC*) and (*F*+*EC*+*NC*+*TC*) with Riedel et al.'s model (*F*+*E*+*N*) as a baseline, the results reveal that we obtained 55.01% of precision and 54.80% of recall in *F*+*NC*+*TC*, and 60.23% of precision and 60% of recall in *F+EC+NC+TC* while Riedel et al. achieved 79.58% of precision and 38.51% of recall. Applying topic models to the models could reduce precision but increase recall significantly when compared to the baseline. Baseline+Topic model (*F+E+N+T*) achieves 51.16% of precision and 53.30% of recall. Our model obtained an improvement in recall when compared to the baseline. However, our models also show lower precision because applying topic-based features in our models will lead to an increasingly higher number of hidden relations for prediction compared to the baseline. This can cause a lower precision in our model even when our model predicts more hidden relations compared to Riedel et al.'s model.

5.5.2. Relation Extraction based on Silver Standard Datasets

The second set of experiments is focused on two unlabeled datasets from New York Times and Wikipedia. In order to be able to use these two datasets, we used the relation patterns provided by the PATTY dataset[6] to automatically extract relation instances from these two datasets, which we refer to as the silver standard. As described in Table 5.5, the two datasets consist of 5,827 sentences of 300 relation types from Wikipedia and 7,388 sentences of 300 relation types from NYTimes. Note that, the NYTimes dataset used here is different from the dataset used in the previous experiment. Now, unlike the first experiment where the gold standard was randomly split into test and train sets, we do not use the silver standard relations extracted using the PATTY patterns in the *training* process. Instead, we use several Open IE systems to extract relation instances from the two datasets that would then form the training set and will be used for initializing the matrix model. More specifically, we employ four Open IE systems to extract relation instances for building universal schemas for the matrix model:

- **ReVerb (Fader et al., 2011)**. The system extracts verb phrase-based relations based on a set of syntactic and lexical constraints to identify relations based on verb phrases and then finds a pair of arguments for each identified relation phrase.
- **OLLIE (Masam et al., 2012)**. The system, an extension of the ReVerb system, uses various heuristics to obtain propositions from dependency parsers. OLLIE performs deep analysis on the identified verb-phrase relations and then extracts all relations mediated by verbs, nouns, and adjectives, among others.

---

[6] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/patty/

- **ClausIE (Corro et al., 2013)**. This system exploits linguistic knowledge about the grammar of the English language to first detect clauses in an input sentence and to subsequently identify each clause type based on the grammatical function of its constituents.
- **LS3RyIE (Vo & Bagheri, 2018)**. The system extends the work by ClausIE with grammatical structure reformulations that help identify discrete relations that are not found in ClausIE and reduce the number of erroneous relation extractions. LS3RyIE has been presented in Chapter 3.

Table 5.5. Wikipedia and NYTimes datasets[7].

	Relation types	Extracted sentences	Raw documents
Wikipedia	300	5,827	55,000
NYTimes	300	7,388	58,860

Table 5.6 presents the characteristics of four Open IE systems. These systems use different forms of linguistic analysis such as syntactic analysis, dependency analysis and grammatical clause analysis. Please note that ClausIE and LS3RyIE systems are evaluated with both when they considered grammatical clause structures as well as when they did not.

Table 5.6. Characteristics of OpenIE systems.

OpenIEs	Syntactic analysis	Dependency analysis	Clause analysis
ReVerb	+	-	-
OLLIE	+	+	-
ClausIE	-	+	+
LS3RyIE	+	+	+

Table 5.7. Accuracy of four OpenIE systems tested against PATTY relation patterns.

	Citation	Wikipedia	NYTimes
ReVerb	(Fader et al., 2011)	19.12 %	18.34 %
OLLIE	(Masam et al., 2012)	29.34 %	27.76 %
ClausIE	(Corro et al., 2013)	41.67 %	43.01 %
LS3RyIE	(Vo & Bagheri, 2017)	44.46 %	46.44 %

5.5.2.1. Results

Table 5.7 depicts the performance of four Open IE systems on the two datasets when compared to the relation instances that are included in the silver standard

---

[7] https://bitbucket.org/thuanvd/matrixfac-data/downloads

produced by the PATTY relation patterns. As expected, the results reveal that clause-based Open IE systems such as LS3RyIE and ClausIE have a better performance compared to ReVerb and OLLIE. Regarding Wikipedia and NYTimes, LS3RyIE produced 44.46% and 46.66%; ClausIE 41.67% and 43.01%; OLLIE 29.34% and 27.76% and ReVerb 19.12% and 18.34% in term of accuracy, respectively. Once the relation instances were derived from the raw sentences from the two datasets by the Open IE systems, we employed the relation instances to initialize the matrix model and perform our proposed matrix completion task, the results of which would then be compared with the relations in the silver standard dataset.

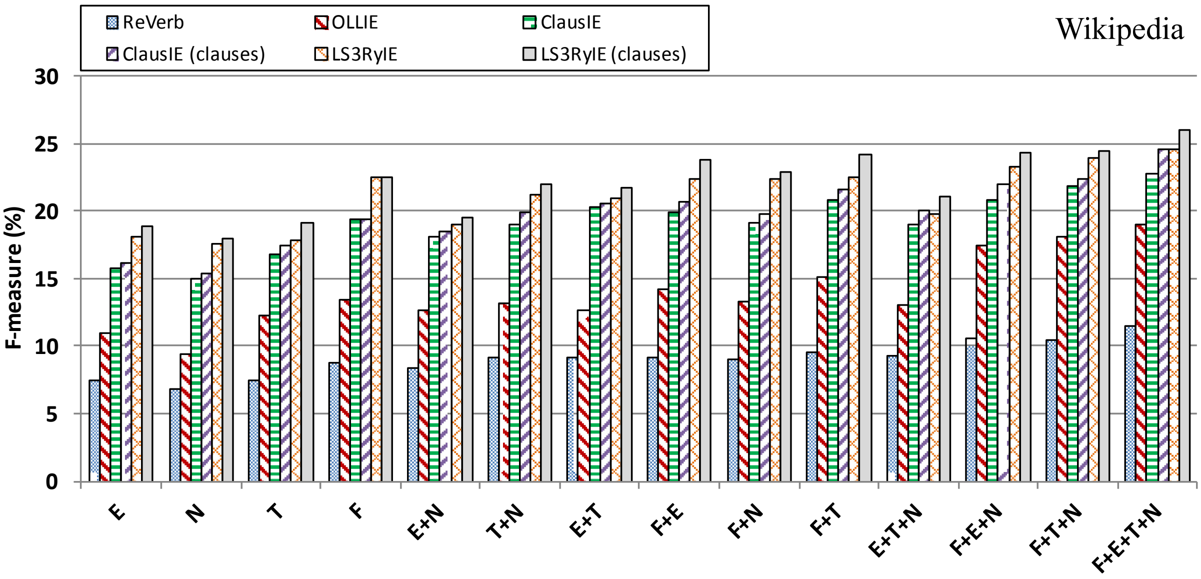
Table 5.8. F+E+N vs. F+E+N+T on the Wikipedia dataset.

OpenIEs	F+E+N (Baseline)			F+E+N+T		
	P	R	F1	P	R	F1
ReVerb	13.47	8.79	10.62	12.16	10.74	11.46
OLLIE	19.49	15.71	17.40	19.52	18.65	19.01
ClausIE	21.43	20.34	20.87	20.24	25.86	22.71
ClausIE(clauses)	**22.04**	21.79	21.91	**21.81**	29.08	24.03
LS3RyIE	21.77	25.07	23.31	20.73	30.22	24.59
LS3RyIE(clauses)	21.95	**27.25**	**24.32**	21.00	**33.23**	**25.47**

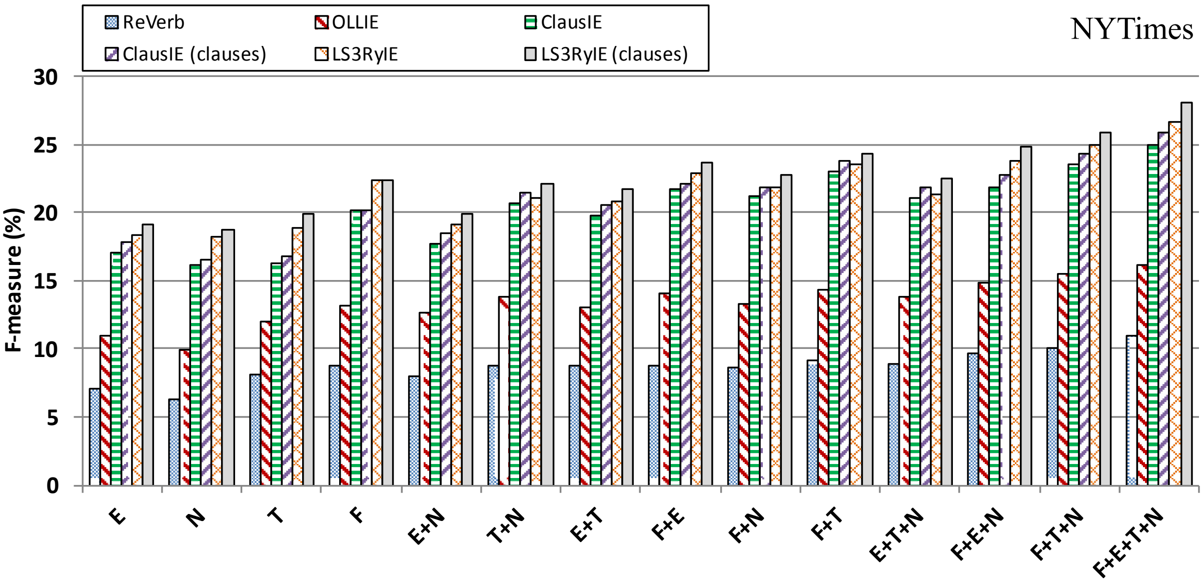
Table 5.9. F+E+N vs. F+E+N+T on the NYTimes dataset.

OpenIEs	F+E+N (Baseline)			F+E+N+T		
	P	R	F1	P	R	F1
ReVerb	11.87	8.17	9.71	11.30	10.73	11.02
OLLIE	17.12	13.66	15.20	15.49	16.98	16.21
ClausIE	**23.72**	20.23	21.84	**25.12**	24.85	24.98
ClausIE (clauses)	21.89	23.69	22.75	23.08	29.51	25.90
LS3RyIE	23.59	23.98	23.78	24.39	29.36	26.65
LS3RyIE (clauses)	22.83	**27.32**	**24.88**	24.08	**33.54**	**28.64**

Similar to the first set of experiments, we conducted the evaluations on both individual and interpolated models for predicting relations with surface schemas extracted from the four Open IE systems. Figures 5.3.a and 5.3.b show the performance of each model using the four Open IE systems for Wikipedia and NYTimes datasets. Similar to the discussion in Section 5.5.1, most of the interpolated models yield better results compared to the individual models. Models *E*+*N*, *E*+*T*, and *E*+*T*+*N* benefit from additional aspects of the data compared to individual models *E*, *N* and *T* and take advantage of presentation from entities, topic models and related neighbors when referencing the argument slot presentation of the matrix. For the Wikipedia dataset, the best performance was observed using the *F+EC+TC+NC* model especially for the LS3RyIE system, which incorporates the use of grammatical clauses.



(a)



(b)

Figure 5.3. The performance of different features based on various Open IE systems. Wikipedia (a) and NYTimes (b).

It is easy to see that the models using clause-based Open IE systems such as LS3RyIE and ClausIE yielded the best performance compared to syntactic-based or dependency-based OpenIE systems (ReVerb or OLLIE). Particularly, the system model *F+E+T+N* obtained 26.03% and 24.56% of F-measure on LS3RyIE when clause information were considered, denoted as *LS3RyIE (clauses)*[8], and *ClausIE (clauses)*, referring to ClausIE when clause information were taken into account. However, when clause information was not considered, LS3RyIE obtained 24.59% and ClausIE obtained 22.71% in terms of F-measure. Regarding ReVerb and OLLIE, a performance of 19.01% and 11.49% on F-measure was observed, respectively. In the F model and based on LS3RyIE and ClausIE without the consideration of clause information, a performance of 22.84% and 19.46% on F-measure was observed, respectively while using OLLIE and ReVerb, we only obtained 12.60% and 8.79% on F-measure. With regards to the NYTimes dataset, similar to Wikipedia, performance of the F+EC+TC+NC model obtained the highest results compared to other models. Particularly, using LS3RyIE (clauses) and ClausIE (clauses), we obtained the highest results with 28.05% and26.11% while we only obtained a performance of 26.65% and 24.98% with LS3RyIE and ClausIE, respectively, when clause information was not considered.

In comparison, we discuss the performance of the model where *T* and *C* showed strong effectiveness compared with the baseline model (*F*+*E*+*N*) proposed by Riedel et al. (2013). Table 5.8 presents the performance of the three models including *F*+*E*+*N* and *F*+*E*+*T*+*N* using four Open IE systems on the Wikipedia dataset. The performance of the *F*+*E*+*N*+*T* model is better than the other models across all Open IE systems. We obtained 11.16%, 19.01%, 22.71%, 24.03%, 24.59%, 25.47% on F-measures in the *F*+*E*+*N*+*T* model using ReVerb, OLLIE, ClausIE, ClausIE (clauses), LS3RyIE and LS3RyIE (clauses), respectively. In contrast, the baseline achieved 10.62%, 17.40%, 20.87%, 23.31% on F-measures using ReVerb, OLLIE, ClausIE and LS3RyIE, respectively. The topic model-based feature presented in the T model leads to better results than the baseline when interpolated with other features. Regarding the performance over the NYTimes dataset, shown in Table 5.9, the baseline succeeded in producing 9.71%, 15.20%, 21.84% and 23.78% on F-measure by using ReVerb, OLLIE, ClausIE, and LS3RyIE, respectively. However, the *F*+*E*+*N*+*T* model proposed in this chapter produced more accurate results with an improved F-measure of 11.02%, 16.21%, 24.98%, 25.90%, 26.65%, and 28.64% using ReVerb, OLLIE, ClausIE,

---

[8] It should be noted that in Figure 3 as well as Tables 9-12, E, N and T are in fact EC, NC and TC when corresponding to the ClausIE (clauses) and LS3RYIE (clauses) rows, because these variations consider clause information. ClausIE and LS3RYIE when mentioned without '(clauses)' do not consider clause information and hence represent all models without interpolation with the C features.

ClausIE (clauses), LS3RyIE and LS3RyIE (clauses), respectively. Thus, it is showing that the proposed model based on the features introduced in this chapter provide meaningful improvements over the baseline especially on recall, which translates to an improved overall F-measure. The improvement in recall is one of the main objectives of our work.

5.5.2.2. Discussion

Similar to the gold standard dataset, in order to perform more in-depth analysis of our work, Tables 5.10-5.11 shows several specific relation types in three models based on pooled results where T and C show strong influence on the various models including the baseline model in all four OpenIE systems. We report top and random relation types in both Wikipedia and NYTimes corpora where top relations are based on pooled results. These models take advantage of the *E*, T and N features in the training process due to co-occurrence information with other relations. Note that, *E*, *T*, and *N* models are considered as *EC*, *TC* and *NC* in ClausIE (clauses) and LS3RyIE (clauses) systems. Now, in terms of the performance of those relation types, in most of the cases, *F*+*E*+*T*+*N* outperform the *F*+*E*+*N* and *F*+*T*+*N* models as it takes advantage of the presentation of argument slots based on *E* and *T* and co-occurrence with neighbors based on *N*. The combination of *F*, *E*, *T* and *N* outperforms all other models showing the synergistic contribution of each of these features. Relations can benefit from rich co-occurrences with other neighboring relations that can provide strong evidence in the training process for learning hidden relations. Moreover, given their argument slots are presented in a higher dimensional space, this could increase the number of desirable co-occurrences for referencing other related relations, e.g., *F*+*E*+*N* vs. *F*+*T*+*N* vs. *F*+*E*+*T*+*N* in ClausIE and LS3RyIE in most of the cases of the relation samples. As such, incorporating more data based on topics within *F*+*E*+*T*+*N* improves the performance compared to *F*+*E*+*N* on both Wikipedia and NYTimes as it allows semantic topic information to determine relation type similarity. Finally, it should be noted that models that use clause information such as ClausIE (clauses) and LS3RyIE (clauses) show improved performance compared to their counterparts that do not use clause information; hence, pointing to the effectiveness of grammatical clause information for improving recall while maintaining precision.

Table 5.10. Top and additional relation samples on the Wikipedia. Bold values indicate best performing model.

Top relation samples		organization/employer			person/country			government/location			educational_institution/location			person/organization_member		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	50.70	39.00	44.08	53.58	33.78	41.43	51.57	30.07	37.98	45.92	32.29	37.91	54.35	31.75	40.08
	F+T+N	60.02	44.73	51.89	61.34	44.73	51.73	35.81	48.76	41.00	49.83	24.03	32.42	35.81	48.76	41.29
	F+E+T+N	45.03	61.96	52.15	36.29	48.66	41.57	29.41	48.67	36.66	27.23	54.86	36.39	32.45	45.45	37.86
OLLIE	F+E+N	56.16	50.25	53.04	60.40	35.83	44.97	42.02	46.62	44.42	50.73	37.54	43.14	54.28	38.03	44.72
	F+T+N	53.21	54.28	53.73	48.64	58.47	53.10	46.99	47.46	47.22	59.28	37.56	45.98	45.70	57.37	50.87
	F+E+T+N	49.24	73.03	58.82	45.12	74.90	56.31	42.07	54.90	47.63	41.68	52.82	46.59	46.21	67.53	54.87
ClausIE	F+E+N	61.12	63.02	62.05	64.13	60.61	62.32	50.06	57.45	53.50	54.36	45.27	49.40	63.32	57.28	60.14
	F+T+N	62.78	58.23	60.41	60.82	65.96	63.28	48.63	59.40	53.47	43.09	58.48	49.61	60.40	60.78	60.58
	F+E+T+N	60.14	65.13	62.53	64.8	70.00	67.29	50.30	64.24	56.42	54.29	57.38	55.69	60.14	65.13	62.53
ClausIE (clauses)	F+E+N	55.96	58.93	57.41	55.78	64.46	59.81	**70.57**	41.09	51.96	61.11	43.47	50.59	64.46	55.78	59.80
	F+T+N	47.43	71.91	57.16	51.93	74.71	62.76	51.88	66.11	58.12	47.18	57.62	51.55	66.79	67.36	**67.16**
	F+E+T+N	57.35	68.86	62.24	67.55	71.24	**69.35**	55.03	64.14	59.23	51.07	68.49	58.99	56.35	69.53	62.25
LS3RyIE	F+E+N	75.04	54.05	62.83	64.96	54.68	59.37	60.97	57.59	59.23	**68.45**	55.61	61.36	68.71	51.69	58.90
	F+T+N	66.06	82.71	73.43	61.37	71.69	66.12	54.81	78.59	64.59	65.44	80.00	71.99	56.92	65.08	60.72
	F+E+T+N	65.20	**80.90**	72.20	59.24	70.03	64.18	60.14	61.15	60.64	63.05	79.26	70.22	60.00	60.60	60.02
LS3RyIE (clauses)	F+E+N	**77.53**	61.32	68.48	**87.03**	51.99	65.09	55.57	73.89	63.43	70.34	61.69	65.73	**70.06**	57.62	63.23
	F+T+N	76.21	76.93	76.60	63.40	**76.08**	69.16	55.98	**81.67**	**66.43**	68.26	78.60	73.06	64.38	67.47	64.42
	F+E+T+N	76.78	77.78	**77.25**	66.79	67.35	67.06	56.75	80.00	66.39	66.06	**84.00**	**73.95**	55.67	**83.80**	66.20
Additional relation samples		tv_actor/program			project/location			employer/location			govermental_jurisdiction/country			business_operation/citytown		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	35.32	23.96	28.55	30.54	12.73	17.97	29.49	24.84	26.96	29.35	15.99	20.70	39.37	19.65	26.22
	F+T+N	22.26	47.49	30.37	11.66	38.33	17.88	23.18	28.23	25.46	13.50	45.87	20.86	26.49	21.17	23.53
	F+E+T+N	36.30	28.31	31.8	17.77	27.66	21.63	24.33	40.13	30.29	24.76	25.10	24.92	35.15	22.88	27.71
OLLIE	F+E+N	35.79	38.55	37.11	**30.56**	13.72	18.93	40.12	29.27	33.84	32.98	16.07	21.61	40.95	26.97	32.52
	F+T+N	44.22	36.97	40.27	13.48	36.50	19.68	35.58	42.71	38.82	18.85	29.69	23.06	32.06	38.74	35.09
	F+E+T+N	41.19	36.70	38.81	19.70	26.63	22.64	29.01	45.62	35.47	27.14	27.07	27.10	38.47	29.25	33.23
ClausIE	F+E+N	59.49	30.32	40.16	19.70	31.90	24.35	49.14	25.41	33.50	**37.80**	19.46	25.69	32.32	41.34	36.28
	F+T+N	43.68	42.20	42.92	16.82	46.70	24.73	46.14	32.04	37.81	18.88	43.95	26.41	31.78	47.10	37.95
	F+E+T+N	50.90	42.07	46.07	18.51	**52.89**	27.42	47.29	31.96	38.14	19.63	**49.24**	28.07	31.89	51.08	39.26
ClausIE (clauses)	F+E+N	61.11	43.57	50.87	17.76	38.11	24.22	**50.57**	29.97	37.64	41.66	19.40	26.47	45.66	31.36	37.18
	F+T+N	38.00	52.32	44.02	17.64	46.72	25.61	29.05	62.70	39.70	28.64	24.21	26.24	30.34	54.38	38.95
	F+E+T+N	46.61	49.19	47.62	19.90	38.43	26.22	36.19	50.62	42.20	36.24	30.21	**32.95**	34.87	52.64	41.95
LS3RyIE	F+E+N	**63.82**	38.18	47.77	28.99	21.49	24.68	53.03	26.93	35.72	37.38	21.53	27.32	54.00	24.98	34.15
	F+T+N	54.32	51.43	52.84	26.35	28.81	27.15	31.01	50.00	40.41	27.83	44.56	34.26	36.88	53.28	43.58
	F+E+T+N	41.34	59.38	48.74	25.09	28.16	26.53	32.45	52.65	40.15	25.20	44.41	32.15	33.04	**59.61**	42.51
LS3RyIE (clauses)	F+E+N	44.20	50.08	46.95	28.46	25.10	26.74	42.33	38.88	40.53	36.07	23.90	28.75	**60.57**	29.19	39.40
	F+T+N	47.60	**70.91**	56.96	25.19	32.58	**28.41**	33.60	**58.88**	42.78	33.75	25.75	29.21	53.55	34.17	41.71
	F+E+T+N	52.61	62.42	**57.10**	20.10	38.72	26.46	37.23	54.58	**44.26**	28.48	34.24	31.10	52.61	37.90	**44.06**

Table 5.11. Top and additional relation samples on the NYTimes. Bold values indicate best performing model.

Top relation samples		country/sport_team_location			actor/film			location/employer			country/organization_member			organization/location		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	53.30	59.67	56.30	53.16	32.48	40.32	61.40	31.22	41.39	61.63	29.03	39.46	41.27	41.42	41.34
	F+T+N	45.81	75.78	57.10	27.30	74.56	39.96	35.04	50.00	41.20	30.67	76.07	43.71	29.11	62.54	39.72
	F+E+T+N	47.42	76.92	58.67	30.38	76.10	43.42	27.95	75.40	40.78	34.02	62.38	44.02	40.07	43.84	41.87
OLLIE	F+E+N	50.70	40.39	44.08	53.58	33.78	41.43	54.35	31.75	40.08	51.57	30.07	37.98	45.92	32.29	37.91
	F+T+N	60.02	45.70	51.89	61.34	44.73	51.73	35.81	48.76	41.29	39.63	42.48	41.00	24.03	49.83	32.42
	F+E+T+N	45.03	61.96	52.15	36.29	48.66	41.57	37.64	38.10	37.86	29.41	48.67	36.66	27.23	54.83	36.39
ClausIE	F+E+N	73.76	68.10	70.81	68.60	61.67	64.95	65.61	62.84	64.19	76.33	48.05	58.97	53.13	61.30	56.92
	F+T+N	63.00	77.73	69.59	61.71	69.08	65.18	54.08	74.23	62.57	56.19	75.11	64.28	53.39	68.57	60.03
	F+E+T+N	70.25	**83.56**	76.34	71.21	73.47	**72.32**	64.11	74.04	69.28	54.33	82.67	65.56	57.87	72.63	64.41
ClausIE (clauses)	F+E+N	75.93	72.09	73.96	76.87	68.98	72.71	75.02	66.79	70.66	**79.80**	60.22	68.66	69.13	56.96	62.45
	F+T+N	74.04	74.04	74.57	61.15	74.84	67.30	61.20	69.73	65.18	53.78	82.25	65.02	52.83	**84.11**	64.89
	F+E+T+N	81.24	81.31	81.12	72.37	73.96	73.15	65.03	81.21	72.22	61.94	**82.45**	**70.66**	60.36	65.56	62.85
LS3RyIE	F+E+N	67.61	75.21	71.20	72.27	62.65	67.11	69.95	64.91	67.33	75.19	52.88	62.09	61.81	60.12	60.95
	F+T+N	69.94	69.96	69.95	72.75	62.61	67.30	65.08	69.87	67.39	56.70	75.38	64.71	53.85	69.33	60.61
	F+E+T+N	77.03	80.19	78.56	69.17	72.65	70.86	69.96	75.50	72.62	64.08	76.25	69.63	64.00	67.87	65.87
LS3RyIE (clauses)	F+E+N	72.33	71.61	71.96	**74.96**	59.96	66.62	**82.71**	59.44	69.17	74.08	57.44	64.70	**75.38**	53.62	62.66
	F+T+N	67.75	76.01	71.64	61.59	**79.34**	69.34	63.08	**80.33**	70.66	61.37	76.52	68.11	67.10	68.03	67.56
	F+E+T+N	**83.57**	83.40	**83.48**	66.59	77.17	71.49	72.77	78.77	**75.65**	68.49	71.24	69.83	61.17	79.26	**69.05**
Additional relation samples		award_winner/employer			kingdom/country			person/alternate_name			olympic_participating_country/location			country/statistical_region		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ReVerb	F+E+N	18.43	18.83	18.37	12.98	33.18	18.66	45.92	32.29	37.92	32.42	25.23	28.37	12.73	30.54	17.96
	F+T+N	14.64	24.95	18.45	11.68	46.27	18.65	24.03	49.83	32.42	25.49	37.82	30.45	11.66	38.33	17.88
	F+E+T+N	17.77	27.66	21.64	14.05	51.65	22.09	27.23	54.86	36.40	31.27	38.67	34.57	13.36	48.38	20.94
OLLIE	F+E+N	22.63	23.96	23.27	24.17	29.00	26.36	**54.63**	42.62	47.88	51.67	50.23	50.93	26.39	28.60	27.45
	F+T+N	16.10	32.42	21.51	17.77	39.83	24.57	41.19	54.98	47.09	51.60	53.74	52.65	20.38	36.23	26.08
	F+E+T+N	22.49	29.83	25.64	27.68	31.18	29.33	46.06	48.52	47.26	48.61	57.46	52.67	30.86	36.08	33.27
ClausIE	F+E+N	16.01	42.92	23.32	28.48	19.35	23.04	38.13	49.26	42.98	**67.85**	42.51	52.27	46.94	19.33	27.38
	F+T+N	15.73	**53.08**	24.26	21.83	27.01	24.15	37.06	56.86	44.87	54.58	56.90	55.71	34.02	25.74	29.31
	F+E+T+N	25.01	37.27	29.93	22.62	39.89	28.86	50.61	52.82	51.69	50.90	70.05	58.96	36.51	28.52	32.02
ClausIE (clauses)	F+E+N	19.83	38.01	26.06	26.51	25.06	25.76	41.37	47.87	44.38	53.30	59.67	56.30	34.95	26.01	29.82
	F+T+N	23.08	41.72	29.71	23.05	38.59	28.86	44.47	54.13	48.82	51.41	60.78	55.70	25.28	58.33	35.27
	F+E+T+N	**29.75**	37.45	33.15	26.39	**39.47**	31.63	42.32	69.03	52.47	48.72	71.21	57.85	35.90	38.78	37.28
LS3RyIE	F+E+N	21.48	50.00	30.05	42.22	20.97	28.02	34.02	62.38	44.03	52.16	50.00	51.05	50.00	35.04	41.20
	F+T+N	21.60	51.01	30.34	35.51	27.26	30.84	34.90	65.96	45.65	50.00	62.76	55.65	28.07	61.50	38.54
	F+E+T+N	29.48	44.04	**35.31**	32.25	30.00	31.08	35.58	**79.05**	48.97	63.06	64.48	**63.76**	30.30	**69.67**	41.94
LS3RyIE (clauses)	F+E+N	27.03	22.12	24.33	**44.52**	19.36	26.98	44.52	53.62	48.65	49.61	57.80	53.40	**51.84**	36.71	43.00
	F+T+N	21.29	50.71	29.98	32.70	25.54	28.68	42.28	61.31	50.47	47.41	70.37	56.65	42.23	52.08	**46.70**
	F+E+T+N	29.20	38.55	33.60	39.55	29.20	**33.60**	51.88	63.59	**57.14**	54.63	**73.89**	62.81	47.06	41.57	44.14

Table 5.12. Time complexity.

Time complexity	Cost
Stochastic Gradient Descent	$O(mnk)$
F Model	$\mathrm{O}(nm^2+n^3)$
N Model	$O(mni)$; $O(mn)$; $O(mn^2)$
E Model	$O(mnj)$; $O(mn)$; $O(m^2n)$
T Model	$O(mnh)$; $O(mn)$; $O(m^2n)$

Table 5.13. The execution time breakdown (in minutes) for the matrix models.

Models	ReVerb		OLLIE		ClausIE (clauses)		LS3RyIE (clause)	
	Wiki	NYTimes	Wiki	NYTimes	Wiki	NYTimes	Wiki	NYTimes
E	39.4	44.5	41.4	46.8	44.5(47.8)	52.8(59.3)	48.4(53.1)	53.7(62.4)
N	37.2	40.5	35.8	42.4	38.0(41.3)	45.2(49.1)	41.3(45.9)	46.0(51.7)
T	42.5	48.7	43.1	47.7	46.7(49.7)	54.4(60.4)	51.9(55.2)	56.6(63.6)
F	49.1	58.7	55.7	60.3	64.2	69.7	65.6	71.2
Mean	42.1	48.1	44.0	49.3	48.3(50.8)	55.5(59.6)	51.8(55.0)	56.9(62.2)
E+N	85.2	94.7	97.6	113.3	98.6(104.9)	117.2(131.2)	107.2(116.5)	128.6(139.5)
T+N	92.3	105.5	92.0	124.4	100.8(106.1)	132.2(145.3)	112.0(117.9)	135.3(154.2)
E+T	89.5	107.6	90.2	122.2	95.7(104.0)	135.7(142.5)	104.6(115.6)	140.7(151.3)
F+E	95.4	119.2	113.2	131.6	103.7(112.7)	149.8(154.3)	113.9(125.2)	155.3(163.8)
F+N	94.6	116.4	102.5	127.8	110.0(118.2)	146.9(149.6)	124.8(131.4)	145.0(158.8)
F+T	97.5	122.3	105.7	131.9	128.6(135.4)	145.4(156.5)	130.1(135.5)	156.2(164.2)
Mean	92.4	111.0	100.2	125.2	106.2(113.6)	137.8(146.6)	115.4(123.7)	145.0(155.3)
E+T+N	102.4	126.5	103.2	116.9	114.3(119.1)	137.0(145.5)	123.0(127.3)	146.5(150.8)
F+E+N	108.4	128.6	114.5	125.6	128.2(132.1)	141.6(157.8)	133.6(141.8)	154.1(162.5)
F+N+T	113.9	128.9	120.4	148.9	134.2(141.3)	149.8(162.4)	136.7(150.2)	159.3(165.9)
Mean	108.2	128.0	112.7	130.4	125.6(130.8)	135.5(155.2)	131.5(139.8)	153.3(159.7)
F+E+T+N	115.7	131.5	122.1	132.3	129.9(137.2)	153.7(161.8)	136.5(143.7)	162.8(166.4)

5.5.3. Execution Time

We have measured the execution time of the different models when we use them to build universal schemas for the matrix model. It should be noted that our work in this chapter works with patterns that are extracted by OpenIE systems at the sentence level where the result of each sentence is independent from other sentences. Therefore, it is possible to easily distribute the processing of the system and immensely scale it as required. Regarding the execution time performance of the matrix models, our work requires time for executing the matrix factorization process. We ran our system on a server computer with 24 cores, 62GB RAM and 10TB hard disk. For the time complexity in the models with m rows and n columns, PCA has $O(nm^2+n^3)$ with covariance matrix computation of $O(nm^2)$ and eigen-value decomposition computation of $O(n^3)$ (Johnstone et al., 2009). The N, E, T models perform a process to calculate cells in the matrix with their references. N model takes $O(mni)$ with i references $(0 \leq i \leq n)$ while E and T models take $O(mnj)$ and $O(mnh)$ with j, h references $(0 \leq j \leq m;\ j \leq h \leq m)$, respectively. Moreover, the process of Stochastic Gradient Descent takes $O(mnk)$ with k iterations (Zhang, 2004). The details of time complexity are as shown in Table 5.12. The best time complexity of N, E, T models are $O(mn)$ when their references are only one. When the references of N, E, T Models are equal with m or n as well, the worst time complexity of N, E, T models are $O(mn^2)$, $O(m^2n)$ and $O(m^2n)$, respectively.

Table 5.13 shows the details of the execution time performance of our work for individual models and interpolated models on both Wikipedia and NYTimes corpora. In most of the cases, the execution time performance of each model is based on how many dimensions the matrix model has. Matrices with higher dimensions require more time than lower dimensional matrices. In individual models, the execution time of the N model is the fastest while the F model takes the most amount of time, e.g., the execution of the N model takes 37.2m while the execution of the F model takes 49.1m when performed using ReVerb. For the interpolated models, there is less difference in terms of execution time between different interpolated models as each individual model can be executed in parallel and only interpolated when the results of all individual models are available, e.g., the mean execution time of the interpolated F+E+T+N model takes 153.7m (161.8m) and 162.8m (166.4m) based on ClausIE (clauses) and LS3RyIE (clauses) while the interpolated model F+E requires around 137m (146.6m) and 145m (155.3m) on the same NYTimes corpus. Based on our experiments, we conclude that it is possible to easily distribute the different models required by our work and hence scale it to large-scale relation extraction scenarios.

5.5.4. Summary of Findings

The objective of our work in this chapter has been to explore how relation extraction based on a matrix completion approach can be performed in such a way that high precision relations can be extracted while many relations are retrieved, i.e., a high recall rate is maintained. We introduced features such as those based on statistical topic models as well as grammatical clause structure, which, theoretically-speaking, had the potential to improve recall. Through our extensive experiments, we have made the following observations:

1. The interpolation of grammatical clause structure information with other features improves both recall and precision as shown in Table 5.2.

2. The interpolation of statistical topic models with other features significantly improves recall rates at the cost of precision as shown in Table 5.3. However, it should be noted that the overall F-measure metric is improved noticeably.
3. The interpolation of any of the base models with either clause structure and/or statistical topic models consistently improves recall and hence leads to improved F-measure.

As such, we find that the work proposed in this chapter is able to address its objective, which was to improve the overall performance of the relation extraction process as well as address the limitation of the earlier work that was related to a low recall. We have shown that our proposed approach improves recall and f-measure while maintaining a reasonable precision.

5.6. Concluding Remarks

In this chapter, we have presented a framework for predicting potential relation instances based on feature enrichments applied to matrix models that are used in a matrix completion process. We have exploited universal schemas that are formed as a collection of patterns from Open IE systems and relation schemas from pre-existing datasets to build a matrix model in order to use matrix factorization and collaborative filtering to predict relations. While previous systems have trained relations only for entities, we further exploited advanced features such as clause types and statistical topic models for predicting implicit relation instances. Particularly, we exploited clause-based features extracted from Open IE systems combined with topic models for predicting potentially relevant relation instances. We have carried out extensive experiments on both gold and silver standard datasets. The results of these experiments show that features based on grammatical clause patterns and statistical topic models are able to increase the recall of the relation extraction task while maintaining a reasonable precision, hence leading to an improved overall performance over F-measure when compared to the baseline.

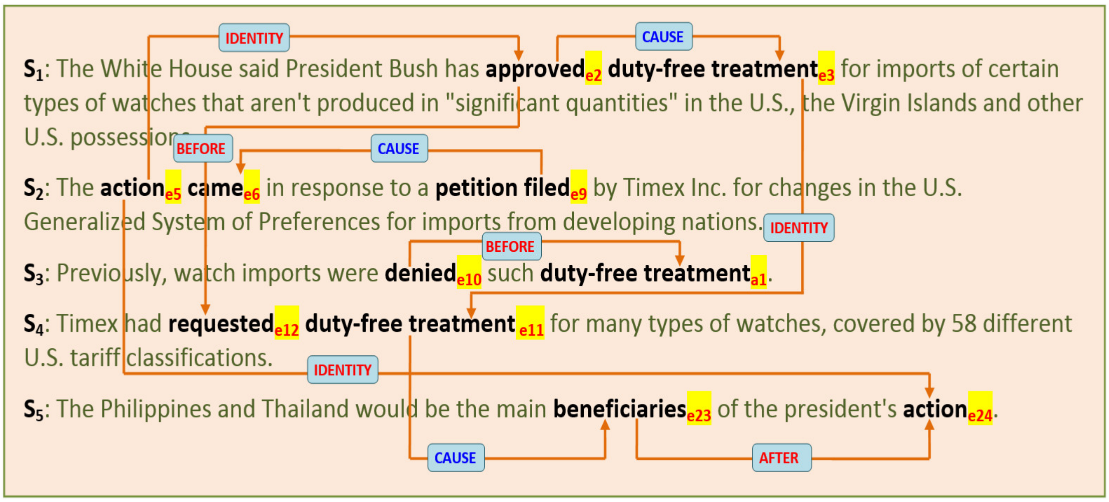
Chapter 6

Extracting temporal and causal event relation based on event network

Event relations specify how different event flows expressed within the context of a textual passage relate to each other in terms of temporal and causal sequences. In this chapter, we present event network method to temporal and causal relation extraction that operates at the document level. More specifically, we benefit from existing Open IE systems to generate a set of triple relations that are then used to build an event network. The event network is bootstrapped by labeling the temporal disposition of events that are directly linked to each other. We then systematically traverse the event network to identify the temporal and causal relations between indirectly connected events.

6.1. Introduction

Learning temporal and causal relationships between events that have been mentioned in a textual passage is an important task in information extraction towards deeper language understanding (Qian et al., 2019; Zhang et al, 2018; Navarro-Colorado et al., 2016, Ye et al., 2019, Fan et al., 2019). Temporal and causal relations can happen both between events within the same sentence and between events across sentences in a document. Understanding relations between events in a document is beneficial to various Natural Language Processing applications such as question answering (Abacha et al., 2016; Ji et al., 2013; Qian et al., 2019), document summarization (Ji et al., 2013; Verhagen et al., 2010, Kim at al., 2018) and textual entailment (UzZaman et al., 2013; Verhagen et al., 2010). Moreover, identifying the causal and temporal relation between events is an important step in predicting occurrence of future events, and can be beneficial in risk analysis as well as proactive decision making (Hogenboom et al., 2016).



S$_1$: The White House said President Bush has **approved**$_{e2}$ **duty-free treatment**$_{e3}$ for imports of certain types of watches that aren't produced in "significant quantities" in the U.S., the Virgin Islands and other U.S. possessions.

S$_2$: The **action**$_{e5}$ **came**$_{e6}$ in response to a **petition filed**$_{e9}$ by Timex Inc. for changes in the U.S. Generalized System of Preferences for imports from developing nations.

S$_3$: Previously, watch imports were **denied**$_{e10}$ such **duty-free treatment**$_{a1}$.

S$_4$: Timex had **requested**$_{e12}$ **duty-free treatment**$_{e11}$ for many types of watches, covered by 58 different U.S. tariff classifications.

S$_5$: The Philippines and Thailand would be the main **beneficiaries**$_{e23}$ of the president's **action**$_{e24}$.

Figure 6.1. Samples of direct relation events and indirect event relations in textual document.

An increasing amount of recent work has focused on recognizing temporal and causal event relations within a document, but mostly limited to identifying intra sentences causal relations with explicit causal indicators. Mirza & Tonelli (2014a) and Laokulrat et al. (2015) have reported that incorporating temporal information can improve the performance of a causal relation classifier. These authors have developed and annotated a causal corpus (Causal-TempBank) based on the TempEval-3 corpus (Mirza and Tonelli, 2014a) for the task of causal event relation. Mirza and Tonelli (2016a) built both a rule-based multi-sieve approach and a feature-based classifier to recognize causal relations in Causal-TimeBank. These approaches to temporal and causal relation classification use machine-learning-based classifiers (Laokulrat et al., 2013; Laokulrat et al., 2015; Mirza et al., 2014; Miraza et al., 2016a) that are trained based on a predefined, finite and fixed schema of relation types. The common strategy of these techniques is to generate linguistic features based on syntactic, dependency, or shallow semantic structures of the text. Based on these features, supervised learning methods are used to identify pairs of events that are related to each other and can be classified into pre-defined relation types. However, the state-of-the-art approaches (Laokulrat et al., 2015; Mirza et al., 2016a) suffer from two key drawbacks. First, they are focused on a limited subset of features, which might not, in many cases, be present in every sentence or be sparsely available. Second, training on linguistic structures such as the output of syntactic and dependency parsers does not necessarily identify all possible types of event relations when they are presented in different sentences or different documents.

One of the main challenges for achieving comprehensive temporal and causal relations is that such event relations are sparse among all the event pairs in a document and few causal event relations are explicitly stated. The challenge is especially true for identifying cross-sentence causal event relations and most of them have no clear causal indicators. For instance, considering the three sentences shown Figure 6.1, the events in both <e_2-e_3>, <e_5-e_8> are related to each other by the "BEFORE" temporal relation type. Here, while e_2 and e_3 are present in the same sentence, events e_5 and e_8 are in different sentences. As such, sentences that rely on features based on grammatical parsers can fail to identify correct relation types. We observe that events occur in action flows that happen throughout the documents. Therefore, for improving event relation identification, we consider an event network representation that involves the connections between event flows of a document and can capture cross-sentence event relations in a document. Our event network captures structural distributions of events within a document, for performing comprehensive event identification for event relation extraction. In addition to the global event structures related to the structure of events in a document, we model event networks of fine-grained event structures in order to accurately identify each individual relation.

In this chapter, our objective is to address these challenges by adopting an Open Information Extraction (Open IE) approach (Corro et al., 2013; Vo & Bagheri, 2018), which is able to extract relations and their arguments without the need to restrict the search to predefined relation types, in order to build the event network for event relation extraction. Particularly, we consider and incorporate all identified Open IE patterns that consist of at least one event instance in the event network, which is then systematically traversed for identifying temporal relations. As an example in Figure 1, both <e_2-e_3> and <e_5-e_3> relations can be extracted from two Open IE patterns, namely ("*President Bush*", "*has approved*", "*duty-free treatment for imports*") and ("*Timex*", "*had requested*", "*duty-free treatment*"), respectively. The event network is built based on such patterns identified and extracted by Open IE where each triple will be represented as two vertices connected by an edge. The traversal of this event network allows us to identify indirect relations through event coreference relations in the network. Furthermore, we employ a shortest path strategy to determine the event flow between two events based on the constructed event network.

Our most significant contributions are as follows:

- We propose an event network structure primarily based on information from Open IE systems for extracting temporal and causal relations between events. The event network is the basis for systematically exploring the event flow between events by considering how events can be reached from one another.
- We present a method to measure the weight of event flows to detect shortest paths between events. Through the weight, shortest paths are identified that can ultimately lead to higher confidence event relation extractions.
- We present algorithms for temporal and causal relation extraction that make it possible to reason over the set of relations observed in the event flow to make a determination about the relation between two events. Indirect relations are inferred through transitive inferences over direct relations.
- Our approach works at the document level and hence can identify event relations between events that have been expressed across different sentences. This is made possible due to the linking of different events in the proposed event network whose linking transcends individual sentences and forms a representation of events in the document.

The rest of this chapter is organized as follows: section 6.2 offers detailed description of our proposed approach to construct event network for event extraction. Following that, we present the methods used for causal and temporal relation extraction with regards to rule-based sieve and transitivity process in the network in Section 6.4 and 6.5. Section 6.6 offers an in-depth analysis of our experiments for temporal and causal event extraction where the results obtained from our proposed approach are compared to the state-of-the-art systems. In the last section, we draw conclusions about the merits of our work and offer ways to advance the literature in the future.

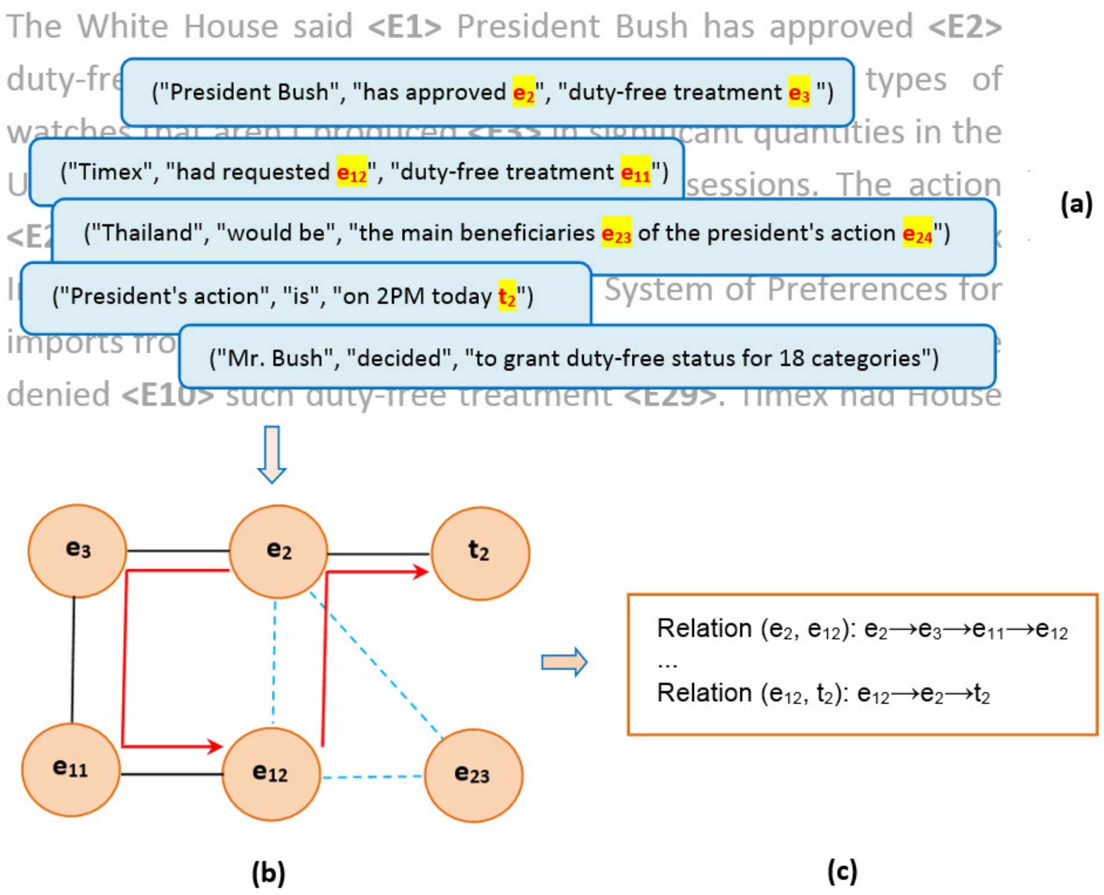
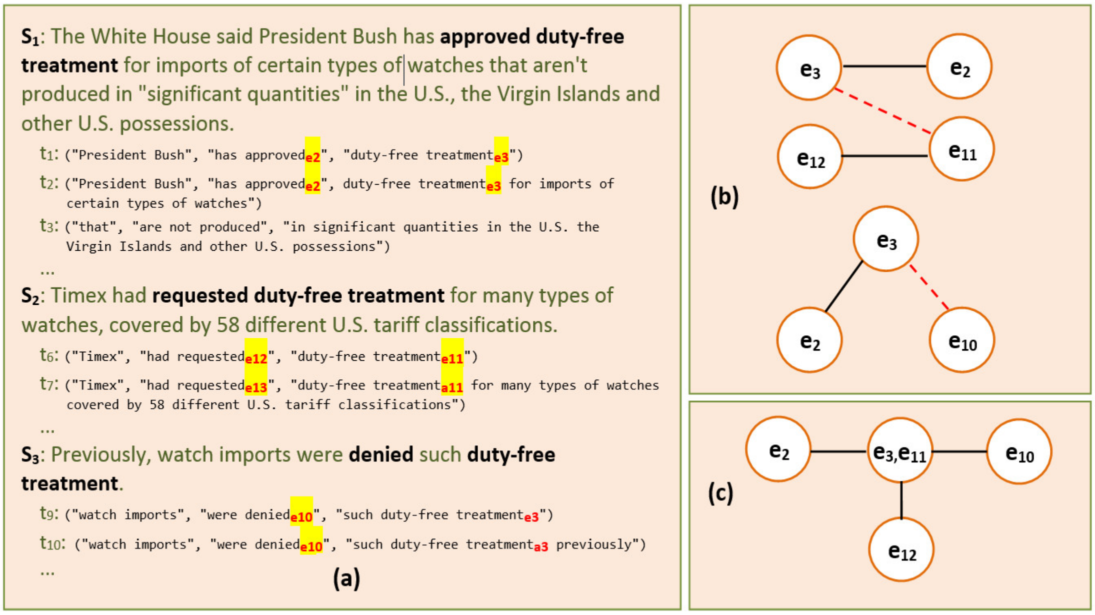

The White House said <E1> President Bush has approved <E2>
("President Bush", "has approved e2", "duty-free treatment e3 ")
("Timex", "had requested e12", "duty-free treatment e11")
("Thailand", "would be", "the main beneficiaries e23 of the president's action e24")
("President's action", "is", "on 2PM today t2")
("Mr. Bush", "decided", "to grant duty-free status for 18 categories")
(a)
e3
e2
t2
e11
e12
e23
(b)
Relation (e2, e12): e2→e3→e11→e12
...
Relation (e12, t2): e12→e2→t2
(c)

Figure 6.2. An overview of the proposed approach with (a) Open IE extraction, (b) event network construction, and (c) event flow extraction.

6.2. Overview of our Proposed Approach

Traditional methods for event relation extraction are primarily developed at the sentence level (Zhou et al., 2010; Mirza et al., 2016a) and hence do capture longer event relations that span over multiple sentences or across a document. In order to address this issue, we build our work on Open IE systems that are able to automatically extract triples in the form of (arg1, rel, arg2) representing basic propositions or assertions from text. The main benefit of using Open IE systems is that they are not limited to a sentence level restriction and can operate over a document. As such, we intend to address the limitation of earlier work by exploiting the characteristics of Open IE systems for identifying event relationships. To this end, we benefit from Open IE to build an event network that can capture temporal and causal event relations across different sentences at the document level. The proposed event network aims at identifying the temporal and causal event relations in a document, both within a sentence with direct links and across sentences with indirect links. Our method captures event relations from event flows by detecting paths in the event network. We present algorithms to analyze event flows in the proposed event network that will be used to identify temporal and causal event relations. An overview of our proposed approach is illustrated in Figure 6.2. As shown in the figure, we first exploit an Open IE system to extract relation triples in a document in order to build an event network (Figure 6.2a). For each triple extracted by the Open IE system, each of the two entities will form a node in the event network and an edge will be connecting the two nodes to each other (Figure 6.2b). Once the network is fully built, we identify potential temporal or causal relations between two nodes of the event network by performing specific forms of traversal on the event network (Figure 6.2c).



S_1: The White House said President Bush has **approved duty-free treatment** for imports of certain types of watches that aren't produced in "significant quantities" in the U.S., the Virgin Islands and other U.S. possessions.

t_1: ("President Bush", "has approved$_{e2}$", "duty-free treatment$_{e3}$")

t_2: ("President Bush", "has approved$_{e2}$", duty-free treatment$_{e3}$ for imports of certain types of watches")

t_3: ("that", "are not produced", "in significant quantities in the U.S. the Virgin Islands and other U.S. possessions")

…

S_2: Timex had **requested duty-free treatment** for many types of watches, covered by 58 different U.S. tariff classifications.

t_6: ("Timex", "had requested$_{e12}$", "duty-free treatment$_{e11}$")

t_7: ("Timex", "had requested$_{e13}$", "duty-free treatment$_{a11}$ for many types of watches covered by 58 different U.S. tariff classifications")

…

S_3: Previously, watch imports were **denied** such **duty-free treatment**.

t_9: ("watch imports", "were denied$_{e10}$", "such duty-free treatment$_{e3}$")

t_{10}: ("watch imports", "were denied$_{e10}$", "such duty-free treatment$_{a3}$ previously")

…

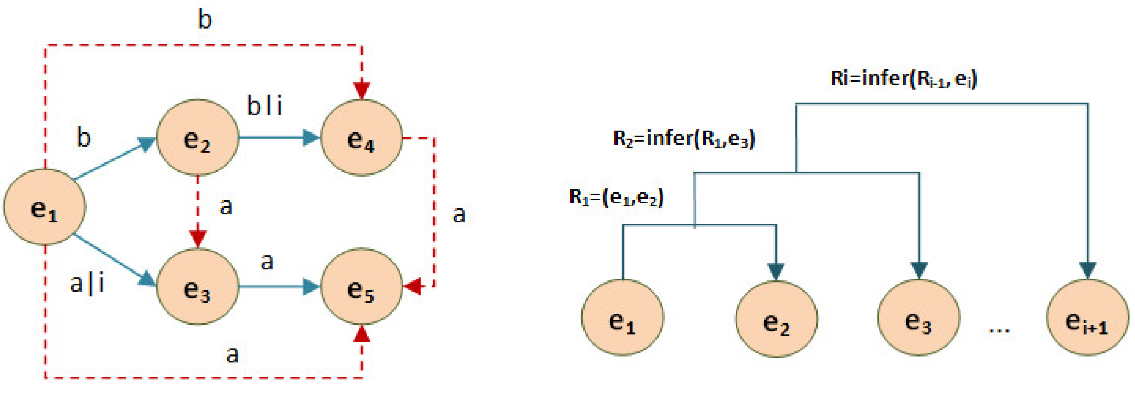
Figure 6.3. Building a sample event network with events e_2, e_3, e_{11}, e_{12} and e_{10}. (a) Open IE relation triples; (b) constructing the event network with matching arguments; (c) finalizing the construction of the event network after matching.

6.3. Graph-based event network

Our proposed graph-based event network is built directly from triples generated by Open IE systems. Two events that are present in the same extracted pattern are considered as two event nodes in the event network that are directly connected to each other with an edge. The collections of all the extracted Open IE triple patterns are used to complete the event network. Let us denote an event network as a graph $G(V, E)$ where each vertex denotes an event and each edge denotes a relation. Based on the event network, the objective is to determine the existence and type of relation between two event nodes such as (*X*, *Y*) expressed as *R*(*X*, *Y*). In the event network, identifying temporal and causal event relations will involve the consideration of event flows in the form of network paths. For example, given a network with set of relation nodes (e_2, e_3), (e_3, e_{11}), (e_{11}, e_{12}), (e_2, t_1) shown in Figure 6.2b, in order to determine the relation (e_2, e_{12}), we need to identify the network path ($e_2 \rightarrow e_3 \rightarrow e_{11} \rightarrow \mathrm{e}_{12}$) between e_2 and e_{12}. It can be observed that the relations between (e_2, e_3), (e_3, e_{11}) and (e_{11}, e_{12}) are informative indicators for predicting the relation (e_2, e_{12}). Based on such a network path and if we know that event e_2 began concurrently with event e_3, and e_{12} started before the occurrence of event e_{11} and e_3, then we can determine that event e_2 happened after e_{12}.

Let us first cover the process through which we build the event network. The basic idea behind building the event network is that it operates based on a set of Open IE triples and a set of events representing the arguments of these triples. Summarily, we process each triple produced by the Open IE system by creating a matching vertex for each of the triple's arguments in the event network. The two vertices corresponding to the triple arguments would then be linked together through an edge in the event network. It should be noted that a *matching* method is used in order to ensure that events already identified in previous triples do not represent new vertices and are matched to existing vertices in the event network, e.g. events e_3 and e_{11} shown in Figures 3a and 3b are matched with each other as they both reference "duty-free treatment". After processing each individual triple and creating corresponding vertices and edges, as well as ensuring that similar vertices are folded into one to avoid duplicates, the resulting graph forms our intended event network.

Now given this event network and by considering the shortest path between each pair of nodes in the network (Olya et al., 2014), it is possible to infer the type of the event relation between source and target nodes based on the set of event relations observed on the shortest path. Indirect event relations can be inferred through transitivity of relations (Allen, 1984; Laokulrat et al., 2015) on direct relations as shown in Figure 6.4. For instance, consider events e_1 and e_4 in Figure 6.4a. Given the fact that the shortest path between events e_1 and e_4 passes through e_2, it is possible to infer that because event e_1 happened before e_2 and e_2 occurred prior to event e_4 that e_1 also happened before e_4.



(a) (b)

Figure 6.4. Inferred relations; a) Inferred sample relations in three nodes path (e_1-e_4, e_2-e_3, e_1-e_5, e_4-e_5) with b: before; i: includes; a: after; b) Recursively inferred relations.

```
Input:   Graph $G = (V, E)$
         Pair of events  $\{X, Y\}$
Output:  Type of relation $R(X, Y)$

if Direct$(X, Y) \in G$ then
    $R(X, Y) \leftarrow$ Rules(E$\{X, Y\}$, $G$)
    Update($G$)
else if Indirect$(X, Y) \in G$ then
        R$(X, Y) \leftarrow$ Time$\{X,Y\} \oplus$Tense$\{X,Y\}$
        Update($G$)
    if R$(X, Y)$=NULL then
            Event-flow$\{X,Y\} \leftarrow$ Shortest-path($\{X, Y\}$, $G$))
            Rules(Event-flow$\{X,Y\}$, $G$)
            $R(X, Y) \leftarrow$Infer(Event-flow$\{X,$Y$\}$)
            Update($G$)
    end if
return $R(X, Y)$
```

Algorithm 6.1. The outline of the process for identifying temporal relations.

6.4. Temporal Event Relation Extraction

According to the TempEval-3 task description (Mirza et al., 2016a and Laokulrat et al., 2015), two pairs of temporal events can be related to each other through one of four groups, namely Timex-Timex, Event-DCT, Event-Timex and Event-Event where DCT denotes *Document Creation Time* and Timex denotes *Temporal Expressions*. A pair of temporal entities, including events and temporal expressions, annotated as a temporal relation, is called a TLINK. Temporal relation recognition is the task of classifying

TLINKs into temporal relation types. It uses a complete set of TLINK relations, which consists of 13 types of temporal relations shown in Table 6.1.

In this chapter, we present a method to extract both direct and indirect temporal event relations in the event network. The objective is to determine the existence and type of temporal relation between two event nodes such as (*X*, *Y*) expressed as *R*(*X*, *Y*) in the event network. Algorithm 6.1 presents the pseudo-code for our proposed algorithm for identifying temporal relations. The basic idea behind our algorithm is that it takes a Graph G(*V*, *E*) and a pair of events (*X*, *Y*) and subsequently determines relation R(*X*, *Y*). Our algorithm distinguishes between relations whose nodes are directly connected to each other in the network, denoted by Direct(*X*, *Y*), and those events whose nodes are not directly connected, shown as Indirect(*X*, *Y*). If the relation is concerned with two directly connected events, we employ a rule-based model to determine the type of the relationship.

Table 6.1. Temporal relation types[9].

Relation	Description
BEFORE	One before the other
AFTER	One after the other
BEGINS	One being the beginning of the other
BEGUN_BY	One being begun by the other
DURING	One holding during the duration of the other
END_BY	One being ended by the other
ENDS	One being the ending of the other
IBEFORE	One immediately before the other
IAFTER	One immediately after the other
IDENTITY	Referring to the same event
INCLUDES	One including the other
IS_INCLUDES	One being included in the other
SIMULTANEOUS	One is simultaneous with the other

Determining temporal relation types for direct relations

Inspired by the work in Chambers et al. (2014) and Mirza et al. (2016a), the temporal rule-based model relies on hand-crafted rules. In the rule-based model, we separate temporal constructs containing temporal features of event relations. Some of the relations between the events are captured based on morpho-syntactic information from their textual expression. Several relations are based on semantic information such as typical event duration while other relations are computed independently based on

---

[9] http://www.timeml.org

dependency paths. We also include rules for predicting relations between reported events (Timex and DCT events) and other events based on their time characteristics and syntactically captured tense. We introduce four categories of rules as follows:

Timex-Timex (T-T). For Timex-Timex relations, we take temporal expressions such as DATE, TIME and DURATION into consideration and then identify the relation types based on their normalized values. For example, the day after last Monday will be normalized into 15-Jan-2019 if last Monday was 16-Jan-2019.

Event-DCT (E-DCT). To predict relation types of E-D event pairs the rules are based on tense and aspect of entity events. For instance, the event entity "will attack$_E$", which is in the future tense, its relation between E-DCT will be recognized as "AFTER". Hence, we have separate rules as follows:

if E ∈ Tense(PAST) ***then*** label(BEFORE)

if E ∈ Tense(PRESENT) and PROGRESSIVE(E, D) ***then*** label(INCLUDES)

if E ∈ Tense(FUTURE) ***then*** label(AFTER)

Event-Timex (E-T). The rules built for Event-Time exploit the senses of prepositions (Litkowski and Hargraves, 2006; Litkowski, 2014). We consider extracting the dependency path between an event *e* and a timex *t* via modifiers. Six types of time prepositions are defined for constructing the rules such as STARTTIME (e.g., from, since), ENDTIME (e.g., until), DURATION (e.g., during), FOLLOWTIME (e.g., after), PRECEDINGTIME (e.g., before), and POINTTIME (e.g., in, on, at). For instance, when *t* is presented in the POINTTIME form and its dependency modifier is `compound,` the event relation will be recognized as 'IS INCLUDED'. These rules are defined as follows:

if DP NMOD(E, T) ∈ STARTTIME ***then*** label(BEGUN_BY)

if DP NMOD(E, T) ∈ ENDTIME ***then*** label(END_BY)

if DP NMOD(E, T) ∈ DURATIONTIME ***then*** label(DURING)

if DP NMOD(E, T) ∈ FOLLOWTIME ***then*** label(AFTER)

if DP NMOD(E, T) ∈ PRECEDINGTIME ***then*** label(BEFORE)

if DP NMOD(E, T) ∈ POINTTIME ***then*** label(IS_INCLUDED)

Event-Event (E-E). The rules for event pairs (e_1, e_2) are constructed in two sets based on dependency paths and the verbs surrounding the events. We extract the dependency path between event e_1 and event e_2 to build the first set of rules. For the second set, the rules are built based on the tense aspects of events e_1 and e_2. For instance, if event e_2 is the logical subject of e_1 as in "...the chain reaction touched$_{e1}$ off by the collapse$_{e2}$ of Soviet Union", events e_1 and e_2 are connected by an AFTER relation. The following rules are based on the tense aspect of each event as follows:

if DP LGS-PMOD(E1, E2) and VERB-AUXPASS(E1) ***then*** label(AFTER)

if DP LOC-PMOD(E1, E2) and AVERB-LOC(E1) ***then*** label(IS_INCLUDED)

if DP XCOMP(E1, E2) and PROGRESSIVE(E1) ***then*** label(SIMULTANEOUS)

if DP XCOMP(E1, E2) and VERB-TERMINATION(E1) ***then*** label(ENDS)

if DP XCOMP(E1, E2) and VERB-INITIATION (E1) ***then*** label(BEGIN)

if DP XCOMP(E1, E2) and VERB-CONT(E1) ***then*** label(INCLUDES)

if DP XCOMP(E1, E2) ***then*** label(BEFORE)

Determining temporal relation types for indirect relations

Once the type of relations for the events that are directly connected to each other in the event network are determined based on the introduced rule sets, we determine the relation types for indirect relations based on their event flow. Algorithm 1 proceeds to detect types of each event relations primarily based on transitivity between direct relations in the shortest path between any given two events as shown in Figure 4a. It is possible to reason over the set of temporal relations observed on the shortest path to make a determination about the type of temporal relation between the two source and target events. In our work, indirect relations are inferred through transitivity of temporal relations (Allen, 1983; Laokulrat et al., 2015). The possible transitive inferences for temporal relations are shown in Table 6.2. The table shows how a transitive inference will be made given the temporal relations shown on the columns and rows of the table. For instance, if event *X* occurs BEFORE event *E* and event *E* happens immediately BEFORE event *Y*, we can infer a new temporal relation "*X* happens BEFORE *Y*" based on transitive inference in Table 6.2 (the cell at the intersection of the BEFORE column with the BEFORE row).

Table 6.2. Transitive table of temporal relation types; and UNDEF is tagged as undefined relation.

	BEFORE	AFTER	IBEFORE	IAFTER	IDENTIFY	INCLUDES	IS_INCLUDED	DURING	SIMULTANEOUS	BEGINS	BEGUN_BY	END	END_BY
BEFORE	BEFORE	UNDEF	BEFORE	UNDEF	BEFORE	BEFORE	UNDEF	BEFORE	BEFORE	BEFOR	BEFORE	UNDEF	BEFORE
AFTER	UNDEF	AFTER	UNDEF	AFTER	AFTER	AFTER	UNDEF	UNDEF	AFTER	UNDEF	AFTER	AFTER	AFTER
IBEFORE	BEFORE	UNDEF	BEFORE	UNDEF	IBEFORE	BEFORE	UNDEF	UNDEF	BEFORE	BEGIN	BEGIN	UNDEF	BEFORE
IAFTER	UNDEF	AFTER	UNDEF	AFTER	IAFTER	AFTER	UNDEF	UNDEF	AFTER	UNDEF	AFTER	IAFTER	IAFTER
IDENTIFY	BEFORE	AFTER	IBEFORE	IAFTER	INDENTIFY	INCLUDES	IS_INCLUDED	DURING	SIMULTANEOUS	BEGIN	BEGUN_BY	END	END_BY
INCLUDES	UNDEF	UNDEF	UNDEF	UNDEF	INCLUDES	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF
IS_INCLUDED	BEFORE	AFTER	BEFORE	AFTER	IS_INCLUDED	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF
DURING	BEFORE	AFTER	BEFORE	AFTER	DURING	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF
SIMULTANEOUS	UNDEF	UNDEF	UNDEF	UNDEF	SIMULTANEOUS	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF
BEGINS	BEFORE	AFTER	BEFORE	IAFTER	BEGINS	UNDEF	UNDEF	UNDEF	UNDEF	BEGINS	UNDEF	UNDEF	UNDEF
BEGUN_BY	UNDEF	AFTER	UNDEF	END_BY	BEGUN_BY	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	BEGUN_BY	UNDEF	UNDEF
END	BEFORE	AFTER	IBEFORE	AFTER	END	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	END	UNDEF
END_BY	BEFORE	UNDEF	IBEFORE	UNDEF	END_BY	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF	END_BY

Table 6.3. Dependency path between two events with causal verbs.

Types	Dependency path
Subject(e_1) → Verb	
+ e_1 is subject of v	`SBJ`
+ v is predicative complement of e_1	`PRD-IM`
+ v is modifier of e_1 (nominal)	`NMOD`
+ v is apposition of e_1	`APPO`
+ v is general adverbial of e_1	`ADV`
+ v is adverbial of purpose/reason of e_1	`PRP-IM`
Verb → Object(e_2)	
+ e_2 is object of v	`OBJ`
+ e_2 is logical subject of v (passive verb)	`LGS-PMOD`
+ e_2 is predicative complement of v (raising/control verb)	`OPRD, OPRD-IM`
+ e_2 is general adverbial of v	`ADV-PMOD`
+ e_2 is adverbial of direction of v	`DIR-PMOD`
+ e_2 is modifier of v (adjective or adverbial)	`AMOD-PMOD`
Verb(e_1) → Verb(e_2)	
+ v_2 is predicate of v_1	`XCOMP`

6.5. Causal Event Relation Extraction

Similar to temporal relations, we propose an algorithm for identifying causal relations (CLINK/CLINK-R) based on the proposed event network shown in Algorithm 6.2. The process for identifying causal relations follows the same flow as that of determining temporal relations. We distinguish between direct and indirect relations.

Determining causal relation types for direct relations

In the case of direct relations, we identify terms that belong to affect, link or causative verbs and find specific dependent structures in which such terms are associated with two directly related events. For instance, one could determine a causal relation between a pair of events (e_1, e_2) when there exists a causal verb v, for which e_1 is the subject of verb v, and e_2 is either object of v or complement of v. Such relations between events and causal verbs are usually expressed syntactically; therefore, we define rules for identifying pairs of events being related to a causal verb in a causal construct by looking at their dependency paths. If such dependencies are found, a CLINK/CLINK-R is automatically set between the two events. Table 6.3 shows the details of dependency paths for this purpose. We consider three types of interactive causal verbs between events as Subject → Verb, Verb → Object and Verb → Verb in order to define the rules. For example, in the case of Subject → Verb and Verb → Object, an event e_1 is predicate complement of cause verb v whose object is event e_2 showing dependencies `PRD-IM` and `OBJ`, respectively. For instance, in the sentence "*The assessment*$_{\mathbf{e1}}$ *was to stop*$_{\mathrm{v}}$ *their investment*$_{\mathbf{e2}}$ ...", 'e_1 causes e_2' because `PRD-IM`(assessment, stop) and `OBJ`(stop, investment).

```
Input:   Graph $G = (V, E)$
         Pair of events $\{X, Y\}$
         List of CA Verbs $K$
Output:  Type of relation $R(X, Y)$

if Direct($X, Y$) $\in G$ then
    $R(X, Y) \leftarrow$ Causal-Rules(E$\{X, Y\}$, $G$)
    Update($G$)
else if Indirect($X, Y$) $\in G$ then
        Event-flow$\{X,Y\} \leftarrow$ Shortest-path($\{X, Y\}$, $G$))
        Causal-Rules($K$, Event-flow$\{X,Y\}$, $G$)
        $R(X, Y) \leftarrow$Infer(Event-flow$\{X,$Y$\}$)
        Update($G$)
    end if
return $R(X, Y)$
```

Algorithm 6.2. The process for identifying causal event relations.

Our work differentiates itself from Mirza et al. (2016a), which uses a predefined set of causal verbs that could face limitations when used in an *open* domain. In our work, we define the causal verbs for the rules presented in Table 6.3 by measuring the *causality association* (cause-effect) between a pair of events e_1 and e_2 (Do et al., 2012). In order to calculate cause-affect association, we measure pointwise mutual information (*PMI*) using two separate components between events with verbs, where we denote P_{pp} by measuring the association between event predicates, and P_{pa} by measuring the association between the predicate of an event and the arguments of the other event. Specifically, we measure cause-affect association (*CA*) as follows:

$$CA(e_1, e_2) = P_{pp}(e_1, e_2) + P_{pa}(e_1, e_2) \tag{6.1}$$

where $P_{pp}(e_1, e_2)$ is defined as:

$$P_{pp}(e_1, e_2) = PMI\left(p_{e_1}, p_{e_2}\right) \times IDF\left(p_{e_1}, p_{e_2}\right) \tag{6.2}$$

which takes into account the *PMI* between verbs p_{e_1} and p_{e_2} of event e_1 and e_2, respectively. In Suppes' Probabilistic theory of Causality (Suppes, 1970; Do et al., 2011), the author showed that the first event e_1 is a possible cause of the second event e_2, if e_1 happens more frequently with e_2 than by itself, i.e. $P(e_1) > P(e_2)$. This can be easily rewritten as $P(e_1, e_2)/P(e_1)P(e_2) > 1$, similar to the definition of PMI:

$$PMI(p_{e_1}, p_{e_2}) = \log \frac{P(p_{e_1}, p_{e_2})}{P(p_{e_1})P(p_{e_2})} \tag{6.3}$$

and,

$$IDF(p_{e_1}, p_{e_2}) = \log \frac{T}{1 + N} \tag{6.4}$$

where T is the total triple tuples and N is the number of triple tuples that contains p_{e_1}, p_{e_2}.

We further define $P_{p,a}(e_1, e_2)$ as follows:

$$P_{pa}(e_1, e_2) = \frac{1}{|A_{e_2}|} \sum_{a \in A_{e_2}} PMI(p_{e_1}, a) + \frac{1}{|A_{e_1}|} \sum_{a \in A_{e_1}} PMI(p_{e_2}, a) \tag{6.5}$$

where A_{e_1} and A_{e_2} are the list of arguments from T that contain e_1 and e_2, respectively.

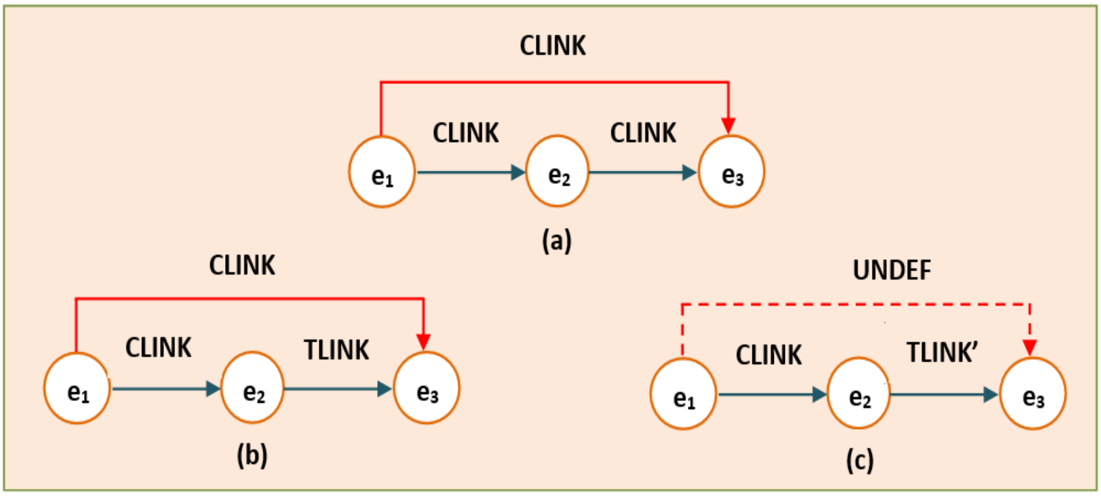


Figure 6.5. Transitivity process of causal relation. a) Causal relation (CLINK) → causal relation (CLINK); b) Causal relation (CLINK) → temporal reverse dependent relation (TLINK); c) Causal relation (CLINK) →temporal dependent relation (TLINK').

Table 6.4. Transitive inference of causal relation types. UNDEF specifies undefined relation.

	CLINK	CLINK-R
Causal link		
CLINK	CLINK	UNDEF
CLINK-R	UNDEF	CLINK-R
Reverse temporal dependent link		
IDENTITY	CLINK	CLINK-R
DURING	CLINK	CLINK-R
INCLUDES	CLINK	CLINK-R
BEGINS	CLINK	CLINK-R
ENDS	CLINK	CLINK-R
SIMULTANEOUS	CLINK	CLINK-R
Temporal dependent link		
IAFTER	UNDEF	UNDEF
IBEFORE	UNDEF	UNDEF
IS_INCLUDED	UNDEF	UNDEF
BEGUN_BY	UNDEF	UNDEF
ENDED_BY	UNDEF	UNDEF

Determining causal relation types for indirect relations

Now for indirect causal relations, on the basis of the transitivity process proposed by Allen (1983), we construct a set of inference relations in order to determine the transitivity of causal relations on both *causal relation → causal relation* (CLINK→CLINK) and *causal relation → temporal relation* (CLINK→TLINK) as shown in Figure 6.5. Regarding the transitive process for CLINK→CLINK (Figure 5a), we base it on the characteristics of causal relations such that if event e_1 causes event e_2 and event e_2 causes event e_3, then e_3 will be considered to depend on e_1. Therefore, a transitive relation will be made between e_1 and e_3 as e_1 causes e_3. Regarding CLINK→TLINK relations, the first part of the relation represents a causal dependency between two events while the second part of the relation expresses a temporal relation. In such cases (Figure 6.5b), similar to CLINK→CLINK, a new transitive causal relation can be generated between events in causal and temporal relations. For instance, when event e_1 causes event e_2 and event e_3 has a temporal relation with event e_2, this could, in the case of *reverse temporal dependent relations*, mean that event e_3 is dependent on event e_1. Consequently, event e_1 will cause event e_3. Similar to Mirza et al., (2016b), we classify these reverse temporal dependent relations where e_3 is dependent on e_2 as IDENTITY, DURING, INCLUDES, BEGINS, ENDS and SIMULTANEOUS. For the case of *temporal dependent relation* in CLINK→TLINK' (Figure 6.5c), although e_2 is dependent on e_1 but e_3 is not dependent on e_2. Therefore, a relation, expressed as "UNDEF", is defined between e_1 and e_3. The list of *temporal dependent relation* where a clear causal relation cannot be inferred from them include IAFTER, IBEFORE, IS_INCLUDED, BEGUN_BY and ENDED_BY. In summary, the details of the transitive inference process is presented in Table 4. We exploit our proposed event network to detect CLINK between e_1 and e_2 and TLINK between e_2 and e_3 for predicting a causal relation between e_1 and e_3 by following the set of transitive inference rules that are listed in Table 6.4, e.g., if e_1 CLINK e_2 and e_2 IDENTITY e_3, then we infer a new causal relation 'e_1 CLINK e_3'.

6.6. Performance evaluation

We have proposed methods to extract event relations based on the proposed event network aiming at two tasks: 1) Temporal event extraction and 2) Causal event extraction. In this section, to carry out evaluations for these tasks, we conduct experiments on several benchmark datasets and compare the performance of our proposed work with strong baseline methods. Particularly, in the first task, we use TempEval-3 corpora. Furthermore, Causal-TimeBank corpora, developed by Mirza et al. (2014a), will be used for the second task. We employ the LS3RyIE system (Vo & Bagheri, 2018), which is an Open IE system, to extract relation instances for constructing the event network. We only consider triples which contain event entities for building the network. Note that, reference mapping was also applied to enhance node matching in the event network. We calculated the context from the Open IE patterns using cosine similarity then merged those nodes with a score $\geqslant 0.5$. We will show how our proposed work can identify temporal and causal relations and reduces the number of erroneous relation types compared to previous baseline approaches.

6.6.1. Temporal event relation extraction

6.6.1.1. Experimental dataset

For benchmarking our approach in this task, we have conducted experiments on the Tempeval-3 corpora task C. TempEval-3 (Pustejovsky et al., 2006) consists of an annotated corpus of temporal relations that was created following the TimeML specification (Sauri et al., 2006). It contains news articles with 183 training documents and 20 testing documents, with just over 61,000 non-punctuation tokens, coming from a variety of news reports, specifically from outlets such as ABC, CNN, PRI, VOA, Wall Street Journal and newswire from AP and NYT. The task of TempEval-3 corpus has been designed by UzZaman et al. (2013) at SemEval-2013, with a complete set of 13 TLINK types. We use the same training and test data released in the context of Tempeval-3 for evaluation. The distribution of the relation types in training and test datasets is shown in Table 6.5.

Table 6.5. Details of the TempEval-3 corpus.

Corpus	Documents	Open IE triples	#relations
Training dataset	183	2,537	2,191
Testing	20	968	1,173

6.6.1.2. Evaluation results

We evaluate our algorithm on both training and testing datasets for predicting event relations because our approach is unsupervised in nature. We first extracted triple patterns from Open IE to construct the proposed event network in each document. We built the event network based on the generated Open IE triples extracted by the LS3RyIE system. As a result of this process, 968 and 2,537 triples were generated by the Open IE system that were then used to build the event network for the testing and training sets, respectively. It should be noted that unlike the baselines that are supervised temporal relation extraction methods, our work is completely unsupervised and as such we do not require separate training and testing datasets. For this reason, we report the performance of our work on the data available in both sets.

Table 6.6 shows the detailed performance of our proposed approach over 13 relation types. Among the relation types, our approach obtains high performance in BEFORE, AFTER, BEGUN_BY, and IS_INCLUDES temporal relation types. The BEFORE relation type yielded 68.65% and 67.76% of F-measures on the two parts of the dataset and the AFTER relation type obtained 67.15% and 65.71% of F-measures in training and testing datasets, respectively. Our method is successful in recognizing event relation types due to the use of transitivity between direct and indirect links in the event network. We observed that several relations are affected or dependent on other relations in the event flow, e.g., it is possible to predict BEFORE($e_1 \rightarrow e_4$) in the path $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4$ when it is presented as BEORE($e_1 \rightarrow e_2$), BEGUN_BY($e_2 \rightarrow e_3$) and

IS_INCLUDED($e_4 \rightarrow e_3$). This explains also the large number of positively predicted temporal relation types in our work.

Table 6.6. Experimental results on four relation types in TempEval-3 corpus.

Categories	Testing set			Training set		
	Precision	Recall	F-measure	Precision	Recall	F-measure
BEFORE	75.74	62.77	68.65	77.54	60.17	67.76
AFTER	76.56	59.80	67.15	74.03	59.07	65.71
BEGINS	62.50	52.63	57.14	50.00	50.00	50.00
BEGUN_BY	75.00	50.00	60.00	80.00	75.50	77.68
DURING	75.00	64.29	69.23	66.66	25.00	36.36
END_BY	36.36	36.36	36.36	50.00	33.33	40.00
ENDS	50.00	45.45	47.62	100.00	66.66	80.00
IBEFORE	83.33	50.00	62.50	66.66	66.66	66.66
IAFTER	85.00	70.00	76.77	70.33	43.25	53.56
IDENTITY	66.67	50.54	57.50	33.33	40.66	36.63
INCLUDES	57.89	49.25	53.22	64.52	53.34	58.40
IS_INCLUDES	72.65	59.44	65.38	83.01	69.39	75.59
SIMULTANEOUS	57.35	48.75	52.70	40.81	38.57	39.66

However, our method obtained lower results in a few relations such as DURING, END_BY, and SIMULTANEOUS. Upon further exploration of these relations, we found that the Open IE system used in our experiments struggles to identify the required triples for these relation types primarily due to the sparseness of the DP link in these events. For instance, when predicting the SIMULTANEOUS relation between e_1 and e_2 in the following sentence "*… refer*$_{\mathbf{e1}}$ *to U.N. resolution 425 in a speech …that would not be attacked, Yitzhak Mordechai said*$_{\mathbf{e2}}$ *at that time*.", the Open IE system could not extract the direct triple (e_1, e_2), ("*a speech*", "*said*") or ("*a speech*", "*time*") because there were no DP links between them. As such, these events were not directly connected to each other in the event network and therefore the relation type was not subsequently identified.

We have also performed a more detailed evaluation by dividing temporal relations into four categories of temporal event types as shown in Table 6. In the testing set, our approach achieved F-measures of 49.49%, 78.58%, 87.86%, and 62.50% for Event-Event (*E-E*), Event-Time (*E-T*), Event-DCT (*E-DCT*), and Time-Time (*T-T*), respectively. On the training set, our work obtained F-measures of 53.04%, 69.47%, 69.68% and 72.91% for Time-Time, Event-DCT, Event-Time and Event-Event, respectively. Particularity, *E-T* yielded the highest results while *E-E* obtained the weakest results compared to the others. This confirms that the structure of the *E-E* relation is more complex than other categories because events in such relations can be expressed in several different relation types when they are in similar grammatical forms. For example, e_1-e_2 is presented in the form of 'IS_INCLUDED' in the sentence "*John used*$_{\mathbf{e1}}$ *the phone when driving*$_{\mathbf{e2}}$ *a car.*" while e_3-e_4 is presented as 'SIMULTANEOUS' in the sentence "*John died*$_{\mathbf{e3}}$ *when bombing*$_{\mathbf{e4}}$ *the hotel.*". In contrast, other event types such as Time can be expressed with fewer language forms such as a limited number of prepositions or time nouns, e.g., time with preposition 'in, at, and on' or time of day in the week or year. Hence, they are easier to be detected.

Table 6.7. Experimental results on four types of temporal event relations.

Categories	Testing set			Training set		
	Precision	Recall	F-measure	Precision	Recall	F-measure
E-E	60.63	41.81	49.49	60.51	47.21	53.04
E-T	84.82	73.21	78.58	82.37	60.06	69.47
E-D	88.24	87.50	87.86	70.63	68.75	69.68
T-T	62.50	62.50	62.50	77.77	68.62	72.91
Overall	72.92	57.54	64.32	69.58	57.18	62.77

In Table 6.8, we compare our method with several strong baseline approaches designed for Task C of TempEval-3. UTTime (Mirza et al., 2014b) employs features based on syntactic parsing including phrase structures while Laokulrat et al. (2013) extracts event relations using time graphs and stacked learning. TRelPro (Laokulrat et al., 2015) and CATENA (Mirza et al., 2016a) employ an SVM classifier based on event linguistic features such as POS tags, chunking, and dependency paths. The numbers reported in Table 6.8 are the results of 5-fold cross-validation evaluation strategy for the baselines because the baselines are *supervised*. The evaluation shows that our proposed method, which is completely unsupervised, shows a competitive performance to the supervised baselines, i.e., better performance on precision and f-measure and competitive on recall. We note that the proposed event network structure is dependent on the performance of the underlying Open IE system and hence in cases when the Open IE system cannot extract event mentions, the corresponding event nodes will not be created in the event network and hence event relations will be missed. The lower recall of our method, noted in Table 6.8, can be explained on this basis. For example, in our experiments, the Open IE system extracted triples in which only one event is present as shown in Figure 6.6. In such cases, it would not be possible to add any edges to the event network given the lack of extracted events. Consequently, our approach will not be able to recognize any additional relations based on such event instances.

S_{10}: He's not telling$_{\mathbf{e14}}$ the truth.
OIE triples: ("He", "is not telling", "the truth")
S_{12}: She was shot$_{\mathbf{e20}}$ with a rifle.
OIE triples: ("She", "was shot with", "a rifle")

Figure 6.6. Sample triples obtained through Open IE.

Table 6.8. Performance comparison.

	Precision	Recall	F-measure
UTTime (2013)	55.60	57.40	56.50
TRelPro (2014)	58.48	58.80	58.17
Laokulrat et al. (2015)	57.60	57.90	57.80
CATENA (2016)	62.60	61.30	61.90
Proposed method	70.82	57.31	63.35

Our approach benefits from the relation patterns extracted by Open IE to build the initial event network and bootstraps the temporal event extraction process by determining the type of temporal relation between two directly linked events. The advantage of our proposed work is two folds: 1) it is completely unsupervised and hence does not require any hand-annotated samples and infers indirect temporal relation types between events by systematically traversing the event network, and 2) it works at the document level and not sentence level and hence can identify temporal relations between events that have not been expressed in the same sentence. This is made possible due to the linking of different events in the network whose linking transcends individual sentences and forms a representation of events in the document.

6.6.2. Causal event relation extraction

6.6.2.1. Experimental dataset

For benchmarking our approach in the second task, we use the Causal-TimeBank and its extended version described in Table 6.9. Causal-TimeBank has been developed by Mirza et al. (2014a), whose annotations for causality are taken from TempEval-3 corpus, containing 183 documents with 6,811 golden events in total. In the Causal-TimeBank dataset, according to Mirza et al. (2014a), a total number of 332 event pairs for CLINK/CLINK-R can be identified from 2,519 events available in the original

TimeBank corpus. Besides, not all documents contain causality relations between events; from the total number of documents in TimeBank, only 110 (around 60%) of them contain explicit causal links. In Table 6.9, we report the statistics of causal relations found in the Causal-TimeBank dataset, along with the corresponding numbers of CLINKs associated with them. Regarding its extension (Causal-TimeBank Ext), different from Mirza et al. (2016a) that only uses cause-effect verbs around temporal events, we expand the dataset with new CLINKs based on combining CLINKs and TLINKs in the event network. We note that all expanded causal relations are based on original CLINKs and TLINKs in Causal-TimeBank, which were annotated using the CAT tool (Lenzi et al., 2012) by two annotators with Dice's coefficient of 0.73. As a result, based on the event network, we annotated an additional 170 causal relation events, which we refer to as Causal-TimeBank Ext. This increased the number of annotated CLINKs by 51.2%. The mean number of CLINKs are 3.02 and 4 in Causal-TimeBank and Causal-TimeBank Ext, respectively. The extended version of Causal-TimeBank has been made publicly available[10].

Table 6.9. Causal-TimeBank and Causal-TimeBank extension corpora.

Corpus	# docs	Causal-TimeBank		Causal-TimeBank Ext	
		#CLINKs	Mean CLINKs	#CLINKs	Mean CLINKs
ABC	3	9	3	16 (77%)	5.33
AP	2	20	10	48 (140%)	24
APW	11	25	2.27	39 (56%)	3.54
CNN	3	3	1	5 (40%)	1.66
EA	2	4	4	5 (25%)	5
NYT	6	22	3.66	30 (36%)	5
PRI	2	3	1.5	4 (33%)	2
SJMN	1	2	2	2 (0%)	2
VOA	2	6	3	6 (0%)	3
WSJ	78	236	3	347 (47%)	4.45
Total	110	332	3.02	502 (51.2%)	4.5

6.6.2.2. Evaluation results

We first extract patterns from Causal-TimeBank and its extension based on the LS3RyIE Open IE system to build the event network. Note that, we consider both causal links (CLINKs) and temporal links (TLINKs) on related causal events. As a result of

---

[10] https://bitbucket.org/thuanvd/EventEx-data/downloads/

this process, 638 and 930 triples were generated by the Open IE system that were then used to build the event network for both Causal-TimeBank and Causal-TimeBank Ext datasets, respectively. Table 6.10 shows the detailed performance of our proposed method on both datasets. In Causal-TimeBank, the system obtained 76.45% and 73.91% on F-measures for WSJ and AP compared to 73.91% on APW. Regarding WSJ, APW and AP in Causal-TimeBank Ext, our approach yielded 71.76%, 70.77% and 56.67% for F-measure, respectively. In both datasets, it should be noted that WSJ and APW account for a high proportion of the total documents while AP has a more complex event network with a higher number of CLINKs. In Causal-TimeBank, WSJ corpus contains 78 documents with mean CLINKs of 3 and APW has 11 documents with mean CLINKs of 2.27. While AP only has 2 documents with mean CLINKs of 25. Regarding Causal-TimeBank Ext, the mean CLINKs of WSJ is 4.45 and APW has a mean CLINKs of 3.5 while AP has a mean CLINKs of 24. In order to show how the performance of our proposed approach compares to the baseline system, known as CATENA (Mirza et al., 2016a), Table 6.11 presents a comparison on both Causal-TimeBank and Causal-TimeBank Ext corpora. CATENA exploits event linguistic features such as POS tags, chunking, and dependency paths to train a causal event relation classifier using SVM. Our method outperforms the baseline in both datasets. Our method yields 76.10% and 71.58% on F-measure while CATENA obtained 62.20% and 58.47% for F-measures in Causal-TimeBank and Causal-TimeBank Ext, repetitively. Different with CATENA, our method is *unsupervised* and operates at the *document level* that traverse the event network to identify the causal relations between indirectly connected events while CATENA is a supervised method that relies on sentence-level textual features to identify causal relations.

S_7: The U.S. maintains that under the U.N. charter, the Kuwaiti request triggers steps$_{\mathbf{e54}}$ for the collective enforcement of international sanction$_{\mathbf{e55}}$.

…

S_9: In a statement, the White House said it would do “Whatever is necessary” to ensure compliance$_{\mathbf{e66}}$ with the sanctions$_{\mathbf{e67}}$.

Output:
e55 → e67 IDENTITY
e54 → e55 CLINK
e67 → e66 CLINK
e54 → e66 CLINK
e54 → e67 CLINK

Figure 6.7. Identified samples based on transitive inference across multiple sentences.

Table 6.10. Experimental result in Causal-TimeBank and Causal-TimeBank Ext in specified corpuses.

Corpus	Causal-TimeBank			Causal-TimeBank Ext		
	Precision	Recall	F-measure	Precision	Recall	F-measure
ABC	77.78	77.78	77.78	84.61	68.75	75.86
AP	71.43	62.50	66.67	73.91	45.95	56.67
APW	77.27	70.83	73.91	76.67	65.71	70.77
CNN	66.67	66.67	66.67	80.00	80.00	80.00
EA	100	100	100	100	100	100
NYT	63.63	63.63	63.63	66.67	63.16	64.87
PRI	66.67	66.67	66.67	75.00	75.00	75.00
SJMN	100	50.00	66.67	100	50.00	66.67
VOA	83.33	83.33	83.33	83.33	83.33	83.33
WSJ	83.88	70.23	76.45	79.71	65.26	71.76

Figure 6.7 shows several extracted relations from a sample document (document wsj900813-0157 from the WSJ corpus). In sentences S_7 and S_9 in this figure, a few temporal and causal event relations can be extracted by using rules based on syntactic and dependency features used in CATENA (Mirza et al., 2016a) such as "e_{55}-IDENTIFY-e_{67}" and "e_{54}-CAUSE-e_{55}", and "e_{67}-CAUSE-e_{66}". However, CATENA fails to correctly determine event relations such as e_{54}-e_{66} and e_{54}-e_{67}. This is primarily due to the fact that events in such relations have no DP connections with each other as they are not present in the same sentence. As such, CATENA cannot extract features pertinent to relations such as (e_{54}, e_{66}) and (e_{54}, e_{67}) and hence fails to identify such relations. In contrast, these relations have been detected in our approach because of the application of transitive inference on the event network, e.g., e_{54}-CAUSE-e_{66} and e_{54}-CAUSE-e_{67} could be determined via e_{54}-CAUSE-e_{55}-IDENTIFY-e_{67} and e_{54}-CAUSE-e_{67}-CAUSE-e_{66}, respectively. As a result, our approach shows up to 13% improvement on F-measure over the baseline.

Table 6.11. Performance comparison in Causal-TimeBank and Causal-TimeBank Ext.

	Precision	Recall	F-measure
Causal-TimeBank			
CATENA	73.70	53.80	62.20
Proposed method	82.47	70.64	76.10
Causal-TimeBank Ext			
CATENA	71.56	48.43	57.76
Proposed method	78.93	65.49	71.58

6.7. Discussions

In our work, we proposed to build an event network based on triples from Open IE systems. We have used LS3RyIE because this system is able to generate high-precision extractions and can be flexibly customized to the underlying application domain compared to other Open IE systems (Vo & Bagheri, 2018). LS3RyIE exploits linguistic knowledge about the grammar of the English language to first detect clauses in an input sentence and to subsequently identify the type of each clause according to the grammatical function of its constituents to extract relations with minimal domain-dependent background knowledge and the least amount of annotated training data. Regarding the overall performance, our system outperforms supervised methods such as CATENA (Mirza et al., 2016a), Laokulrat et al., (2015), TRelPro (Mirza et al., 2014b), and UTime (Laokulrat et al., 2013). These supervised methods exploit grammatical features such as syntactic and dependency features, but are limited to identifying intra sentence relations with explicit temporal and causal indicators. Our approach does not require any hand-annotated samples for inferring indirect temporal and causal relation types between events by systematically traversing the proposed event network. The network is built based on directly observable temporal and causal relations at the document level. It is possible to predict direct and indirect relations by linking different events in the event network whose linking transcends individual sentences and forms a representation of events across sentences. Hence our method can determine event relations both within the same sentence and across sentences in the whole document. We ran our system on a desktop computer with Intel Core i5 3.1 ghz, 16GB RAM and 1TB hard disk. Regarding the process of identifying m target relations in a document, the process of temporal and causal relation identification first finds the shortest path and then determines direct relations k in the path ($1 \leq k \leq m$). This process in both temporal and causal relation identification takes $O(2 \times m \times k)$. Moreover, the process of temporal relation identification takes $O(5m)$ for processing direct relation rules, time and tense rules and graph update (Lines 2-3, 5-6, 11 in Algorithm 6.1) while the process of causal relation identification takes $O(3m)$ for processing direct relation rules and graph update (Lines 2-3, 8 in Algorithm 6.2). The details of the time complexity are shown in Table 6.12. In case when each path has only one relation, the best time complexity is $O(m)$. In case when the paths have m relations, the worst time complexity is $O(m^2)$.

Table 6.12. Time complexity.

Time complexity	Cost
Open IE	$O(n)$
Temporal relation identification	$O(6m+2mk)$; $O(m)$; $O(m^2)$
Causal relation identification	$O(3m+2mk)$; $O(m)$; $O(m^2)$

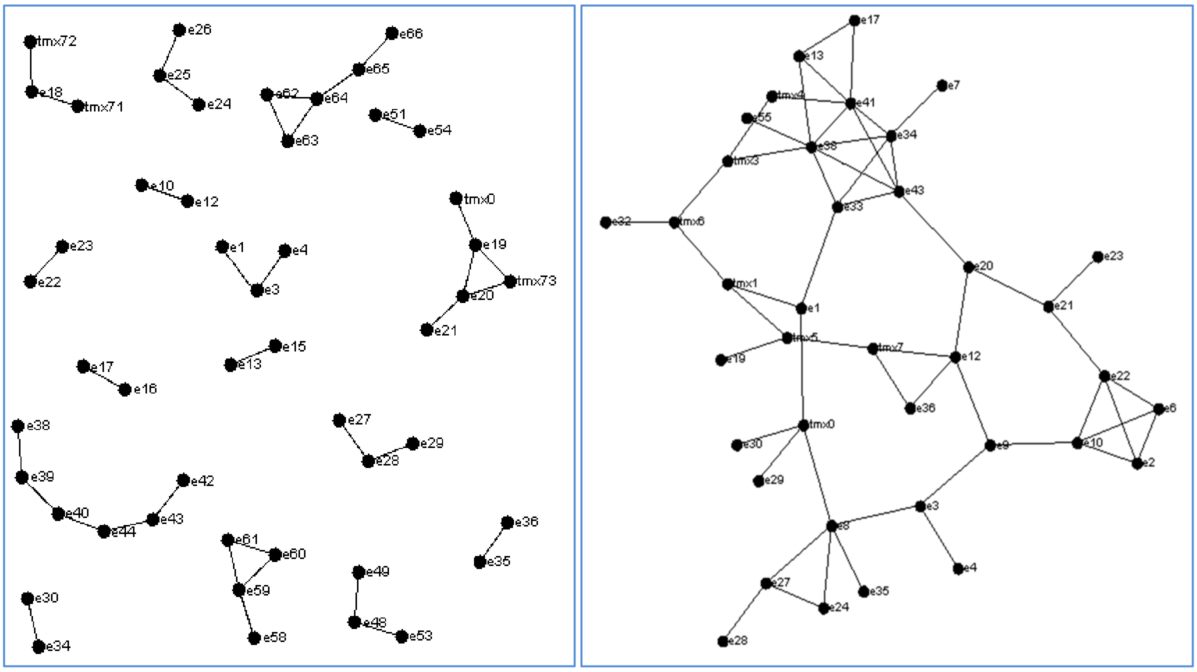

(a) (b)

Figure 6.8. Examples of event networks: a) A sparse network; b) A complex network.

Now from a performance perspective, we note that the results produced by the event network can depend on the structure of the network including the number of event nodes, the number of relation edges and their connections in the network. Figure 6.8 shows two sample event networks extracted from documents APW19980227.0487 and APW19980227.0487 in the APW corpus used in our experiments. As seen in the figure, one event network is quite sparse (Figure 6.8a) while the other event network is quite dense (Figure 8b). We find that when the event network is sparse, our approach is not effective in that it would not have sufficient number of edges in the network to perform inference. As result, our approach cannot identify a sufficient number of relations in such a case, e.g., relations $R(e_1, e_{35})$ and $R(e_1, e_{27})$, which should have been extracted, were not identified in Figure 6.8a due to missing edges in the event network. On the other hand, the dense event network consists of a higher number of edges connecting the event nodes to each other and hence provides the opportunity for a higher number of inferred causal and temporal relations. For instance, the relation $R(e_{33}, e_{41})$, which is difficult to extract, was identified by traversing a long path consisting of $e_{33}\rightarrow e_{38}\rightarrow e_{41}$, $e_{33}\rightarrow e_{43}\rightarrow e_{41}$, $e_{33}\rightarrow e_{34}\rightarrow e_{41}$, $e_{33}\rightarrow e_{38}\rightarrow e_{13}\rightarrow e_{41}$. Based on this observation, we believe that our approach faces some limitations: 1) in light of the sparse event networks, our proposed approach is dependent on the performance of the underlying Open IE system and hence in cases when the Open IE system cannot extract event mentions, the corresponding event nodes will not be created in the event network and hence relations will be missed. 2) our method relies on reference mapping to identify similar event nodes in the event graph, which is currently performed through cosine similarity between the textual description of the events. However, more complex text matching methods for event types can be used in the future to improve the reference mapping process that can lead to better overall performance.

6.8. Concluding remarks

In this chapter, we have presented an unsupervised method for extracting temporal and causal relations between events by building an event network structure primarily based on information from Open IE systems. The event network is the basis for systematically exploring the possible temporal relations between events by considering how events can be reached from one another. We performed comparative benchmarking of our proposed method using the TempEval-3, Causal-TimeBank, Causal-TimeBank datasets and compared our work against several strong baselines. The results reveal that our method outperforms not only in temporal event extractions but also causal event extraction over other state-of-the-art IE systems. Our future work will consist of addressing the two limitations of our work, namely quantifying the impact of the performance of the Open IE systems on our work and also exploring more systematic ways for performing reference mapping.

Chapter 7

Conclusions and Future Work

7.1. Concluding Remarks

The purpose of this thesis was to take and explore advantage of clause-based patterns for information extraction tasks. The research directions of the thesis focus on four important challenges in information extraction: 1) Clause-based Open Information Extraction and (OIE), 2) Self-training in Relation Extraction (RE), 3) Matrix models for Relation Extraction, and 4) Event Networks for Event Relation Extraction.

Clause-based Open Information Extraction. We have proposed refinements to the grammatical structure of syntactic and dependency parsing and have used clause-based relations in order to determine clause types based on their grammatical constituents for open information extraction presented in Chapter 3. Regarding the context of open information extraction, relation extraction is concerned with identifying a variety of relations with their arguments in arbitrary sentences. Some of the existing work that focus on the use of syntactic and dependency parsing for detecting relations sometimes result in incoherent and uninformative extractions such that the extracted relation has no meaningful interpretation. In this study, we use the English clause structure and clause types in an effort to generate propositions that can be deemed as extractable relations. Moreover, we propose refinements to the grammatical structure of syntactic and dependency parsing that help reduce the number of incoherent and uninformative extractions from clauses. In our extensive experiments, we compared our method with other OIE systems using three standard benchmark datasets. The results reveal that our method obtained not only higher precision extractions but also had more flexible generation of relations over other state-of-the-art IE systems.

Self-training on Relation Extraction. We have proposed clauses-pattern extraction with a self-training approach for unsupervised relation extraction presented in Chapter 4. Bootstrapping techniques utilized for relation extraction have shown to be effective in terms of interactively expanding a set of initial relations. Such tasks are primarily carried out through semi-supervised classification approaches. Considering that choosing the most efficient seeds are pivotal to the success of the bootstrapping process, these methods depend on a reliable set of seeds or rules that incorporate domain knowledge. In this thesis, we have proposed clause-pattern extraction with self-training for unsupervised relation extraction. We extract patterns based on a clause-based approach that strives to consider several clause types that may contain a relation. The proposed self-training algorithm relies on the clause-based approach to extract a small set of seed instances in order to identify and derive new patterns. A fundamental distinction between our proposed method and other prominent approaches is that we automatically and iteratively extract seeds based on high confidence patterns that are identified through the clause-based approach. In our experiments, we show that our approach improves upon the performance of the current state-of-the-art systems such as DARE up to 26.88% and 14.13% on F-measure over the Nobel and MUC-6 datasets, respectively.

Matrix Models for Relation Extraction. In Chapter 5, we focus on the problem of information extraction as a matrix completion problem where we employ the notion of universal schemas formed as a collection of patterns derived from clause-based open information extraction systems as well as additional features derived from grammatical clause patterns and statistical topic models. One of the challenges with earlier work that employ matrix completion methods is that such approaches require a sufficient number of observed relation instances to be able to make predictions. However, in practice there is often insufficient number of explicit evidences supporting each relation type that could be used within the matrix model. Hence, existing work suffer from a low recall. In our work, we extend the work in the state of the art by proposing novel ways of integrating two sets of features, i.e., topic models and grammatical clause structures, for alleviating the low recall problem. More specifically, we propose that it is possible to (1) employ grammatical clause information from textual sentences to serve as an implicit indication of relation type and argument similarity. The basis for this is that it is likely that similar relation types and arguments are observed within similar grammatical structures, and (2) benefit from statistical topic models to determine similarity between relation types and arguments. We employ statistical topic models to determine relation type and argument similarity based on their co-occurrence within the same topics. We have performed extensive experiments based on both gold standard and silver standard datasets. The experiments show that our approach has been able to address the low recall problem in existing methods, which lead to improved overall F-measures up to 8% over the state of the art.

Event Networks for event relation extraction. We present an event network representation for temporal and causal relation extraction that operates at the document level presented in Chapter 6. Event relations specify how different event flows expressed within the context of a textual passage relate to each other in terms of temporal and causal sequences. There have already been impactful work in the area of temporal and causal event relation extraction; however, they are mostly supervised methods that rely on syntactic and grammatical structure patterns at the sentence-level. In this thesis, we benefit from existing Open IE systems to generate a set of triple relations that are then used to build an event network. The event network is bootstrapped by labeling the temporal disposition of events that are directly linked to each other. We then systematically traverse the event network to identify the temporal and causal relations between indirectly connected events. We have performed experiments based on the widely adopted TempEval-3 and Causal-TimeBank corpora and compared our work with several strong baselines. We have shown that our unsupervised method is able to show better performance over its supervised counterparts.

7.2. Future work

There is increasing interest in interpreting natural languages and machine learning techniques that focus on the tasks of information extraction explored in this dissertation. Here, we briefly list two promising future directions.

The first direction considers deep learning for clause-based Open Information Extraction. Currently, more recent deep learning methods focus on tasks such as multitask learning as they allow the learning of shared representations among different tasks (Nguyen et al., 2017, Nguyen et al., 2018). This encourages the building of IE methods that can support multiple tasks. Clause-based OIE are usually built on hand-crafted patterns from other NLP tools such as syntactic and dependency parsers; hence clause patterns could face problems with proactive identification and correction of repetitive errors (Corro et al., 2013; Vo & Bagheri 2018). Our main idea for future work is to explore deep learning based multi-task learning models that can solve several information extraction tasks by learning highly confident arguments and relation tuples bootstrapped for Open IE. Once the Open IE improves its performance in extracting relation triples, it will benefit the reliability of other models that use it for bootstrapping, matrix models and event networks.

The second direction concerns the consideration of neural embeddings (Bagheri et al., 2018; Ensan et al., 2017) within the context of the proposed models for information extraction. We would like to explore how neural embedding-based features could be developed that measure relation type and relation argument relevance and similarity for predicting potentially relevant yet unobserved relation instances in matrix model and event networks. For example, one might be able to learn neural embedding models that determine that a relation type such as "CEO-of" would be more similar to the "Director-of" relation type compared to the "President-of" relation type. Neural embedding models have already shown improved performance on several IR tasks (Ensan et al., 2017; Jansen et al., 2010) and hence could be helpful in improving the performance of the relation extraction task as well. We are also interested in defining features based on graph distance and traversal methods for event networks such as random walks to establish a measure of relevance between relation types and arguments. In order to achieve this, we will explore how graphs can be formed based on the unification of relation types and/or relation argument entity matching. We also consider temporal features in order to recognize out of date information that help avoid extracting incorrect relations.

Bibliography

[1] Abacha, A.B., Zweigenbaum, P. (2016). MEANS: A medical question-answering systems combining NLP techniques and semantic Web technologies. *Information Processing & Management*, vol. 51, pp. 570-594.

[2] Abebe, A.M., Tekli, J., Getahun, T.F., Chbeir, R., Tekli, G. (2019). Generic metadata representation framework for social-based event detection, description, and linkage. *Knowledge-Based Systems*, In press, corrected proof, Available online 9 July 2019.

[3] Ahn, D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, Sydney, Australia, July 23 - 23, 2006.

[4] Agichtein, E., Gravano, L. (2000). Snowball: Extracting Relation from Large Plain-text Collections. In *Proceedings of the fifth ACM conference on Digital Libraries 2000 (DL 2000)*, San Antonio, Texas, USA, June 02-07, 2000.

[5] Akbik, A., Visengeriyeva, L., Herger, P., Hemsen, H. and Loser, A. (2012). Unsupervised discovery of relations and discriminative extraction patterns. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 17–32, Mumbai, India, December 2012.

[6] Akbik, A., Bergmann T. and Vollgraf R. (2019). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies NAACL-HLT 2019*, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, USA. Association for Computational Linguistics, pp. 724–728.

[7] Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26 (11), pp. 832–843.

[8] Allerton, D. J. (2002). Stretched Verb Constructions in English. *Routledge Studies in Germanic Linguistics*. Routledge (Taylor and Francis), New York, (2002).

[9] Amigo, E., Artiles, J., Li, Q., and Ji, H. (2011). An evaluation framework for aggregated temporal information extraction. In *Proceedings of SIGIR Workshop on Entity-Oriented Search*, July 28, 2011, Beijing, China.

[10] Angeli, G., Tibshirani, J., Wu, J., Manning, C.D. (2014). Combining Distant and Partial Supervision for Relation Extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2012)*, Doha, Qatar, October 25-29, 2014.

[11] Bagheri, E., Ensan, F., Al-Obeidat, F. (2018). Neural Word and Entity Embeddings for Ad hoc Retrieval. *Information Processing & Management*, vol. 54, pp. 339–357, 2018.

[12] Baldwin, T., Marneffe, M.C., Han, B., Kim, Y.B., Ritter, A., Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

[13] Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O. (2007). Open Information Extraction from the Web. In *Proceedings of the 20th international joint conference on Artificial Intelligence (IJCAI 2007)*, pp. 2670-2676, Hyderabad, India, 06-12 January 2007.

[14] Barrio, P., Gravano, L. (2017). Sampling strategies for information extraction over the deep web. *Information Processing & Management*, vol. 53, Issue 2, March 2017, Pages 309-331

[15] Batista, D.S., Martins, B., Silva, M.J. (2015). Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics. In *Proceedings of the 2016 conference on Empirical Methods in Natural Language Processing (EMNLP2015)*, pp. 499-504, Lisbon, Portugal, 17-21 September 2015.

[16] Bethard, S. and Martin, J. H. (2006). Identification of event mentions and their semantic class. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 146–154, Sydney, July 2006.

[17] Bethard, S., Martin, J. H., and Klingenstein, S. (2007). Timelines from text: Identification of syntactic temporal relations. In *Proceedings of the International Conference on Semantic Computing*, 17-19 Sept. 2007, Irvine, CA, USA.

[18] Blei, D., Andrew, Y. Ng., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.

[19] Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *In Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann Publishers (pp. 92–100).

[20] Bollegala, D., Matsuo, Y., Ishizuka, Y. (2010). Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th international conference on World Wide Web (WWW 2010)*, North Carolina, USA, April 26-30, 2010

[21] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data, In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*, Lake Tahoe, Nevada, December 05-10, 2013.

[22] Bravata, D. M., & Olkin, I. (2001). Simple Pooling versus Combining in Meta-Analysis. *Evaluation & the Health Professions*, 24(2), 218–230. doi:10.1177/01632780122034885

[23] Brin, S. (1998). Extracting Patterns and Relations from the World Wide Web. In *Proceedings of the International Workshop on the World Wide Web and Databases (WebDB98)*, pp. 172-183, 27-28 March 1998.

[24] Bunescu, R. & Mooney, R. J. (2005). A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, Vancouver, B.C, Canada - October 6-8, 2005.

[25] Carreras, X., Marquez, L. and Padro, L. (2002). Named entity extraction using adaboost. In *Proceedings of the 6th conference on natural language learning.* August, 31:1–4.

[26] Chambers, N. and Jurafsky, D. (2008). Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 2008)*, pp. 698-706. Honolulu, Hawaii, October 25 - 27, 2008.

[27] Chambers, Nate (2013). NavyTime: Event and Time Ordering from Raw Text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 73–77. 2013.

[28] Chambers, N., Cassidy, T., McDowell, B., Bethard, S. (2014). Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

[29] Choi, M., Kim, H. (2013). Social relation extraction from texts using a support vector machine-based dependency. *Information Processing & Management*, vol. 49. pp. 303-311.

[30] Christensen, J., Mausam, Soderland, S., and Etzioni, O. (2011). An analysis of open information extraction based on semantic role labeling. In *Proceedings of KCAP (2011)*. 2011.

[31] Christensen, J., Mausam, Soderland, S., and Etzioni, O. (2013). Towards coherent multi-document summarization. In *Proceedings of NAACL (2013)*. 2013.

[32] Christensen, J., Soderland, S., Bansal, G., Mausam. (2014). Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of ACL (2014)*. 2014.

[33] Collins, M., and Singer, Y. (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

[34] Collins, M., Dasgupta, S., Schapire., R.S. (2001). A generalization of principal component analysis to the exponential family. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pp. 617-624, Vancouver, B.C, Canada - December 3-8, 2001.

[35] Corro, L.D., Gemulla, R. (2013). ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the 22nd international conference on World Wide Web (WWW 2013)*, pp. 355-366, Rio de Janeiro, Brazil, 13-17 May 2013.

[36] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, vol. 12, pages 2493-2537. 2011.

[37] Croce, D., Moschitti, A., Basili, R. (2011). Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the 2011 conference*

on Empirical Methods in Natural Language Processing (EMNLP 2011), pp. 1034-1046, Edinburgh, UK, 27-31 July 2011.

[38] Culotta, A., Sorensen, J. (2004). Dependency Tree Kernels for Relation Extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, July 21-26, pp. 423–429.

[39] Culotta, A., McCallum, A., & Betz, J. (2006). Integrating probabilistic extraction models and datamining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (pp. 296–303). New York: Association for Computational Linguistics.

[40] Cui, L., Wei, F., Zhou, M. (2018). Neural Open Information Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), Melbourne, Australia. 2018.

[41] D'Souza, Jennifer and Vincent Ng (2013). Classifying Temporal Relations with Rich Linguistic Knowledge. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Atlanta, Georgia: Association for Computational Linguistics, pp. 918–927.

[42] Denis, P. and Muller, P. (2011). Predicting globally coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, p.1788-1793, July 16-22, 2011, Barcelona, Catalonia, Spain.

[43] Dernoncourt, F., Lee, J.Y., Szolovits, P. (2017). Neuroner: an easy-to-use program for named-entity recognition based on neural networks. *arXiv* preprint arXiv:1705.05487.

[44] Do, X.Q., Lu, W. and Roth, D. (2012). Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687. Association for Computational Linguistics. 2012.

[45] Durme, B.V., Schubert, L.K. (2008) Open knowledge extraction using compositional language processing. In *Proceedings of STEP (2008)*.

[46] Ensan, F. and Bagheri, E. (2017). Document retrieval model through semantic linking. In *Proceedings of the 10th ACM International conference on web search*

and data mining WSDM 2017, pages 181-190, Cambridge, UK – February 6-10, 2017.

[47] Eltyeb, S., Salim, N. (2014). Chemical named entities recognition: a review on approaches and applications. *Journal of Cheminformatics*. vol. 6(1):17.

[48] Etzioni, O., Banko, M., Soderland, S., Weld, D.S. (2008). Open Information Extraction from the Web. *Communication of the ACM*, vol. 51, no. 12 (2008).

[49] Etzioni, O., Cafarella, M., Downey, D, Popescu, A., Shaked, T., Soderland, S., Weld, D.S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, vol. 165(1):91 – 134.

[50] Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam. (2011). Open Information Extraction: the Second Generation. In *Proceedings of the 22nd international joint conference on Artificial Intelligence (IJCAI 2012)*, pp. 3-10, Barcelona, Catalonia, Spain, 16-22 July 2011.

[51] Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam. (2012). Open Information Extraction: the Second Generation. In *Proceedings of IJCAI (2012).*

[52] Fan, B., Fan, W., Smith, C., Garner, H. (2019). Adverse drug event detection and extraction from open data: A deep learning approach. *Information Processing & Management*, 2019. Doi: https://doi.org/10.1016/j.ipm.2019.102131.

[53] Fader, A., Soderland, S., Etzioni, O. (2011). Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pp. 1035-1545, Edinburgh, UK, 27-31 July 2011.

[54] Feng, J., Huang, M., Zhao, Li., Yang, Y., Zhu, X. (2018). Reinforcement Learning for Relation Classification from Noisy Data. In *Proceeding of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, New Orleans, Louisiana, USA, February 2-7, 2018.

[55] Fei, H., Ren, Y., Ji, D. (2019). A tree-based neural network model for biomedical event trigger detection. *Information Sciences*, Volume 512, February 2020, Pages 175-185.

[56] Fukui, K., Okada, Y., Satoh, K., Numao, M. (2019). Cluster sequence mining from event sequence data and its application to damage correlation analysis. *Knowledge-Based Systems*, Volume 179, pp. 136-144 1 September 2019.

[57] Gamallo, P., Garcia, M., Fernandez-Lanza, S. (2011). Dependency-based Open Information Extraction. In *Proceedings of the ROBUS-UNSUP 2012: Joint*

Workshop on Unsupervised and Semi-Supervised Learning in NLP at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Avignon, France, April 23-27, 2012.

[58] Garcia, M., Gamallo, P. (2011). Dependency-based text compression for semantic relation extraction. In *Proceedings of the Workshop Information Extraction and Knowledge Acquisition (IEKA 2011)*, Hissar, Bulgaria, 16 September 2011.

[59] Greenwood, M. A., Stevenson, M. (2006). Improving Semi-supervised Acquisition of Relation Extraction Patterns. In *Proceedings of the Workshop on Information Extraction Beyond the Document (IEBD 2006)*, pp. 19-35, Sydney, Australia, 22 July 2006.

[60] Grishman, R. and B. Sundheim (1996). Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen.

[61] Grishman, R., Huttunen, S., and Yangarber, R. (2002). Real-time event extraction for infectious disease outbreaks. In *Proceedings of the second international conference on Human Language Technology Research (HLT)*, San Diego, California, March 2002.

[62] Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, D.P., Yuret, D. (2009). Classification of semantic relations between nominals. *Language Resources and Evaluation*, vol. 43(2): 105-121 (2009).

[63] GuoDong, Z., Jian, S., Jie, Z., & Min, Z. (2002). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 419–444.

[64] Gupta. S., Manning, C. D. (2014). Spied: Stanford pattern-based information extraction and diagnostics. In *Proceedings of the ACL Workshop on Interactive Language Learning, Visualization, and Interfaces (ACL-ILLV 2014)*, pp. 38-44. Baltimore, Maryland, USA, 27 June 2014.

[65] Habibi, M., Weber, L., Neves, M., Wiegandt, D.L, Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, vol. 33(14): i37–i48. 2017.

[66] Han, R., Hsu, I., Yang, M., Galstyan, A., Weischedel, R., Peng, N. (2019). Deep Structured Neural Network for Event Temporal Relation Extraction. In *Proceedings of The SIGNLL Conference on Computational Natural Language Learning 2019 (CoNLL 2019)*, Hongkong, November 3-4, 2019.

[67] Hardy, H., Kanchakouskaya, V., and Strzalkowski, T. (2006). Automatic event classification using surface text features. In *Proceedings of the 2006 AAAI Workshop on Event Extraction and Synthesis*. 2006.

[68] Hoang, T.B. N., Mothe, J. (2018). Location extraction from tweets. *Information Processing & Management*, Volume 54, Issue 2, March 2018, Pages 129-144.

[69] Hashimoto, Kazuma, Stenetorp, P., Miwa, M., and Tsuruoka, Y. (2015). Task-Oriented Learning of Word Embeddings for Semantic Relation Classification. *In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, China: Association for Computational Linguistics*, pp. 268–278.

[70] Hogenboom, F., Frasincar, F., Kaymak, U., Jong, F.D., Caron, E. (2016). Survey of event extraction methods from text for decision support system. *Decision Support Systems*, vol. 85, pp. 12-22, May 2016.

[71] Hu, Z., Rahimtoroghi, E. and Walker, M.A. (2017). Inference of fine-grained event causality from blogs and films. *arXiv* preprint arXiv:1708.09453. 2017.

[72] Huang, Z., Xu, W., Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv* preprint arXiv:1508.01991.

[73] Ittoo, Ashwin and Gosse Bouma, G. (2011). Extracting Explicit and Implicit Causal Relations from Sparse, Domain-Specific Texts. In *Natural Language Processing and Information Systems - 16th International Conference on Applications of Natural Language to Information Systems (NLDB 2011)*, Alicante, Spain, June 28-30, 2011. Proceedings, pp. 52–63.

[74] Jansen, B. J. and Rieh, S. (2010). The Seventeen Theoretical Constructs of Information Searching and Information Retrieval. *Journal of the American Society for Information Sciences and Technology*. Vol. 61(8), 1517-1534.

[75] Ji, H. and Grishman, R. (2008). Refining event extraction through cross document inference. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, Pages 254-262, Columbus, Ohio, USA, June 2008.

[76] Ji, H., Grishman, R., and Dang, H. T. (2011). Overview of the tac2011 knowledge base population track. In *Proceedings of Text Analysis Conference 2011 (TAC2011)*. 2011.

[77] Ji H., Favre B., Lin WP., Gillick D., Hakkani-Tur D., Grishman R. (2013). Open-Domain Multi-Document Summarization via Information Extraction: Challenges

and Prospects. In *Multi-source, Multilingual Information Extraction and Summarization. Theory and Applications of Natural Language Processing.* Springer, Berlin, Heidelberg.

[78] Kambhatla, N. (2004). Combining Lexical, Syntactic and Semantic Features with Maximum Entropy Models for Extracting Relations. In *Proceedings of the Association for Computational Linguistics (ACL2004)*, pp. 178-181, Barcelona, Spain, 21-26 July 2004.

[79] Kang, U., Papalexakis, E., Harpale, A., &Faloutsos, C, Gigatensor. (2012). Scaling tensor analysis up by 100 times - algorithms and discoveries, In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, Beijing, China, August 12 - 16, 2012.

[80] Katsumasa, Y., Riedel, S., Asahara, M. and Matsumoto, Y. (2009). Jointly Identifying Temporal Relations with Markov Logic. 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*. Suntec, Singapore: Association for Computational Linguistics, pp. 405–413. 2009.

[81] Kim, H.H., Kim, Y.H. (2018). Video summarization using event-related potential responses to shot boundaries in real-time video watching. *Journal of the Association for Information Science and Technology*, vol. 70, Issue 2, November 2018.

[82] Kemp, C., Tenenbaum, J.B., Griffiths, T.L. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st national conference on Artificial intelligence 2006*, Volume 1, pp. 381-388, Boston, Massachusetts - July 16-20, 2006.

[83] Kok, S. and Domingos, P. (2008). Extracting Semantic Networks from Text via Relational Clustering. In *Proceedings of the 2008 European conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 2008*, pp. 624-639, Antwerp, Belgium, 15-19 September 2008.

[84] Kolda, T. G., Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, vol. 51, pp. 455-500, 2009.

[85] Koren, Y. (2009). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining 2009*, pp. 426-434, Las Vegas, Nevada, USA - August 24-27, 2008.

[86] Johnstone, I. M., Lu, A. Y. (2009). On Consistency and Sparsity for Principal Components Analysis in High Dimensions. *Journal of the American Statistical Association*, vol. 204, pp. 682-693. 2009.

[87] Junuthula, R., Haghdan, M., Xu, K.S, Devabhaktuni, V. (2019). The block point process model for continuous-time Event based Dynamic Networks. In *Proceedings of the World Wide Web conference (WWW 2019)*, pp. 829-839, San Francisco, CA, USA, May 13-17, 2019.

[88] Laokulrat, N., Miwa, M., Tsuruoka, Y. (2015). Stacking approach to temporal relation classification with temporal inference. *Journal of Natural Language Processing*, 22(3):171–196.

[89] Laokulrat, N., Miwa, M., Tsuruoka, Y., Chikayama, Takashi. 2013. Uttime: Temporal relation classification using deep syntactic features. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 88–92, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

[90] Lenzi, B., Valentina, Moretti, G. and Sprugnoli, R. (2012). CAT: the CELCT Annotation Tool. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. 21-27 May 2012, Istanbul, Turkey.

[91] Levy, O., Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014),* Baltimore, Maryland.

[92] Lodhi, H., Saunders, C., Shawe-Taylor, J., & Cristianini, N. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, vol. 2, pp. 419–444.

[93] Li, H., Ji, H., Deng, H., and Han, J. (2011). Exploiting background information networks to enhance bilingual event extraction through topic modeling. In *Proceedings of International Conference on Advances in Information Mining and Management (IMMM 2011)*. 2011.

[94] Li, Y., Bontcheva, K., Cunningham, H. (2005). SVM based learning system for information extraction. In *Deterministic and statistical methods in machine learning*, pages 319–339. Springer.

[95] Limsopatham, N., Collier, N.H. (2016). Bidirectional LSTM for named entity recognition in twitter messages. In *Proceedings of the 2nd workshop on Noisy User-generated Text*, December, 2016, Osaka, Japan.

[96] Lin, Y., Liu, Z., Sun, M. (2017). Neural Relation Extraction with Multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. 2017.

[97] Liu, T., Wang, K., Chang, B., Sui, Z. (2017). A Soft-label Method for Noise-tolerant Distantly Supervised Relation Extraction. In *Proceedings of the 2017 conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. 2017.

[98] Malouf, R. (2002). Markov models for language-independent named entity recognition. In *Proceedings of the 6th conference on natural language learning*. August, 31:1–4.

[99] Mani, I., Verhagen, M., Wellner, B., Lee, C.M., Pustejovsky, J. (2006). Machine Learning of Temporal Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, pp. 753–760. 2006.

[100] Mani, Inderjeet, Wellner, B., Verhagen, M. and Pustejovsky, J. (2007). Three Approaches to Learning TLINKs in TimeML. *Tech. rep*. Computer Science Department, Brandeis University. 2007.

[101] Mausam, Schmitz, M., Bart, R., Soderland, S. (2012). Open Language Learning for Information Extraction. In *Proceedings of the 2012 conference on Empirical Me-thods in Natural Language Processing (EMNLP 2012)*, pp. 523-534, Jeju Island, Korea, 12-14 July 2012.

[102] Moon, S., Neves, L., Carvalho, V. (2018). Multimodal Named Entity Recognition for Short Social Media Posts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), New Orleans, Louisiana, 2018.

[103] Min, B., Grishman, R., Wan, Li., Wang, C., Gondek, D. (2013). Distant Supervision for Relation Extraction with an Incomplete Knowledge Base. In *Proceedings of the 2013 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies (NAACL 2013), Atlanta, Georgia, 9-14 June 2013.

[104] Mintz, M., Bills, S., Snow, R., Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association on Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009)*, pp. 1003-1011, Suntec, Singapore, 2-7 August 2009.

[105] Mirza, P., and Sara Tonelli, S. (2014a). An analysis of causality between events and its relation to temporal information. In P*roceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2097–2106, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics. 2014.

[106] Mirza, P., Sara Tonelli. (2014b). Classifying temporal relations with simple features. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 308–317, Gothenburg, Sweden, April. Association for Computational Linguistics.

[107] Mirza, P., Tonelli, S. (2016a). Catena: Causal and temporal relation extraction from natural language texts. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 64–75. Association for Computational Linguistic.

[108] Mirza, P.(2016b). Extracting temporal and causal relations between events. *Ph.D Dissertation*. 2016.

[109] Mendes, N.P., Mühleisen, H., and Bizer, C. (2012). Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops (EDBT-ICDT '12)*, 116-123. 2012.

[110] Nadeau, D., Turney, P.D., Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 266–277. Springer.

[111] Navarro-Colorado, B., Saquete, E. (2016). Cross-document event ordering through temporal, lexical and distributional knowledge. *Knowledge-Based Systems*, Volume 110, p.p. 244-254, 15 October 2016.

[112] Nebot, V. and Berlanga, R. (2014). Exploiting semantic annotations for open information extraction: an experience in the biomedical domain. *Knowledge and Information Systems*, vol. 38(2): 365-389 (2014).

[113] Negi, K., Pavuri, A., Patel, L., Jain, C. (2019). A novel method for drug-adverse event extraction using machine learning. *Informatics in Medicine Unlocked*, 2019. Doi: https://doi.org/10.1016/j.imu.2019.100190.

[114] Nguyen T.H. and Grishman R. (2015). Relation extraction: perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, USA. Association for Computational Linguistics, pp. 39–48.

[115] Nguyen T.H., Cho K. and Grishman R. (2016). Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California. Association for Computational Linguistics, pp. 300–309.

[116] Nguyen, D.B., Theobald, M., Weikum, G. (2017). J-REED: Joint Relation Extraction and Entity Disambiguation. In *Proceedings of CIKM 2017*, November 6-10, 2017, Singapore.

[117] Nguyen T.H. (2018). Deep Learning for Information Extraction. *Ph.D thesis*. New York University. 2018.

[118] Nguyen, T.N., Dernoncourt, F., Nguyen, T.H. (2019). On the Effectiveness of the Pooling Methods for Biomedical Relation Extraction with Deep Learning. *arXiv* preprint arXiv:1911.01055. 2019.

[119] Qian, Y., Deng, X., Ye, Q., Ma, B., Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. *Information Processing & Management*, Volume 56, Issue 6, November 2019.

[120] Qin, P., Xu, W., Wang, W. Y. (2018). Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. 2018.

[121] Oramasa, S., Espinosa-Ankeb, L., Sordoc, M., Saggionb, H., Serraa, H. (2016). Information extraction for knowledge base construction in the music domain. *Data & Knowledge Engineering*. http://dx.doi.org/10.1016/j.datak.2016.06.001.

[122] Pantel, P., Pennacchiotti, M. (2006). Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 113-120, Sydney, Australia, 17-18 July 2006.

[123] Patwardhan S., Riloff E. (2007). Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *Proceedings of the 2007 joint conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pp. 717-727, Prague, June 2007.

[124] Peng, H., Song, Y., Roth, D. (2016). Event Detection and Co-reference with Minimal Supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-2016)*, pages 392–402, Austin, Texas, November 1-5, 2016.

[125] Phan, X. H., Nguyen, C. T., Le, D. T., Nguyen, L. M., Horiguchi, S., & Ha, Q. T. (2011). A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, 961–976.

[126] Poon, H., Domingos, P. (2009). Unsupervised semantic parsing. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP 09)*, Singapore, August 06 - 07, 2009.

[127] Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., and Katz, G. (2003). Timeml: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*. 2003.

[128] Pustejovsky, James, Littman, J., Saurí, R., and Verhagen, M. (2006). TimeBank 1.2 Documentation. *Tech. rep.*, Brandeis University. 2006.

[129] Olya, Hessam, M. (2014). Applying Dijkstra's algorithm for general shortest path problem with normal probability distribution arc length. International *Journal of Operational Research*. Vol. 21 (2): pp. 143–154. 2014. doi:10.1504/IJOR.2014.064541.

[130] Ozdikis, O., Oğuztüzün, H., Karagoz, P. (2016). Evidential estimation of event locations in microblogs using the Dempster–Shafer theory. *Information*

Processing & Management, Volume 52, Issue 6, Pages 1227-1246, November 2016.

[131] Qian, Y., Deng, X., Ye, Q., Ma, B., Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. *Information Processing & Management*, Volume 56, Issue 6, November 2019.

[132] Quirk, R., Greenbaum, S., Leech, G., Svartvik, J. (1985). A Comprehensive Grammar of the English Language. Longman, 1985.

[133] Ravichandran, D. and Hovy, E.H. (2002). Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pp. 41-47, Philadelphia, July 2002.

[134] Reddy, S., Lapata, M., Steedman, M. (2014). Large-scale Semantic Parsing without Question-Answer Pairs. *TACL*, vol. 2, pp. 377-392 (2014).

[135] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. (2009). Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence 2009*, pp. 452-461, Montreal, Quebec, Canada - June 19-21, 2009.

[136] Riaz, M. and Girju, R. (2010). Another look at causality: Discovering scenario-specific contingency relationships with no supervision. In *Proceedings of the International Conference on Semantic Computing 2010*. 22-24 Sept. 2010, Pittsburgh, PA, USA. 2010.

[137] Riaz, M. and Girju, R. (2013). Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'13)*. 21-30. 2013.

[138] Riedel S., Yao. L, McCallum, A., Marlin, M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)*, pp. 74-84, Atlanta, Georgia, 9-14 June 2013.

[139] Riedel, S., Yao, L., McCallum, A. (2010). Modeling Relations and Their Mentions without Labeled Text. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2010)*, Barcelona, Spain, 20-24 September 2010.

[140] Rink, B., Bejan, C.A. and Harabagiu, M.S. (2010). Learning textual graph patterns to detect causal event relations. In *Proceedings of the twenty-Third International Florida Artificial Intelligence Research Society Conference 2010* (FLAIRS 2010).

[141] Rosenfeld, B. and Feldman, R. (2007). Clustering for unsupervised relation identification. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM2007)*, pages 411–418, Lisbon, Portugal, 6-10 November 2007.

[142] Ryu, P.M., Jang, M.G., Kim. (2015). Open domain question answering using Wikipedia-based knowledge model. *Information Processing & Management*, Vol. 50, pp. 683-692.

[143] Santos, C.D, Xiang, B., Zhou, B. (2015). Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pp. 626-634, Beijing, China, 26-31 July 2015.

[144] Sager, N., C. Friedman, and S. Margaret. (1987). Medical Language Processing: Computer Management of Narrative Data. *Addison-Wesley*.

[145] Singhal, A., Simmons, M., Lu, Z. (2016). Text mining for precision medicine: automating disease-mutation relationship extraction from biomedical literature. *Journal of the American Medical Informatics Association*. http://dx.doi.org/10.1093/jamia/ocw041.

[146] Socher, R., Huval, B., Manning, C.D., Ng, A.Y. (2012). Semantic compositionality through recursive matrix vector spaces. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP 2012)*, pp. 1201-1211, Jeju Island, Korea, 12-14 July 2012.

[147] Soderland, S., Gilmer, J., Bart, R., Etzioni, O., Weld, D.S. (2013). Open Information Extraction to KBP Relation in 3 Hours. In *Proceedings of KBP* (2013).

[148] Song, M., Kim, W.C., Lee, D., Heo, G.E., Kang, K.Y. (2015). PKDE4J: Entity and relation extraction for public knowledge discovery. *Journal of Biomedical Informatics*, Vol. 57, October 2015, Pages 320-332

[149] Stevenson, M. (2007). Fact distribution in information extraction. *Language Resources and Evaluation*, vol. 40, pp. 183-201.

[150] Stevenson, S., Fazly, A., North, R. (2004). Statistical measures of the semi-productivity of light verb constructions. In *2nd ACL Workshop on Multiword Expressions*, pages 1–8 (2004).

[151] Suchanek, F., Kasneci, G. and Weikum, G. (2007). YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pp. 697-706, Banff, Alberta, Canada, 8-12 May 2007.

[152] Sudo, K., Sekine, S., Grishman, R. (2003). An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pp. 224-236, Sapporo, Japan, 7-12 July 2003.

[153] Suppes, P. (1970). A Probabilistic Theory of Causality. Amsterdam: North-Holland Publishing Company. 1970.

[154] Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D. (2012). 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 (EMNLP-CoNLL 2012)*, pp. 455-465, Jeju Island, Korea, 12-14 July 2012.

[155] Swampillai, K., Stevenson, M. (2010). Inter-sentential Relations in Information Extraction Corpora. In *Proceedings of the seventh international conference on Language Resources and Evaluation (LREC 2010)*, Malta, 17-23 May 2010.

[156] Takeuchi, K., Collier, N. (2002). Use of support vector machines in extended named entity recognition. In *Proceedings of the 6th conference on Natural language learning.* Volume 20, pages 1–7. Association for Computational Linguistics.

[157] Takamatsu, S., Sato, I. and Nakagawa, H. (2011). Probabilistic matrix factorization leveraging contexts for unsupervised relation discovery. In *Proceedings of the 15th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2011)*, Schezhen, China, 24-27 May 2011.

[158] Takamatsu, S., Sato, I., Nakagawa, H. (2012). Reducing Wrong Labels in Distant Supervision for Relation Extraction. In *Proceedings of the 50the Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, Jeju, Korea, 8-14 July 2012.

[159] Tanev, H., Piskorski, J., and Atkinson, M. (2008). Real-time news event extraction for global crisis monitoring. In *Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems*. LNAI, Springer-Verlag Berlin Heidelberg. 2008.

[160] Tatu, Marta and Srikanth, M. (2008). Experiments with Reasoning for Temporal Relations between Events. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, Manchester, UK: COLING 2008 Organizing Committee, pp. 857–864.

[161] Thelen M., Riloff E. (2002). A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. In *Proceedings of the 2002 conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 214-221, Stroudsburg, PA, USA, 2002.

[162] Thenmozhi, D., Aravindan, C. (2015). An Automatic and Clause-Based Approach to Learn Relations for Ontologies. *The Computer Journal*, DOI: 10.1093/comjnl/bxv071.

[163] Turney, P.D. (2008). The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, vol. 33, pp. 615-655.

[164] UzZaman, Naushad and Allen, J. (2010). TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden: Association for Computational Linguistics, pp. 276–283.

[165] UzZaman, N., Lorens, H., Derczynski, L., Allen, J., Verhagen, M., Pustejovsky, J. (2013). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

[166] Uzuner, O., Luo, Y., Szolovits, P. (2007). Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

[167] Uzuner, O., South, B.R., Shen, S., DuVall, S.L. (2010). 2010 i2b2 challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

[168] Verhagen, M., Sauri, S., Caselli, T., Pustejovsky, J. (2010). SemEval-2010 task 13: TempEval-2. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval ’10, pages 57–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

[169] Vlachidis, A., Tudhope, D. (2016). A knowledge-based approach to Information Extraction for semantic interoperability in the archaeology domain. *Journal of the Association for Information Science and Technology*, vol. 67, pp. 1138-1152.

[170] Vo, D.T. & Bagheri, E. (2017). Open information extraction. *Encyclopedia Semantic Computing Robotic Intelligence*. 1(1). 2017. doi:10.1142/S2425038416300032

[171] Vo, D.T. and Bagheri, E. (2018). Self-training on refined clause patterns for relation extraction. *Information Processing and Management*, vol. 54, pp. 686-706, July 2018.

[172] Wang, Z., Zhang, Y., Chang, C-Y. (2017). Integrating Order Information and Event Relation for Script Event Prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP-2017)*, pages 57–67, Copenhagen, Denmark, September 7–11, 2017.

[173] Weston, J., Bordes, A., Yakhnenko, O., &Usunier, N. (2013). Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1366–1371, Seattle, Washington, USA, 18-21 October, 2013.

[174] Wu, F., Weld, D.S. (2010). Open Information Extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 118-127, Uppsala, Sweden, 11-16 July 2010.

[175] Xu, F., Uszkoreit, H., and Li, H. (2007). A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity. In *Proceedings of the 45th Annual Meeting of the Association for Comput7ational Linguistics (ACL 2007)*, pp. 584-591, Prague, Czech, June 2007.

[176] Xu, F., Uszkoreit, H., Krause, S., Hong Li, H. (2010). Boosting Relation Extraction with Limited Closed-World Knowledge. In *Proceedings of the 23rd international conference on Computational Linguistics (COLING 2010)*, pp. 1354-1362, Beijing, China, 23-27 August 2010.

[177] Xu, W., Zhang, C. (2014). Trigger word mining for relation extraction based on activation fore. *International Journal of Communication Systems*, vol. 27, pp. 2134–2146.

[178] Xu, Y., Kim, M. Y., Quinn, K., Goebel, R., Barbosa, D. (2013). Open Information Extraction with Tree Kernels. In *Proceedings of the 2013 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pp. 868-877, Atlanta, Georgia, 9-14 June 2013.

[179] Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H. and Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 1785-1794, Lisbon, Portugal, 17-21 September 2015.

[180] Yadav, V., Sharp, R., Bethard, S. (2018). Deep affix features improve neural named entity recognizers. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages. Pp. 167–172, New Orleans, Louisiana, June 2018.

[181] Yahya, M., Whang, S. E., Gupta, R., Halevy. (2014). ReNoun: Fact Extraction for Nominal Attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 325-335, Doha, Qatar, 25-29 October 2014.

[182] Yan, S., Hardmeier, C., Nivre, J. (2016). Multilingual named entity recognition using hybrid neural networks. In *The Sixth Swedish Language Technology Conference (SLTC)*.

[183] Yangarber, R., Grishman, R., Tapanainen, P., Huttunen, S. (2000). Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the 18th conference on Computational Linguistics (COLING 2000)*, pp. 940-946, Saarbrucken, Germany, 31 July - 04 August 2000.

[184] Yao, L. (2015). Ph.D Thesis. University of Massachusetts at Amherst (2015).

[185] Yao, L., Haghighi, A., Riedel, S. and McCallum, A. (2011). Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pp. 1456-1466, Edinburgh, Scotland, UK, 27-31 July 2011.

[186] Yao, L., Riedel, S. and McCallum, A. (2012). Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pp. 712-720, Jeju Island, Korea, 8-14 July 2012.

[187] Yao, X., Durme, B.V. (2014). Information Extraction over Structured Data: Question Answering with Freebase. In *Proceedings of ACL (2014)*.

[188] Yao, L., Liu, H., Liu, Y., Li, X., Anwar, W. (2015). Biomedical named entity recognition based on deep neutral network. *International journal of hybrid information technology*, vol. 8, no. 8, pp.279–288, 2015.

[189] Yates, A. and Etzioni, O. (2007). Unsupervised resolution of objects and relations on the web. In *Proceedings HLT-NAACL (2007)*. 2007.

[190] Ye, H., Luo, Z. (2019). Deep ranking based cost-sensitive multi-label learning for distant supervision relation extraction. *Information Processing & Management*, 2019. Doi: https://doi.org/10.1016/j.ipm.2019.102096.

[191] Yoshikawa, K., Riedel, S., Asahara, M., Matsumoto, Y. (2009). Jointly Identifying Temporal Relations with Markov Logic. 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*, pp. 105-413, Suntec, Singapore.

[192] Zelenko, D., Aone, C., Richardella, A. (2003). Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, vol. 3, pp. 1083–1106.

[193] Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pp. 23-29, Dublin, Ireland, 23-29 August 2014.

[194] Zeng, D., Liu, K., Chen, Y., Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 1753-1762, Lisbon, Portugal, 17-21 September 2015.

[195] Zhao, Y., Jin, X., Wang, Y. (2018). Document Embedding Enhanced Event Detection with Hierarchical and Supervised Attention. In *the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 414-419, Melbourne, Australia, July 15-20, 2018.

[196] Zhao S. and Grishman R. (2005). Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, Ann Arbor, Michigan. Association for Computational Linguistics, pp. 419–426.

[197] Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning ICML '04*, page 116, Banff, Alberta, Canada — July 04 - 08, 2004.

[198] Zhang, S., Elhadad, N. (2013). Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*. Vol. 46(6):1088–1098.

[199] Zhang, C., Xu, X., Ma, Z., Gao, S., Li, Q., Guo. J. (2015a). Construction of Semantic Bootstrapping Models for Relation Extraction. *Knowledge-Based Systems*, vol. 83 (2015), Pages 128-1370.

[200] Zhang, C., Zhang, Y., Xu, W., Ma, Z., Leng, Y., Guo, J. (2015b). Mining Activation Force Defined Dependency Patterns for Relation Extraction. *Knowledge-Based System,* vol. 83, pp. 128-137.

[201] Zhang, S., Lv, Q. (2018). Hybrid EGU-based group event participation prediction in event-based social networks. *Knowledge-Based Systems*, Volume 143, 1 March 2018.

[202] Zhang, H., Boons, F., Batista-Navarro, R. (2019a). Whose story is it anyway? Automatic extraction of accounts from news articles. *Information Processing & Management*, Volume 56, Issue 5, September 2019, Pages 1837-1848.

[203] Zhang, Y., Xu, G., Wang, Y., Liang, X., Huang, T. (2019b). Empower event detection with bi-directional neural language model. *Knowledge-Based Systems*, Volume 167, 1 March 2019, Pages 87-97.

[204] Zheng, S., Hao, Y., Lu, D., Bao, H., Xu, J., Hao, H., Xu, B. (2017). Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, vol. 257, pp. 59-66.

[205] Zheng, X., Suri, A. (2019). Collecting event-related tweets from twitter stream. *Journal of the Association for Information Science and Technology.* Vol. 70, issue 2, pp. 176-186. February 2019.

[206] Zhou, G. and Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for*

Computational Linguistics, pages 473–480. Association for Computational Linguistics.

[207] Zhou, G., Qian, L., Fan, J. (2010). Tree kernel based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, vol. 180, 2010.

[208] Zhou, G., Zhang, M. (2007). Extracting Relation Information from Text Documents by Exploring Various Types of Knowledge. *Information Processing & Management*, vol. 43, 969–982.

[209] Zhu, J., Nie, Z., Liu, X., Zhang, B., Wen, J.R. (2009). StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th International World Wide Web WWW (2009)*, April 20-24, Madrid, Span.