Advancing Open Information Extraction Methods to Enrich Knowledge Bases

by

Seyed Iman Mirrezaei
B.Sc. (Azad University, Central Tehran Branch, Tehran, Iran)
M.Sc. (Sharif University of Technology, Tehran, Iran)

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2017

Chicago, Illinois

Defense Committee:

Professor Isabel F.Cruz, Chair and Advisor
Professor Barbara Di Eugenio
Professor Bing Liu
Professor Brian Ziebart
Professor Bruno Martins, IST and INESC-ID, University of Lisbon

Dedicated to my beloved family

for their limitless love, endless support, and encouragement.

ACKNOWLEDGMENTS

I want to thank my thesis advisor, Professor Isabel F. Cruz, for all her advices, suggestions, and encouragement, and for believing in my capabilities from the beginning of this work. I would like to thank her for guiding me towards the right path. Additionally, I truly appreciate Professor Bruno Martins regarding his constructive comments, suggestions, and guidance for my thesis and papers. I'm grateful to him for giving me the opportunity to work with him and enjoying my time working under his mentorship. I cannot thank Professor Cruz and Professor Martins enough for their help and support during the last years.

My great thanks to all of my friends at ADVIS Lab (Cosmin Stroe, Amruta Nanavaty, Booma Sowkarthiga Balasubramani, and Vivek Revanna Shivaprabhu) for their assistance during my studies. I would like to thank the CS department staff for their indirect help and support. Last but not the least, I would especially like to thank my amazing family for their love and constant encouragement during my study at UIC.

SIM

Contribution of Authors

Chapter 2 is a literature review and we briefly summarize related work in the area of open-domain information extraction. The TRIPLEX pipeline is presented in Chapter 3 and it represents two published manuscripts ("Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In ESWC Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD), 2015" and "Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In The SemanticWeb: ESWC Satellite Events, Revised Selected Papers, volume 9341, DOI: $10.1007/978-3-319-25639-9-39$, pages 230243, 2015") for which I was the primary author and major driver of the research. Professor Cruz and Professor Martins reviewed the manuscripts and suggested some comments about the manuscripts. Chapter 4 represents a published manuscript ("Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. A Distantly Supervised Method for Extracting Spatio-Temporal Information from Text. In ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), DOI: 10.1145/2996913.2996967, 2016") for which I was the primary author and major driver of the research. Professor Cruz and Professor Martins reviewed the manuscripts and suggested some comments about the manuscripts. Chapter 4 describes how TRIPLEX-ST captures spatio-temporal information from text and presents the improvement of OIE methods through rewriting complex sentences. Chapter 4 also represents a series of my own unpublished experiments and it will ultimately be published as a manuscript for which I will be the primary author and major driver of the research.

Chapter 5 presents the possible directions for future work. Finally, Chapter 6 concludes the thesis and summarizes the main aspects that were discussed.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
OIE	Open Information Extractor
IE	Information Extractor
POS tagging	Part-of-speech tagging
PMI	Point wise mutual information
NE	Named entity
NER	Named entity recognition
KB	Knowledge base

SUMMARY

Discovering knowledge from textual sources and subsequently expanding the coverage of knowledge bases like DBpedia or Google's Knowledge Graph currently requires either extensive manual work or carefully designed open information extractors. An open information extractor (OIE) captures triples from textual resources. Each triple consists of a subject, a predicate/property, and an object. Triples can be mediated via verbs, nouns, adjectives, or appositions. The research that we conducted in the area of OIE resulted on the development of OIE systems, named TRIPLEX and TRIPLEX-ST. We focus on further advancing OIE methods to support the expansion of spatio-temporal information in knowledge bases.

TRIPLEX extracts triples from grammatical dependency relations involving noun phrases and modifiers that correspond to adjectives and appositions. TRIPLEX constructs templates that express noun-mediated triples during its automatic bootstrapping process, which finds sentences that express noun-mediated triples by leveraging Wikipedia. The templates express how noun-mediated triples occur in sentences and include rich linguistic annotations. Finally, the templates can be used to extract triples from previously unseen text.

TRIPLEX-ST is a novel information extraction system that can capture spatio-temporal information from text. It extends current open-domain information extraction (OIE) systems in several dimensions, including the ability to extract facts associated with spatio-temporal contexts (i.e., spatio-temporal information that constrains the facts). The system uses Wikipedia sentences and triples in existing knowledge bases, such as YAGO, to automatically infer templates during a bootstrapping process. These

SUMMARY (Continued)

templates include rich linguistic annotations, and they can be used to extract both facts associated with spatio-temporal contexts and spatio-temporal facts from previously unseen sentences. TRIPLEX-ST also includes syntax-based sentence simplification methods, which contribute to improving extraction effectiveness. Our experiments show that TRIPLEX-ST outperforms a state-of-the-art OIE system on the extraction of spatio-temporal facts. We also show that our approach can accurately extract useful new information, in the form of triples connected to spatio-temporal contexts, using a large Wikipedia dataset.

CHAPTER 1

INTRODUCTION

Deriving useful knowledge from unstructured text is a challenging task. Nowadays, knowledge needs to be extracted almost instantaneously and automatically from continuous streams of information, such as those generated by news agencies or published by individuals on the social web, to update properties related to people, places, and organizations in existing large-scale knowledge bases, such as Freebase (5), DBpedia (2), YAGO (17), or Google's Knowledge Graph (44). Subsequently, these values can be used by search engines to provide answers for user queries (e.g., the resignation date of a given politician, or the ownership after a company acquisition). For many natural language processing (NLP) applications, including question answering, information retrieval, machine translation, and information extraction, it is also important to extract facts from text. For example, a question answering system may need to find the profession for *Michelle Obama* in the fragment *Michelle Obama is American lawyer and writer,* or the location of the *Microsoft Visitor Center* in the sentence *The video features the Microsoft Visitor Center, located in Redmond.*

Open Information Extractors (OIE) aim to extract knowledge in the form of triples from text, with each triple consisting of a subject, a predicate/property that expresses a given relation, and an object. These triples can be expressed via verbs, nouns, adjectives, or appositions. More formally, OIE systems extract triples from an input sentence according to the format `<subject; relation; object>`. In these triples, a relation phrase (i.e., a predicate or property) expresses a semantic relation between the subject and the object. The subject and the object are noun phrases and the relation phrase is a textual fragment that indicates a semantic relation between two noun phrases. The semantic relation can be either verb-mediated or noun-mediated. For example, an extractor may find the triples `<Kevin Systrom;profession;cofounder>` and `<KevinSystrom;appears on; NBC News>` in the sentence *Instagram cofounder Kevin Systrom appears tonight on NBC News.* The first triple is noun-mediated and the second one is verb-mediated. Most OIE systems described in the literature, such as TextRunner (3), WOE (55), ReVerb (14) or Stanford OIE (1) , focus on the extraction of verb-mediated triples. Other OIE systems, such as OLLIE (30), ClauseIE (11), Xavier and Lima's system (57), or ReNoun (60), may also, or only, extract noun-mediated triples from text. OLLIE was the first approach for simultaneously extracting verb-mediated and noun-mediated triples, although it can only capture noun-meditated triples that are expressed in verb-mediated formats.

The research that we conducted in the area of OIE resulted on the development of OIE systems, named TRIPLEX (33) and TRIPLEX-ST (35). TRIPLEX extracts triples from grammatical dependency relations involving noun phrases and modifiers that correspond to adjectives and appositions. TRIPLEX recognizes templates that express noun-mediated triples during its automatic bootstrapping process. Then, it constructs templates from sentences in the bootstrapping set. The templates express how noun-mediated triples occur in sentences and they allow for information to be extracted relating to different levels of text analysis, from lexical (i.e., word tokens) and shallow syntactic features (i.e., part-of-speech tags), to features resulting from a deeper syntactic analysis (i.e., features derived from dependency parsing). Finally, the templates can be used to extract triples from previously unseen text.

We focus on further advancing OIE methods to support the expansion of information in knowledge bases. We perform the realization of experiments focusing on the following problems. The first problem is to extract spatio-temporal facts from textual resources. Factual knowledge is transient and changes over time and space. Thus, it is essential to extract facts from textual resources and to ground these facts in time and space in order to enrich existing knowledge bases. Spatio-temporal triples can also be expressed by either verb-mediated or noun-mediated formats in sentences. We extend the TRIPLEX pipeline to identify and categorize semantic relations between spatio-temporal expressions and named entities in textual resources. We also consider extending the general triples that are extracted from textual sentences, by associating them to their corresponding temporal and/or spatial contexts, when this information is available in the text. For instance, from the sentence *Bob Dylan released Blonde on Blonde in 1966*, the triple `< Bob Dylan; released; Blonde on Blonde>` should be extracted, and this triple should also be associated to the date 1966. Similarly, from the sentence *The PlayStation was released in Japan on 1994, and was released in North America in 1995*, two triples should be extracted: the triple `<PlayStation; released; 1994>` should be extracted and associated to the geospatial region of Japan, whereas the triple `<PlayStation; released; 1995>` should be extracted and associated to the geospatial region of North America.

Finally, we improve OIE results by rewriting complex and long sentences. OIE systems may rely on NLP tools to extract triples. Therefore, splitting complex sentences into several simple sentences can reduce flaws in the output of NLP tools and consequently improve the output of OIE systems. For example the sentence, *The Koeberg Nuclear Power Station, Africa's only nuclear power plant, was inaugurated in 1984 by the apartheid regime and is the major source of electricity for the Western Cape's 4.5 million population* can be simplified to the following sentences *The Koeberg Nuclear Power Station was inaugurated in 1984 by the apartheid Regime*, *The Koeberg Nuclear Power Station is the* *major source of electricity for the Western Cape's 4.5 million population*, and *Koeberg Nuclear Power Station is Africa's only nuclear power plant*. Consequently, simplification techniques can have a strong impact on OIE systems that heavily rely on NLP tools to extract triples.

The main contributions of my thesis are thus as follows:

- We propose a novel OIE method to extract triples from noun phrases, adjectives, and appositions;
- We propose a novel OIE method to extract facts connected to spatio-temporal contexts from text;
- We show how the proposed method can also extract facts involving spatio-temporal information from text;
- We propose a novel method to re-structure complex sentences that include coordinated constituents and/or dependent clauses. Our sentence simplification improves the extraction results of OIE systems (e.g., Stanford OIE (1)) by rewriting complex sentences into a set of simple sentences;
- We evaluate TRIPLEX and TRIPLEX-ST according to the automated framework of Bronzi et al. (7), extending it to assess noun-mediated triples and triples associated to spatio-temporal contexts.

The remainder of the thesis is organized as follows[1]: Chapter 2 is a literature review and we briefly summarize related work in the area of open-domain information extraction. Moreover, we introduce previous work that extract spatio-temporal information from text. The TRIPLEX pipeline is presented

[1] We use English Wikipedia sentences to explain our approach in the next chapters.

in Chapter 3 and it represents two published manuscripts ("Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In ESWC Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD), 2015" and "Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In The SemanticWeb: ESWC Satellite Events, Revised Selected Papers, volume 9341, DOI: $10.1007/978-3-319-25639-9-39$, pages 230243, 2015") for which I was the primary author. Chapter 4 represents a published manuscript ("Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. A Distantly Supervised Method for Extracting Spatio-Temporal Information from Text. In ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), DOI: 10.1145/2996913.2996967, 2016") for which I was the primary author and it describes how Triplex-ST captures spatio-temporal information from text and presents the improvement of OIE methods through rewriting complex sentences. Chapter 4 also represents a series of my own unpublished experiments and it will ultimately be published as a manuscript. Chapter 5 presents the possible directions for future work. Finally, Chapter 6 concludes the thesis and summarizes the main aspects that were discussed.

CHAPTER 2

RELATED WORK

The content of this chapter is a literature review and has been published as “Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In ESWC Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD), 2015”, “Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In The SemanticWeb: ESWC Satellite Events, Revised Selected Papers, volume 9341, DOI: $10.1007/978-3-319-25639-9-39$, pages 230243, 2015”, and “Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. A Distantly Supervised Method for Extracting Spatio-Temporal Information from Text. In ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), DOI: 10.1145/2996913.2996967, 2016” for which I was the primary author.

Current information extraction systems can be used to extract facts from various web documents to construct or enrich the KBs. Open-domain Information Extraction (OIE) systems often depend on distant supervision, using existing information (e.g., triples in the KBs, together with matching sentences), which may be noisy, to generate patterns.

While state-of-the-art OIE systems address to some degree the extraction of static spatio-temporal facts, they have not properly addressed the extraction of the spatial and/or the temporal contexts of facts. For example, some systems connect temporal contexts to facts based upon the document creation or publication (12; 16). However, this assumption may not be true for all facts within a document. For instance, the Wikipedia page that includes the *List of Mayors of Chicago* includes facts whose temporal context is not related to the time of the document creation. In this chapter, we briefly summarize related work in the area of open-domain information extraction. Moreover, we present previous work that consider exclusively the extraction of temporal and/or spatial information from text.

2.1 Open-domain Information Extraction Systems

OIE systems are used to extract triples from text and they can be classified into two major groups. The first group includes systems that extract verb-mediated triples (i.e., TextRunner (3), WOE (55), ReVerb (14), OLLIE (30), or Stanford OIE (1)). The second group includes systems that extract noun-mediated triples (i.e., OLLIE (30), ClauseIE (11), Xavier and Lima's system (57), and ReNoun (60)).

2.1.1 TextRunner

In the first group, the earliest proposed OIE system is TextRunner (3). This system first labels each word in a sentence by a part-of-speech tagger. Then, it detects noun phrases by using a lightweight noun phrase chunker. Afterwards, it finds a sequence of words as a potential relation (i.e., predicate) between each pair of noun phrases. Finally, a classifier judges the trustworthiness of the extracted triples.

2.1.2 WOE

WOE is a self-supervised approach to learn patterns from Wikipedia (55). There are three main components in the WOE architecture: (a) preprocessor, (b) matcher, and (c) learning extractor. The preprocessor segments Wikipedia articles into sentences using OpenNLP[1], and also detects sentences

[1] https://opennlp.apache.org/

which correspond to infobox values for each Wikipedia page. The training data is constructed by the matcher, which loops through all infobox values and their corresponding sentences to find a unique sentence which contains both the subject of Wikipedia articles and infobox values. There are two types of learning extractors. The first one uses features from dependency parse trees. It learns patterns to identify whether the shortest dependency path between two noun phrases shows a semantic relation. The other extractor uses shallow features, such as POS tags.

2.1.3 ReVerb

ReVerb is a verb-based relation extractor that uses syntactic and semantic constraints (14). First, ReVerb finds the longest relation phrase which satisfies syntactic and lexical constraints. If there are overlapped matches, they are merged into a single match. Next, it identifies a pair of noun phrase arguments for each extracted relation phrase. Then, it discovers the closest noun phrase to the left and to the right of a relation phrase. Finally, a confidence score is assigned to each extracted triple by using a logistic regression classifier with shallow syntactic features.

2.1.4 OLLIE

OLLIE, which is a member of both the first and the second groups, was the first approach to extract both noun-mediated and verb-mediated triples (30). It uses high confidence triples extracted by ReVerb as a bootstrapping set to learn patterns. These patterns, mostly based on dependency parse trees, indicate different ways of expressing triples in textual sources. It is important to note that OLLIE only extracts noun-mediated triples that can be expressed via verb-mediated formats. Therefore, it only covers a limited group of noun-mediated triples.

2.1.5 Stanford OIE

Angeli et al. (1) used a small set of patterns and a sentence simplification method prior to extracting triples, this way outperforming OLLIE (30) by extending its coverage. They used a classifier to learn and capture self-contained clauses from complex sentences. Then, they performed natural logic inference over short sentences to extract specific subjects and objects for each extracted triple.

2.1.6 ClauseIE

ClauseIE uses knowledge about the English grammar to detect clauses based on the dependency parse trees of sentences (11). Subsequently, triples are generated depending on the type of those clauses. ClauseIE has predefined rules to extract triples from dependency parse trees and it is able to generate both verb-mediated triples from clauses and noun-mediated triples from possessives and appositions.

2.1.7 Xavier and Lima's System

Xavier and Lima proposed a boosting approach to expand the training set for information extractors so as to cover an increased variety of noun-mediated triples (57). They find verb interpretations for noun-based and adjective-based phrases. Then, the verb interpretations are transformed into verb-mediated triples to enrich the training set. These verb interpretations can create long and ambiguous sentences, and therefore filtering unrelated interpretations is essential before adding the inferred verb interpretations to the training set of information extractors.

2.1.8 ReNoun

ReNoun uses an ontology of noun attributes and a manually crafted set of extraction rules to collect seeds (60). Then, ReNoun generalizes from the extracted seeds via distant supervision to find a much larger training set. Subsequently, the training set is used to learn dependency parse patterns for extracting

	ClauseIE	OLLIE	ReVerb	WOE	TextRunner
Extractions	1050/ 1707	547/1242	388 / 727	447 / 1028	286 / 798
Triple type	verb-mediated	noun-mediated and verb-mediated	verb-mediated	verb-mediated	verb-mediated

TABLE I

EVALUATION RESULTS FOR DIFFERENT OIE SYSTEMS. THE TABLE PRESENTS THE NUMBER OF CORRECT EXTRACTED TRIPLES AND THE TOTAL NUMBER OF EXTRACTED TRIPLES BY EACH OIE SYSTEM.

triples. ReNoun gives a pattern a high score if it extracts triples that have semantically similar attributes, and then assign this score to the facts extracted by the pattern.

2.1.9 Overview

Corro et al. (11) compared ClauseIE with other OIE systems on the same dataset[1] that was originally used to evaluate ReVerb, these results are presented in Table I, which shows the total number of correctly extracted triples as well as the total number of extracted triples for each OIE system.

The same authors also noticed that the performance of OIE systems varies with sentence length. TextRunner, ReVerb, and WOE show promising performance on simple sentences. However, their performance deteriorates on complex sentences. The reason behind this is that long sentences have long relation phrases and shallow syntactic features (e.g., POS tags) cannot capture them. The performance of Renoun, ClauseIE and OLLIE decreases more slowly with respect to sentence length, and dependency parser features are perhaps more useful for handling long sentences.

[1] http://reverb.cs.washington.edu/

TextRunner, WOE, and ReVerb only look for a relation phrase between the subject and object. Thus, they ignore triples related to adjectives, apposition, possessives, and noun phrases. However, OLLIE can detect triples related to adjectives and noun phrases. ClauseIE also extracts triples from appositions. OLLIE does not have any restrictions for the position of relation phrases in the sentences, unlike TextRunner, WOE, and ReVerb. Moreover, OLLIE can find relation phrases outside of argument phrases (subjects and objects).

ReVerb, WOE, and TextRunner have two important problems. On one hand, they use a limited subset of patterns to express relation phrases. ReVerb also has even more limitations due to its set of verbal patterns for capturing relation phrases. On the other hand, they cannot ignore the intervening clauses or prepositional phrases between the relation phrase and the argument phrases since it only uses shallow syntactic features.

2.2 Extration of Spatio-temporal Information from Text

Factual knowledge is transient and changes over time (20), leading to the need for frequently updating KBs. Significant attention has been given to the problem of extracting facts from text to update the KBs accordingly (46; 21). In this context, OIE systems aim to extract knowledge in the form of triples from text. Previously proposed OIE systems (1; 30; 55) use different levels of text analysis to create patterns and capture triples from text, going from lexical (i.e., word tokens) and shallow syntactic features (i.e., part-of-speech tags) to features resulting from a deeper syntactic analysis.While current OIE systems are already quite useful in the extraction of static facts, they still have problems in finding dynamic facts associated with spatio-temporal contexts (35). An essay by Etzioni et al. provides an introductory overview to the area of open-domain information extraction (13).

Some previous studies have considered exclusively the extraction of temporal and/or spatial information from text (49; 53). For instance, two recent joint evaluations were concerned with the extraction of spatial and temporal relations, respectively SpaceEval (39) and TempEval (48). The SpaceEval task concentrated on systems for automatically labeling words and phrases in a sentence with a set of predefined spatial roles (e.g., trajectory, landmark, spatial indicator, distance, and direction). Specifically, the task involved recognizing and classifying the spatial entities in textual sentences that are triggered by spatial expressions (31). TempEval instead addressed the automatic recognition of temporal expressions, events, and temporal relations. In typical systems, temporal entities are first extracted and temporal relations between pairs of entities are then identified and classified. Both these joint evaluation tasks focused on supervised relation extraction approaches, instead of considering open domain approaches. The set of temporal and spatial relations that were considered is closed, and is also relatively small in comparison to the types of relations that can be extracted with OIE systems.

Hoffart et al. described the YAGO knowledge base, where facts can be associated with spatial and temporal contexts (17). YAGO is based on extracting facts from Wikipedia and other similar resources. The extraction method is based on declarative rules to extract facts from infoboxes, lists, tables, and categories in Wikipedia. Moreover, YAGO leverages regular expression patterns to extract facts from Wikipedia text, although these patterns are rather simplistic. Instead, we adapt OIE methods to better deal with temporal and spatial contexts in information extraction from text. In this sense, our work is more closely related to previous research addressing the temporal slot filling task (20; 21) or the temporal scoping of facts (12), although we rely on different methods and address also the spatial contexts of facts. We nonetheless use YAGO3 as a source of distant supervision (4).

Wang et al. made one of the first attempts to extract spatio-temporal facts in the form of triples from textual resources (54). Their approach involves entity extraction and disambiguation modules, and also a fact generation module. Named entities are recognized by regular expressions and are disambiguated through rules. The authors then generate all possible facts from all combinations of disambiguated entities within a sentence. Label propagation and an integer linear program are used to check for constraints in temporal facts and consequently remove noise. It should nonetheless be noted that the types of triples that constitute the focus of TRIPLEX-ST are different (e.g., Wang et al. merely considered events, describing that a person was at a specific time and place (54)). Another difference is that we aim to extract spatial and temporal contexts for many different types of relations.

Derczynski and Gaizauskas (12) introduced a task to associate entities with their temporal contexts if the distance between entities and temporal contexts is less than ten tokens. In contrast, our approach does not impose any constraints on the distance between entities and their spatio-temporal contexts. Garrido et al. (16) use a graph-based document representation to determine links between time expressions and events across multiple documents. Temporal expressions are extracted from text and the document-level metadata, called *Document Creation Time* (DCT). Afterwards, they map the resulting links into five temporal classes. Moreover, they assume that the DCT is within the time-range of the event expressed in the document. However, their assumption is not necessarily true. We particularly focus on extracting spatio-temporal facts and context from text and not from the document metadata. Khan et al. (25) propose a parser to identify spatial relations between recognized places from place descriptions. They use a conditional random field model trained on the manually annotated corpus to extract spatial triples. The set of considered spatial relations in their approach is relatively small. Additionally, their approach only focuses on place descriptions. In contrast, our approach focuses on spatio-temporal facts to enrich knowledge bases.

2.3 Sentence Simplification

Previous studies have already shown that OIE systems can yield a higher accuracy for simplified sentences, and simplification techniques can have a strong impact on OIE systems that heavily rely on NLP tools to extract triples (42; 1; 37).

Most previous works (58; 43; 9) in the area of text simplification was performed within the context of automated document summarization systems. These systems may rewrite input sentences via substitution, reordering, splitting and deletion operations. They often remove the parts of sentences that are less informative. Several recent text simplification approaches (6; 50) do not use any syntactic information, instead using models over sequences of words to classify words as `to-remove` versus `not-to-remove`, with basis on sequences of words from labeled datasets (e.g., gathered automatically through the alignment of sentences from the English Wikipedia and the simple English Wikipedia).

Schemidek and Barbosa (42) proposed a rule-based OIE system that used a sentence simplification method as a pre-processing step before extracting triples. They used syntactic chunking methods, originally proposed by Jurafsky and Martin (24), to split an original sentence into syntactic chunks. Their method then recognizes connections among all chunks in the sentence. According to these connections, some chunks may be joined together to construct a partial sentence. They suggested two different methods to determine the connections between two different chunks. The first method uses a dependency parser to detect a connection between two chunks. The second method determines the connections among chunks based upon a Naïve Bayes classifier. The features used by the classifier are

POS tags of the first and last word of each chunk, the chunk tag (e.g., NP, VP, etc.) and the number of tokens between chunks. They use distant supervision to train the classifier. Finally, they show that re-structuring via the dependency parser is better than using the classifier in the context of improving the accuracy of state-of-the-art OIE systems. However, their approach requires dependency parsing as a preprocessing step before extracting triples. Angeli et al. (1) used a small set of patterns and a sentence simplification method prior to extracting triples, this way outperforming OLLIE (30) by extending its coverage. Niklaus et al. (37) manually defined rules to simplify sentences by analyzing the structure of hundreds of complex sentences from Wikipedia. They wrote context sentences for each prepositional phrase (e.g., spatial or temporal contexts) mentioned in complex sentences. In TRIPLEX-ST, we do not add new sentences for spatio-temporal contexts mentioned in sentences, given that we want to extract facts together with their contexts. We also rewrite dependent clauses or verb phrase conjoint as simple sentences after recognizing their corresponding subjects.

CHAPTER 3

TRIPLEX

The content of this chapter has been published as “Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In ESWC Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD), 2015” and “Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In The SemanticWeb: ESWC Satellite Events, Revised Selected Papers, volume 9341, DOI: $10.1007/978 - 3 - 319 - 25639 - 9 - 39$, pages 230243, 2015” for which I was the primary author.

Preliminary work has resulted on the development of an Open-domain Information Extraction (OIE) system named TRIPLEX, designed specifically to extract triples from noun phrases, adjectives, and appositions. This preliminary work has been described in a paper that was presented at the Know@LOD workshop (34). Systems like OLLIE, which can only extract triples corresponding to relations expressed through verb phrases, can be assisted by TRIPLEX, which extracts triples from grammatical dependency relations involving noun phrases and modifiers that correspond to adjectives and appositions. TRIPLEX recognizes templates that express noun-mediated triples during its automatic bootstrapping process, which finds sentences that express noun-mediated triples by leveraging Wikipedia. Then, it constructs templates from sentences in the bootstrapping set. The templates express how noun-mediated triples occur in sentences and they allow for information to be extracted relating to different levels of text analysis, from lexical (i.e., word tokens) and shallow syntactic features (i.e., part-of-speech tags), to features resulting from a deeper syntactic analysis (i.e., features derived from dependency parsing). In addition, semantic constraints may be included in the templates to obtain more precise extractions. Templates are then generalized to broaden their coverage (i.e., those with similar constraints are merged together). Finally, the templates can be used to extract triples from previously unseen text. We evaluated TRIPLEX according to the automated framework of Bronzi et al. (7), extending it to assess noun-mediated triples.

The remainder of this chapter is organized as follows: The TRIPLEX pipeline is presented in Section 3.1. Section 3.2 describes our experiments, ending with a discussion over the obtained results. Finally, Section 3.3 summarizes the main aspects behind TRIPLEX and presents possible directions for future work.

3.1 Open-Domain Relation Extraction with TRIPLEX

The TRIPLEX approach focuses on noun-mediated triples expressed through noun phrases, adjectives, and appositions. First, TRIPLEX finds sentences that express noun-mediated triples, using information from Wikipedia infoboxes as a kind of distant supervision (32). These sentences are processed with a dependency parser to find grammatical relations between nouns, adjectives, and appositions. Second, TRIPLEX automatically extracts templates from the sentences. Finally, these templates are used to extract noun-mediated triples from previously unseen text.

The TRIPLEX pipeline uses the Stanford NLP toolkit[1] to parse sentences, extract syntactic dependencies, label tokens with named entity (NE) and with part-of-speech (POS) information, and per-

[1] http://nlp.stanford.edu/software/corenlp.shtml

form coreference resolution. The dependency parser discovers the syntactic structure of input sentences, producing a directed graph whose vertices are words and whose edges are syntactic relations between the words. Each dependency edge corresponds to a binary grammatical relation between a governor and a dependent (10). For example, the dependency relations `nsubj<went,Obama>` and `prep-to<went,Denver>` can be found in the sentence *Obama went to Denver*. In the dependency relation `prep-to<went,Denver>`, the word `went` is the governor and the word `Denver` is the dependent. The Named Entity Recognition (NER) model labels sequences of words according to predefined entity categories: *Person*, *Organization*, *Location*, and *Date*. The part-of-speech tagger assigns a morpho-syntactic class to each word, such as noun, verb, or adjective. The coreference resolution module is used to replace, in all the sentences of a given document, pronouns and other coreferential mentions with the corresponding entity spans prior to subsequent processing.

The other components of the pipeline are a noun phrase chunker, which complements the POS, NER, and dependency parsing modules from the Stanford NLP toolkit, WordNet[1] synsets, and Wikipedia synsets. The noun phrase chunker extracts noun phrases from sentences by using an implementation from OpenNLP. WordNet is a lexical database that categorizes English words into sets of synonyms called synsets. WordNet synsets are used to recognize entities within each sentence according to the pre-defined categories, complementing the Stanford NER system. Several synsets are also built for each Wikipedia page. There are different mentions for a Wikipedia page (e.g., redirects and alternative names) that are used in URLs and also in the hypertext anchors that point to a Wikipedia page. For

[1] https://wordnet.princeton.edu/

example, in the Wikipedia page for the *University of Illinois at Chicago*, the word *UIC* is extensively used to refer to the university. Synsets of Wikipedia pages are constructed automatically by using redirection page links, backward links, and hypertext anchors. These links are retrieved using the Java-based Wikipedia Library.[1]

TRIPLEX also uses Wikipedia infobox properties and infobox values during its bootstrapping process. We use a Wikipedia English dump[2] to extract all Wikipedia pages and we query Freebase and DBpedia, according to the Wikipedia page ID, to determine the type of the page. Wikipedia pages are categorized under the following types: *Person*, *Organization*, or *Location*. Additionally, we perform coreference resolution on the extracted Wikipedia pages to identify words that refer to the same Wikipedia page subject. We then use these words to enrich synsets of the respective Wikipedia page. We now describe the TRIPLEX approach for extracting templates, starting with the generation of the bootstrapping set of sentences.

3.1.1 Bootstrapping Set Creation

Following ideas from the OLLIE system (30), our first goal is to construct automatically a bootstrapping set that expresses in multiple ways how the information in noun phrases, adjectives, and appositions is encapsulated. The bootstrapping set is created by processing the extracted Wikipedia pages and their corresponding infoboxes.

[1] https://code.google.com/p/jwpl/

[2] http://dumps.wikimedia.org/backup-index.html

Wikipedia pages without infobox templates are ignored during sentence extraction, while the other pages are converted into sets of sentences. We then perform preprocessing on the sentences from the extracted Wikipedia pages and we use custom templates (i.e., regular expressions) to identify infobox values from the text. In this process, we also convert dates to strings. For instance, the infobox with value 1961|8|4 is translated to August 4, 1961. We begin template extraction by processing 3,061,956 sentences from the extracted Wikipedia pages that are matched with infobox values.

The sentence extractor automatically constructs a bootstrapping set by matching infobox values of the extracted Wikipedia pages with phrases from the text of the corresponding Wikipedia pages. If in a sentence there exists a dependency path between the current infobox value and the synset of the page name, and if this dependency path only contains nouns, adjectives, and appositions, then the sentence is extracted. For instance, given the page for *Barack Obama*, the extractor matches the infobox value *August 4, 1961* with the sentence *Barack Hussein Obama II (born August 4, 1961).* This process is repeated for all infobox values of a Wikipedia page.

In order to match complete names with abbreviations such as *UIC*, the extractor uses a set of heuristics that was originally proposed in the WOE (55) system, named *full match*, *synset match*, and *partial match*. The full match heuristic is used when the page name is found within a sentence of the page. The synset match heuristic is used when one member of the synset for the page name is discovered within a sentence. The partial match heuristic is used when a prefix or suffix of a member of the synset is used in a sentence. Finally, a template is created by marking an infobox value and a synset member in the dependency path of a selected sentence. We apply a constraint on the length of the dependency path between a synset member and an infobox value to reduce bootstrapping errors. This constraint sets the

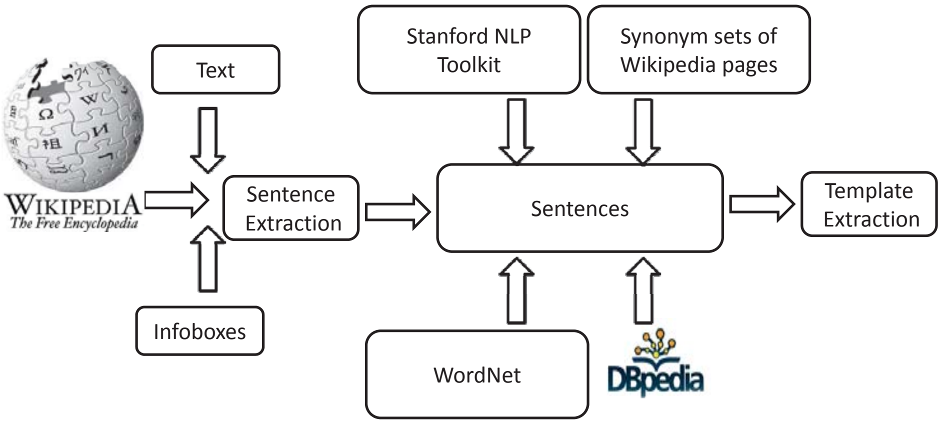


Figure 1. A schematic representation of the TRIPLEX pipeline to extract relation templates.

maximum length of the dependency path to 6, a value which was determined experimentally by checking the quality of our bootstrapping set. Specifically, we randomly selected 100 sentences for manual examination; of these, 90% satisfied the dependency path length constraint.

3.1.2 Template Extraction

After creating the bootstrapping set, the next step is to automatically create templates from dependency paths that express noun-mediated triples. Figure 3 shows TRIPLEX's architecture to extract templates from Wikipedia pages. Templates describe how noun-mediated triples can occur in textual sentences. Each template results from a dependency path between a synset member (a subject) and an infobox value (an object). We annotate these paths with POS tags, named entities, and WordNet synsets. In the template, to each infobox value we add the name of the infobox property. In addition, a template includes a template type, based on the type of the Wikipedia page where the sentence occurred. The types of dependencies between synset members and infobox values are also attached to the template. If there is a copular verb or a verbal modifier in the dependency path, we will add them as a lexical constraint to the template. For example, *headquartered* is a verbal modifier added as a lexical constraint to the corresponding template, in the case of the sentence: *Microsoft is an American corporation headquartered in Redmond* (see Figure 2). *Born* is another lexical constraint for templates related to nationality, as in the sentence *The Italian-born Antonio Verrio was frequently commissioned.* We merge templates if the only differences among them relate to lexical constraints. We keep one template and a list of lexical constraints for the merged templates. Table II shows frequent templates extracted by TRIPLEX.

Infobox members are then cross-referenced with DBPedia properties. Since there is a mapping between infobox names and DBPedia properties, we add the range of the corresponding property, as available in DBPedia, to the templates. For example, the range for the *profession* property is a literal string and the range for *headquarters* is a *Location*, as shown in Figure 2. These constraints aid to avoid errors in our extraction approach.

Infobox values may occur before or after synset members of the page name in sentences. If there exists a dependency path between these values independently of their position, the related template is extracted. For example, the infobox value occurs before the synset member in the sentence *Instagram co-founder Kevin Systrom announced a hiring spree.* In this example, *co-founder* is the infobox value and *Steve Hafner* is the synset member of the Wikipedia page. The infobox value may also occur after the synset member, as shown in the sentence *Microsoft is an American corporation headquartered in*

Template Type	**POS tags** **NER tags** **WordNet tags** **Subject object order**	**Dependency Type**	**Lexical Constraint**	**Property**

Person	`NNP , DT NN`
	`PER O O O`
	`O O O PER`
	`SUBJ O O OBJ`
		apposition	No	Profession

Person	`NNP NN NNP`
	`ORG O PER`
	`O PER O`
	`O OBJ SUBJ`
		nn	No	Profession

Organization	`NN VBN IN NNP`
	`O O O LOC`
	`ORG O O O`
	`SUBJ O O OBJ`
		vmod	Headquartered	Headquarters

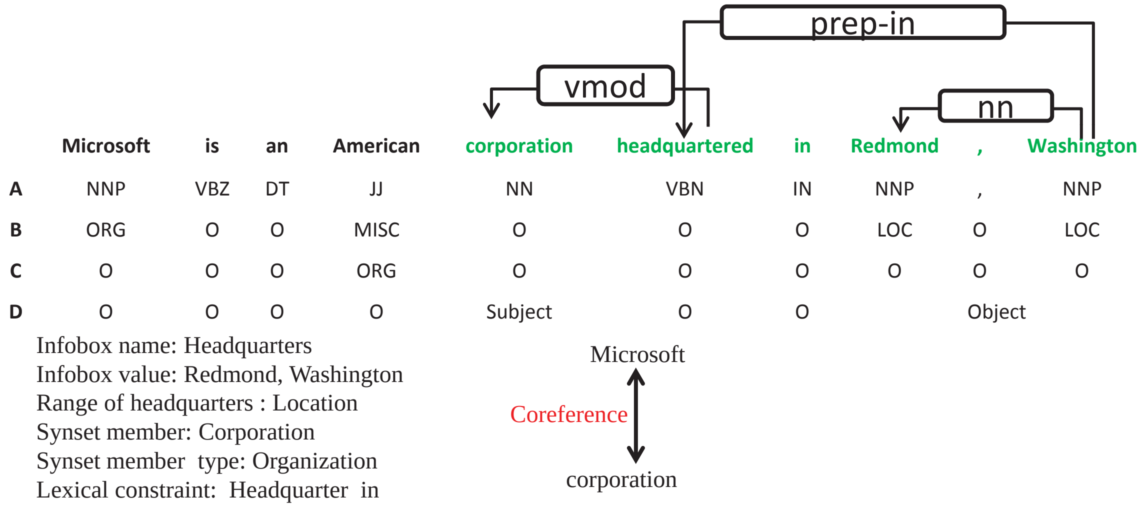
Person	`NNP -LRB-VBN NNP`
	`PER O O DATE`
	`O O O DATE`
	`SUBJ O O OBJ`
		dep	Bear	Birthday

Person	`NN,NNP`
	`O PER`
	`PER,O`
	`OBJ SUBJ`
		nn	No	Title

TABLE II

SAMPLE TEMPLATES EXTRACTED BY TRIPLEX. THESE TEMPLATES INCLUDE SYNTACTIC FEATURES, LEXICAL AND SEMANTIC CONSTRAINTS.

Redmond. In this case, *corporation* is the synset member and *Redmond* is the infobox value (see the example in Figure 2).



	Microsoft	is	an	American	corporation	headquartered	in	Redmond	,	Washington
A	NNP	VBZ	DT	JJ	NN	VBN	IN	NNP	,	NNP
B	ORG	O	O	MISC	O	O	O	LOC	O	LOC
C	O	O	O	ORG	O	O	O	O	O	O
D	O	O	O	O	Subject	O	O		Object	

Infobox name: Headquarters
Infobox value: Redmond, Washington
Range of headquarters : Location
Synset member: Corporation
Synset member type: Organization
Lexical constraint: Headquarter in

A: POS tags B: named entities C: WordNet synsets D: Occurrence of subjects and objects

O: No label PER: person NUM: number ORG: organization

Figure 2. An example sentence annotated with the corresponding dependency relations, POS tags for the word tokens, named entities, WordNet synsets, and occurrences of the synset member of the Wikipedia page (subjects) and the infobox values (objects).

3.1.3 Noun Conjunctions

We also consider some additional heuristics in the template generation process. When there is a conjunction dependency between nouns in a sentence, if one of the nouns has a specific template, the template is expanded to include all the noun relations joined by conjunctions in the sentence. For instance, the template for *lawyer* is expanded to include *writer* in the sentence, *Michelle Obama, an American lawyer and writer, is the wife of the current President of the USA.*

3.1.4 Extension of Dependency Paths

In the context of the OLLIE system, Mausam et al. (30) indicated that lexical and semantic constraints are necessary in templates to extract noun-mediated triples. We extend the dependency path between a Wikipedia synset member and the infobox value when a noun phrase is related to one of the words in the path and when it occurs before or after the dependency path.

3.1.5 Template Generalization:

In order to generalize templates, the POS tags are first converted to universal coarse-grained POS tags (38). Subsequently, similar templates are merged together if they have the same named entity types, WordNet synsets, infobox name, and template type. If there exist neighboring words that have the same named entity type in the dependency path, these words will be considered as a single noun phrase. A module in the pipeline detects these words and merges them and their tags. For instance, the words *Seattle, Washington, U.S.A.*, are considered as a single *Location* phrase. The noun phrase chunker is finally used to search dependency paths and merge words that are part of the same noun phrase chunk. In addition, we do not apply the results of the noun phrase chunker if a synset member and corresponding infobox value occur in the same chunk. For example, *Microsoft*, *co-founder* and *Bill Gates* are in the same noun phrase chunk in the sentence *Microsoft co-founder Bill Gates had his fortune increased.*

3.1.6 Template Matching

This section describes how we use the dependency paths of a sentence together with the extracted templates to detect noun-mediated triples. First, named entities and WordNet synsets are used to recognize the candidate subjects of a sentence together with their types. Then, dependency paths between candidate subjects and all potential objects are identified and annotated by the NLP pipeline. Finally, candidate infobox names (which are properties in DBpedia) are assigned to a candidate subject and a candidate object, derived from matching templates with subject types, dependency types, WordNet synsets, POS tags, and named entity annotations. If there are lexical constraints in a template, the words in the dependency path between a subject and an object must be matched with one of the phrases in the lexical constraint list. We also consider the semantic similarity between the words and the member of the lexical constraint list, using Jiang and Conrath's approach to calculate the semantic similarity between words (22).

When there is a specific range (i.e., *Person*, *Organization*, *Location*, or *Date*) for an infobox name (i.e., property) of a triple, and when the object type of a triple is unknown, a previously trained confidence function is used to adjust the confidence score of the triple. A logistic regression classifier is used in this confidence function after it is trained using 500 triples extracted from Wikipedia pages. Our confidence function is an extension of the confidence function proposed for OLLIE (30) and for ReVerb (14). A set of features (i.e., frequency of the extraction template, existence of particular lexical features in templates, range of properties, and semantic object type) are computed for each extracted triple. The confidence score corresponds to the probability computed by the classifier.

3.1.7 Filtering Candidate Triples

Finally, each candidate triple has an infobox name that is mapped to a DBpedia property. The object type of a candidate triple should also match with the range of that property. When the range of a property is a literal, all possible values of the property are retrieved from DBpedia and compared with the candidate object. If their values are not matched, the candidate triple is discarded.

3.2 Evaluation

We conducted a comprehensive set of experiments to compare the outputs of TRIPLEX against those produced by OLLIE and by ReVerb, leveraging the automated evaluation approach by Bronzi et al. (7). These authors introduced an approach to evaluate verb-mediated information extractors automatically. We improved on their approach by expanding it to the evaluation of noun-mediated triples. Additionally, we compared TRIPLEX, OLLIE, and ReVerb using a manually constructed gold standard. Finally, we compared the various information extractors according to the quality of the produced triples.

We first created a dataset by taking 1000 random sentences from Wikipedia that have not been used during the bootstrapping process. All extracted facts, gathered by the different information extractors from these sentences, needed to be verified.

In their automated evaluation procedure, Bronzi et al. start by noting that a fact is a triple `<subject; predicate; object>` that expresses a relation between a subject and an object. A fact is correct if its corresponding triple can be found in a given knowledge base or if there is a significant statistical association between the entities (subjects and objects) and the predicate on a large collection such as the

Web (7). The authors propose to use the following formula to estimate the precision of an OIE system, which combines correct facts present in existing knowledge bases together with facts that are estimated to be correct, with basis on a strong statistical association between their elements.

$$Precision = \frac{|a| + |b|}{|S|} \tag{3.1}$$

In the equation, $|b|$ is the number of extracted facts by the OIE system that occur in a KB, $|S|$ is the total number of facts extracted by the OIE system, and $|a|$ is the number of correct facts extracted by the OIE system, which have been verified by using the Triple-PMI as defined in Equation 3.2. Since information in knowledge bases is incomplete, Bronzi et al. propose to compute the Triple-PMI to validate extracted facts by the OIE system that do not appear in a knowledge base. We use Lucene to index a large corpus (the entire English Wikipedia and the New York Times collection) and use queries over Lucene to compute the function $\text{Count}(c)$ in Equation 3.2. The higher the Triple-PMI score, the more likely that a fact is correct. In particular, a fact is deemed correct if its Triple-PMI score is above the threshold value of 10^{-2}, which was determined experimentally.

Bronzi et al. used an adapted version of the PMI-IR metric advanced by Turney (47), which in this document we refer to as Triple-PMI, to capture dependency between variables (i.e., text expressions) with basis on occurrence frequency. Although Bronzi et al. called the metric PMI, the adapted metric does not correspond to the well-known point wise mutual information metric used in information theory and statistics. The Triple-PMI of a fact measures the likelihood of observing the fact given that we observed its subject ($\mathtt{subj}$) and object ($\mathtt{obj}$), independently of the predicate ($\mathtt{pred}$):

$$\text{Triple-PMI}(\mathtt{subj}, \mathtt{pred}, \mathtt{obj}) = \frac{\text{Count}(\mathtt{subj} \ \wedge \ \mathtt{pred} \ \wedge \ \mathtt{obj})}{\text{Count}(\mathtt{subj} \ \wedge \ \mathtt{obj})} \tag{3.2}$$

The AIDA system (18), that is used in our NLP pipeline, disambiguates named entities in the evaluation dataset, and thus we can retrieve all possible properties and their values from the knowledge bases for all disambiguated entities. These values are used to verify extracted facts from sentences. The semantic similarity between the properties of those knowledge bases and the predicate of an extracted fact is calculated with the metric proposed by Jiang and Conrath (22). This similarity measure uses WordNet together with corpus statistics to calculate the semantic similarity between concepts. If the semantic similarity is above a predetermined threshold, and if the entities corresponding to the subject and object also match with the knowledge base properties, the fact is deemed correct (7).

The function $\text{Count}(q)$ returns the number of results retrieved by the Google search engine for query q, where the elements of the query occur within the maximum distance of 4 words. The range of the Triple-PMI function is between 0 and 1. The higher the Triple-PMI value, the more likely that the fact is correct. Specifically, a fact is deemed correct if its Triple-PMI value is above the threshold of 10^{-3}, which was determined experimentally.

To compute recall, we can use the procedure described by Equation 3.3, where $|a|$ and $|b|$ are computed as in Equation 3.1. The idea is to estimate the number of existing facts by combining known facts existing in knowledge bases, together with possible that that exhibit a strong statistical association.

$$\text{Recall} = \frac{|a| + |b|}{|a| + |b| + |c| + |d|} \tag{3.3}$$

We now explain how to compute $|c|$ and $|d|$. First, all correct facts within sentences of the dataset are identified. Each fact contains two entities and a relation predicate. All possible entities of a sentence are recognized by Stanford NER, or through exact matching with WordNet synsets. AIDA are also used to normalize the entity names. Moreover, the NLP pipeline identifies all verb phrases (i.e., predicates) in a sentence. Finally, we enlarge the set of predicates in the sentences by adding knowledge base properties.

We can use three sets S, P, and O to generate all the possible facts. These sets are respectively the recognized subjects, predicates, and objects in the sentences. Considering the sentence *Marco Antonio Barrera (born January 17, 1974 in Mexico City) is a retired professional boxer*, the sets S and O of recognized entities contains *Marco Antonio Barrera*, *January 17, 1974*, *Mexico city*, *professional*, and *boxer*. The set P of recognized predicates contains *born*, *is*, and several knowledge base properties. The Cartesian product of these three sets generates all possible facts, $G = (S \times P \times O)$.

Assuming that D is the set of all facts in the different knowledge bases, $|c|$ is computed as follows:

$$|c| = |D \cap G| - |b| \tag{3.4}$$

Finally, $|d|$ is determined by subtracting $|a|$ from the size of the set of all facts in G that are not in D, which have been validated using Triple PMI (that is, those whose PMI score is above the threshold).

We further select 50 sentences from the dataset of 1000 sentences, and a human judge extracts all of the correct facts. Then, we use the method by Bronzi et al. (7) to compute the agreement between the automatic and manual evaluations. The agreement is defined as the ratio between the number of facts where the human and automatic evaluators agree and the total number of facts. The agreement is computed as follows:

$$Agreement = \frac{Number\ of\ facts\ where\ A = H}{Number\ of\ facts} \tag{3.5}$$

This agreement was found to be 0.71. With this information, we are able to determine the precision and recall of our information extractors.

We ran OLLIE, ReVerb, and TRIPLEX individually and then we combined TRIPLEX with OLLIE and with ReVerb. Table XIII shows the obtained results in terms of precision, recall, and the F_1 metric (harmonic mean of precision and recall).

ReVerb only generates verb-mediated triples and OLLIE extracts verb-mediated triples and also noun-mediated triples, if they are expressed in verb-mediated styles. TRIPLEX generates noun-mediated triples and it can complement the results of OLLIE and ReVerb. OLLIE, ReVerb, and TRIPLEX all assign a confidence score to each extracted triple. In these experiments, the extracted triples are only considered if their confidence scores are above a threshold of 0.2. TRIPLEX achieved better results in Table XIII when using the manual evaluation instead of the automatic evaluation because extracted facts with very low PMI scores are considered false in the automatic evaluation. However, these facts are often evaluated as true by a human judge. We also analyzed the errors made by TRIPLEX in the gold standard dataset that was manually annotated. The errors made by TRIPLEX can be classified into two groups: false positives and false negatives. In the gold standard, 65% of the triples are related to verb-mediated triples, which are not extracted by TRIPLEX.

	Automatic evaluation			**Manual evaluation**		
	Precision	Recall	F1	Precision	Recall	F1
ReVerb	0.61	0.15	0.24	0.55	0.11	0.18
OLLIE	**0.64**	0.30	0.40	**0.65**	0.32	0.42
OLLIE*	0.62	0.10	0.17	0.63	0.11	0.18
TRIPLEX	0.55	0.17	0.25	0.62	0.22	0.32
TRIPLEX + OLLIE	0.57	**0.40**	**0.47**	0.63	**0.44**	**0.51**
TRIPLEX + ReVerb	0.58	0.32	0.41	0.55	0.35	0.42

TABLE III

AUTOMATIC AND MANUAL EVALUATION OF INFORMATION EXTRACTORS. OLLIE* ONLY GENERATES NOUN-MEDIATED TRIPLES. THE CONFIDENCE SCORES OF ALL EXTRACTED TRIPLES ARE ABOVE 0.2.

Table XIV shows the results associated with the triples in the gold standard that are not extracted by TRIPLEX. Of those, 10% obtain low confidence scores (false negatives) because the NER module and WordNet could not find the semantic type for the objects. We strongly penalize the confidence score of a candidate triple if its predicate has one particular property type and if no type is detected for the triple's object. For example, the range of the *nationality* property in DBpedia is a *Location* constraint but neither the NER module nor WordNet can recognize a type in the phrase *Swedish writer* or *Polish-American scientist*. Also, 12% of the errors are related to the dependency parser, specifically when the parser could not detect a correct grammatical relation between the words in a sentence. Another 7% of the errors occur when the coreferencing module did not properly resolve coreferential expressions during template extraction. This problem is alleviated by assigning low confidence scores to this group of templates. Finally, 6% of the errors are caused by over-generalized templates. During template generalization, POS tags are substituted by universal POS tags (38). Since some templates only extract

	Missed extractions
10%	No semantic types
12%	Dependency parser problems
7%	Coreferencing errors
6%	Over generalized templates
65%	Verb-mediated triples (outside the of scope for TRIPLEX)

TABLE IV

PERCENTAGE OF THE GOLD STANDARD TRIPLES MISSED BY TRIPLEX.

triples for proper nouns, nouns, or personal pronouns, generalizing and merging these templates together did not produce correct triples.

Approximately 20% of the false positives are in fact correct triples. This stems from the fact that there are few Google search results for queries that contain the subjects of these triples, thus impacting the computation of the PMI scores. Applying the same PMI threshold that is used for prominent subjects proved to be ineffective. For example, triples extracted by TRIPLEX are judged incorrect in the sentence, *Alexey Arkhipovich Leonov (born 30 May 1934 in Listvyanka, Kemerovo Oblast, Soviet Union) is a retired Soviet/Russian cosmonaut*. These triples include information about birth date, birth place, origin, and profession, but they are not available in the gold standard. Other false positives are due to problems resulting from dependency parsing, named entity recognition, chunking, and over generalized templates.

3.2.1 Correct Extractions in TRIPLEX

OIE systems such as ReVerb and OLLIE usually fail to extract triples from compound nouns, adjectives, conjunctions, reduced clauses, parenthetical phrases, and appositions. TRIPLEX only covers noun-mediated triples in sentences, addressing this limitation.

Type	**Percentage of cases**	**Triple category**
Noun-mediated	12%	Conjunctions, adjectives and noun phrases
	9%	Appositions and parenthetical phrases
	6%	Titles or professions
	8%	Templates with lexicon
Verb-mediated	65%	Verb-mediated triples

TABLE V

DISTRIBUTION OF CORRECTLY EXTRACTED TRIPLES FOR TRIPLEX + OLLIE BASED ON THEIR CATEGORIES. THE CONFIDENCE SCORE OF EXTRACTED TRIPLES BY TRIPLEX AND OLLIE IS ABOVE 0.2.

Besides using the automated evaluation that was described in the previous section, we also examined the output of TRIPLEX with respect to a gold standard with human annotations, as shown in Table XVI. The table shows that 12% of the noun-mediated triples are related to conjunctions, adjectives, and noun phrases, meaning that TRIPLEX is also able to extract noun-mediated triples from noun conjunctions. For example, TRIPLEX extracts triples about Rye Barcott's professions from the sentences *Rye Barcott is author of It Happened on the Way to War and he is a former U.S. Marine and cofounder of Carolina for Kibera*. Moreover, TRIPLEX is able to extract triples from appositions and parenthetical phrases, and 9% of the extracted triples are contained within the triple category of appositions and parenthetical phrases. For example, extracted triples from *Michelle LaVaughn Robinson Obama (born January 17, 1964), an American lawyer and writer, is the wife of the current president of the United States* contain Michelle Obama's two professions, birth date, and nationality. Approximaty, 6% of the triples are related to titles or professions. OLLIE is similarly able to capture this kind of triples because they are expressed in a verb-mediated style. However, TRIPLEX does so without using a verb-mediated format. The final fraction of 8% is for noun-mediated triples that rely on the lexicon of noun-mediated templates. For example, the headquarter of Microsoft is extracted from the sentence, *Microsoft is an American multinational corporation headquartered in Redmond, Washington*. Finally, 65% of the extracted triples are verb-mediated triples. Both ReVerb and OLLIE generate verb-mediated triples from sentences. For instance, the triples `<his peers; avoided; attracting attention>` and `<Gotti; became known; as the Dapper Don>` are extracted from the sentence *while his peers avoided attracting attention, Gotti became known as the Dapper Don.* The majority of errors produced by OLLIE and ReVerb are due to incorrectly identifying subjects or objects. ReVerb first locates verbs in a sentence and then looks for noun phrases to the left and right of the verbs. ReVerb's heuristics sometimes fail to find correct subjects and objects because of compound nouns, appositions, reduced clauses, or conjunctions. OLLIE relies on extracted triples from ReVerb for its bootstrapping process and learning patterns. Although OLLIE produces noun-meditated triples if they can be expressed with verb-mediated formats, it does not cover all formats of noun-mediated triples.

3.2.2 Low Recall in OLLIE, ReVerb and TRIPLEX

Finally, we analyzed some sentences to figure out why different information extractors are not able to produce all of the triples in the gold standard. The first reason is that we may not have sufficient information in a sentence to extract triples. For example, TRIPLEX can find the triple `<Antonio; nationality; Italian>` but it cannot find the triple `<Antonio; nationality; England>` in the sentence *The Italian-born Antonio Verrio was responsible for introducing Baroque mural painting into England.* Second, OLLIE and ReVerb cannot successfully extract verb-mediated triples from sentences that contain compound nouns, appositions, parentheses, conjunctions, or reduced clauses. When OLLIE and ReVerb cannot yield verb-mediated triples, recall will be affected because verb-mediated triples are outside of the scope of TRIPLEX. For instance, OLLIE cannot extract any triples from the sentence *Michelle LaVaughn Robinson Obama (born January 17, 1964), an American lawyer and writer, is the wife of the 44th and current President of the United States, and the first African-American....* Improvements to OLLIE and ReVerb could substantially lead to better results in TRIPLEX. Also, improvements to the different NLP components can lead to better precision and recall for information extractors that rely heavily on them.

3.3 Conclusions

This chapter presented TRIPLEX, an information extractor to generate triples from noun phrases, adjectives, and appositions. First, a bootstrapping set is automatically constructed from infoboxes in Wikipedia pages. Then, templates with semantic, syntactic, and lexical constraints are constructed automatically to capture triples. Our experiments found that TRIPLEX complements the output of verb-mediated information extractors by capturing noun-mediated triples. The extracted triples can for instance be used to populate Wikipedia pages with missing infobox attribute values or to assist authors in the task of annotating Wikipedia pages. We also extended an automated evaluation method to include noun-mediated triples, allowing us to effectively test our method, and to compare it against other existing approaches.

CHAPTER 4

TRIPLEX-ST

The content of this chapter has been published as “Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. A Distantly Supervised Method for Extracting Spatio-Temporal Information from Text. In ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), DOI: 10.1145/2996913.2996967, 2016” for which I was the primary author.

Nowadays, we need to discover knowledge instantaneously and automatically from a variety of textual resources to enrich properties in existing knowledge bases (KBs), such as YAGO (17; 4), Freebase(5), DBpedia (2), and Wikidata (52). Existing KBs are mostly constructed by manual curation or by large scale gathering of facts from Wikipedia infoboxes, containing rich information about entities and about their relations. The information in these KBs consists of *facts*, which are triples following the format `<subject; relation; object>`.

Existing KBs already include a large number of spatio-temporal facts that are uniformly valid over time or space, and that are therefore referred to as *static*. These facts include dates of major events (e.g., dates of major battles or birthdates for celebrities), or spatial information related to past events involving people or organizations (e.g., Gene Ammons died in *Chicago*; The Lord of the Rings was filmed in *New Zealand*). This kind of information can be used by innovative spatio-temporal information retrieval services (8; 23).

However, there are other facts that change over time and/or space and are therefore *dynamic*. For instance, soccer clubs can change coaches over time, and new products are often released in different countries at different times. Connecting dynamic facts in KBs to their spatial and/or temporal contexts is thus of great importance, since dynamic facts are only valid inside those contexts. A *temporal context* can be represented using a temporal expression (denoting an instant or time interval). A *spatial context* can in turn be represented as a geospatial region, denoted by a place reference. Preferably, dynamic facts extracted automatically should be connected to a *temporal context* (Ancelotti was the manager of Real Madrid from *2013 to 2015*) and/or with a *spatial context* (the Spring starts in March on the *Northern Hemisphere*), in order to exactly constrain their occurrence.

Current information extraction systems can be used to extract triples from various web documents to construct or enrich KBs (1). For instance, open-domain Information Extraction (OIE) systems capture triples from input sentences, independently of the type of relation. The subject and the object of triples are typically noun phrases, and the relation phrase is a textual fragment that expresses a semantic relation (i.e., a predicate or property) between the subject and the object. The semantic relations in triples can be expressed either by verb phrases (verb-mediated) or by noun phrases (noun-mediated). OIE systems often depend on distant supervision, using existing information (e.g., triples in existing KBs, together with matching sentences), which may be noisy, to generate patterns for extracting triples from text (30; 1). Moreover, most OIE (30; 14; 1) systems are based on symbolic processing (e.g., processing the results of syntactic parsing according to constituents and dependencies). While state-of-the-art OIE systems address to some degree the extraction of static spatio-temporal facts, they have not properly addressed the extraction of the spatial and/or the temporal contexts of facts. For example, some systems connect temporal contexts to facts based upon the document creation or publication date (12; 16). However, this assumption may not be true for all facts within a document. For instance, the Wikipedia page with the *List of Mayors of Chicago* includes facts whose temporal context is not related to the time of the document creation. Moreover, supervised learning models (59; 61), particularly deep learning approaches, have been achieving good results on relation extraction, but these models focus on closed domains and the set of considered relations is relatively small in comparison with OIE systems.

Given that previous OIE systems have not solved the extraction of facts connected to spatio-temporal contexts, the main focus of our research is on extracting this type of spatio-temporal information from text. This information can be used to automatically infer missing facts in KBs, and also to properly constrain facts according to spatio-temporal contexts.

In this Chapter, we specifically describe TRIPLEX-ST, a significant extension to our previous TRIPLEX system (33) since we now extract spatio-temporal information, including both dynamic and static information, about entities and their properties. A preliminary version of TRIPLEX-ST was presented as a poster at ACM SIGSPATIAL 2016 (35). Here, we further detail the proposed method and present, for the first time, the experimental evaluation results. TRIPLEX-ST also advances the general model of triples used in the majority of OIE systems, by considering information about the temporal and/or spatial contexts that qualify the dynamic facts according to the format `<subject; relation; object> (spatial_context; temporal_context)`. Using TRIPLEX-ST, sentences that express triples are first found in Wikipedia text, leveraging a bootstrapping method (19; 30). Then, TRIPLEX-ST uses rich linguistic annotations (e.g., dependency relations, named entities, and lexical constraints), together with information available in existing KBs, specifically YAGO (17; 4), to generate templates.

Templates are used to extract new facts from previously unseen sentences. The templates show how spatio-temporal facts can be expressed in sentences, and also how dynamic facts appear associated with spatio-temporal contexts.

We also propose to improve the extraction results by rewriting complex sentences. The accuracy of OIE systems depends on the output of the NLP tools that they use, and the performance of these tools is deteriorated on complex sentences (i.e., sentences that contain dependent clauses, pronominal references, or coordinated constituents). Through sentence restructuring, we overcome some of the flaws in NLP tools by splitting complex sentences into structurally simpler sentences from which triples are more easily extracted. This rewriting must nonetheless preserve all the relations expressed in the original sentences without modifying their meaning. For example, consider the following sentence; *Zauli was transferred to Bologna for a season and then to Palermo from 2002 till 2005 due to experiencing relegation in 2001.* The previous example can be rewritten into a set of simpler sentences: *Zauli was transferred to Bologna for a season*; *Zauli was transferred to Palermo from 2002 till 2005*; and *Zauli experiencing relegation in 2001.* These three sentence can still capture all original triples: `<Zauli; was transferred; Bologna>`, `<Zauli; was transferred; Palermo> (temporal_context: [2002-2005])` and `<Zauli; experiencing; relegation>`, `(temporal_context: 2001).`

The main contributions of this chapter are thus as follows:

- We propose a novel OIE method to extract dynamic facts connected to spatio-temporal contexts from text;

- We show how the proposed method can also extract static facts involving spatio-temporal information from text;
- We propose a novel method to re-structure complex sentences that include coordinated constituents and/or dependent clauses, improving OIE results;
- We present extensive experimental results, using both automatic and manual evaluation procedures for both static and dynamic facts, together with a detailed error analysis. We have specifically extended the automated evaluation procedure advanced by Bronzi et al. (7) to evaluate triples associated with spatio-temporal contexts.

The rest of the chapter is organized as follows. Section 4.1 describes how TRIPLEX-ST captures semantic relations involving spatio-temporal entities and other noun phrases. Section 4.3 discusses the experimental evaluation of TRIPLEX-ST and the obtained results. We show that TRIPLEX-ST outperforms a state-of-the-art OIE system on the extraction of static spatio-temporal facts, while additionally extending OIE systems to consider dynamic information. Finally, Section 4.4 summarizes our conclusions, while also presenting possible directions for future work in the area.

4.1 Triplex-ST

TRIPLEX, and also the extension named TRIPLEX-ST, includes an offline step of gathering training instances (i.e., sentences matching known facts), followed by template extraction from these instances. Our previously proposed TRIPLEX OIE system (33) only addressed noun-mediated relations associated with the usage of nouns and adjectives (e.g., TRIPLEX could be used to complement the results of other OIE systems focused on verb-mediated relations), but we now extend the system to also directly consider verb-mediated relations when extracting spatio-temporal information. TRIPLEX-ST focuses on spatio-temporal information, including dynamic or static information about entities and their properties. It considers information available in textual sources regarding the temporal and/or spatial contexts that qualify dynamic facts. For instance, the triple `<Microsoft; released; Xbox One;>` `(spatial_context: Japan; temporal_context: February 22, 2002)` should be extracted from the sentence *Microsoft released Xbox One on February 22, 2002 in Japan.* TRIPLEX-ST can also identify whether there are relations between spatio-temporal expressions and named entities in sentences. The subject type or the object type of these triples is either *Location* or *Date*. These triples thus include static information about entities (e.g., persons) and their spatio-temporal properties (e.g., place of birth) as expressed in text.

In TRIPLEX-ST, sentences that express spatio-temporal triples, and optionally also the context of triples, are first discovered using information from Wikipedia infoboxes and example facts from several existing knowledge bases. These sentences are processed with a NLP pipeline, they are transformed through simplification rules, and they are used to detect grammatical relations between spatio-temporal expressions and named entities. Afterwards, templates are generated automatically from the annotated sentences. Finally, facts and their contexts are extracted based upon the templates.

Figure 3 shows the architecture of TRIPLEX-ST to extract templates from Wikipedia pages. TRIPLEX-ST uses Wikipedia infobox values as well as triples in YAGO (4), DBpedia (2), Freebase (5), and Wikidata (52), during its bootstrapping process. We use a recent Wikipedia English dump to extract all Wikipedia pages. Then, we query the different KBs according to the Wikipedia page identifier, to recognize the type of the page. We categorize Wikipedia pages according to the following types: *Per-*

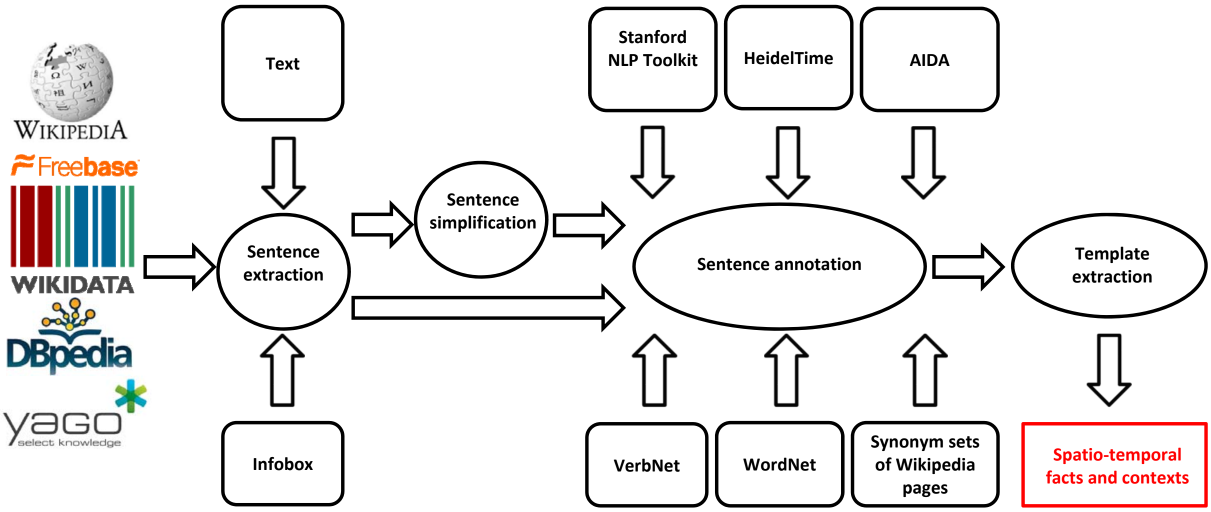


Figure 3. A schematic representation of the TRIPLEX-ST pipeline to infer extraction templates from sentences.

son, *Organization*, *Location*, or *Unknown*. TRIPLEX-ST also uses the Stanford NLP toolkit[1] to parse and chunk sentences from the text of Wikipedia pages, extract syntactic dependencies and constituent parse trees, label tokens with named entity (NE) and with part-of-speech (POS) information, and resolve coreferences.

Specifically, the Stanford dependency parser discovers the syntactic structure of input sentences, producing a directed graph whose vertices are words and whose edges are syntactic relations between words (10). We use dependency relations to detect dependent clauses and coordinating constituents. For example, a relation `nsubj<played, Bergkamp>` exists between the governor word `played`

[1] http://nlp.stanford.edu/software/corenlp.shtml

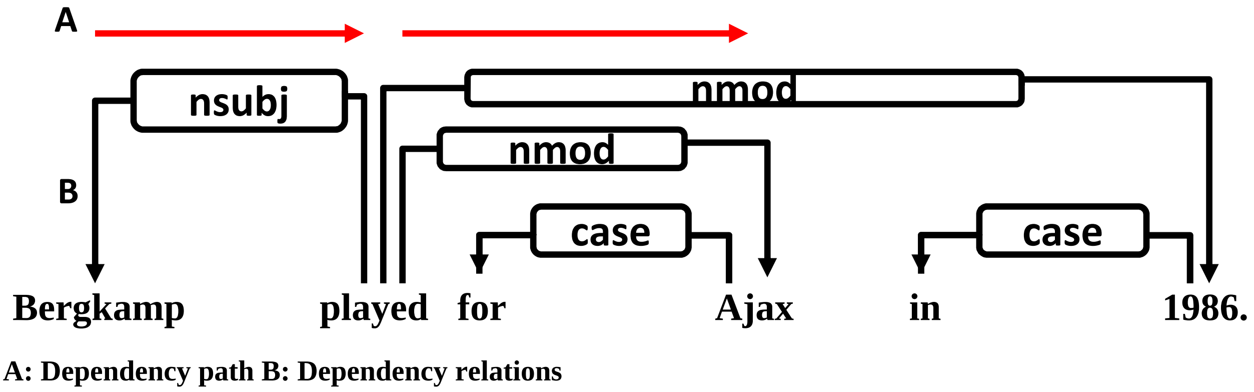


Figure 4. The dependency path between the words Bergkamp and Ajax.

and the dependent word Bergkamp, in the example from in Figure 4. TRIPLEX-ST also uses the resulting dependency paths during the bootstrapping process. A dependency path connects words of a sentence by using dependency relations. In Figure 4, the arrows show the dependency path between the word Bergkamp and the word Ajax using the relations nsubj<played, Bergkamp> and nmod<played, Ajax>.

Sentences are also labeled by a Named Entity Recognition (NER) model according to pre-defined entity categories: *Person*, *Organization*, *Location*, and *Date*. A morpho-syntactic class (e.g., noun, verb, etc.) is also assigned to each word by a part-of-speech tagger. We use the constituency tree produced by the Stanford parser to identify noun phrases and verb phrases. Finally, the coreference resolution module substitutes all the pronouns and other coreferential mentions, in the sentences of a given document, with the corresponding entity spans prior to subsequent processing.

We also complement the results from the Stanford NLP toolkit with those from other tools (i.e., the HeidelTime temporal expression resolver (45), and the AIDA system for named entity disambiguation (18)), and through resources such as WordNet[1] and VerbNet.[2] The HeidelTime temporal expression tagger complements the results from Stanford NER, also classifying (e.g., according to classes such as date, time, duration, or set) and normalizing the recognized temporal expressions. The AIDA system disambiguates entities recognized by Stanford NER into the corresponding concepts in YAGO, which are in turn associated with concepts in other KBs. This may be useful for integrating the extracted facts with existing KBs although, in our case, entity disambiguation is only used to support the evaluation procedure.

TRIPLEX-ST also uses WordNet to enrich the annotations provided by the NLP pipeline. WordNet is a lexical database that classifies English words and phrases into sets of synonyms called synsets. Entities within each sentence are matched against WordNet synsets and classified according to the predefined categories. VerbNet is in turn a hierarchical verb lexicon. It includes syntactic and semantic information for English verbs, derived from Levin's classification (29) of verbs according to syntactic and semantic properties. Each VerbNet class contains a set of syntactic frames and semantic restrictions. Syntactic frames consist of the actual verbs, thematic roles (e.g., agent, theme, and location), required prepositions, and any other lexical items required to alternate or construct the associated verb phrases. Moreover, semantic restrictions (e.g., *person* and *organization*) are used to show the preference

[1] https://wordnet.princeton.edu/

[2] https://verbs.colorado.edu/ mpalmer/projects/verbnet.html

of thematic roles (26). Verbs occurring in textual sentences are matched against VerbNet and classified accordingly. The NLP pipeline also creates several synsets for each Wikipedia concept. We use the different mentions (e.g., redirects and alternative names) that are associated with Wikipedia URLs, and also the hypertext anchors that point to a Wikipedia page. For instance, the acronym *WTO* is extensively used in the Wikipedia page for the *World Trade Organization*. We specifically use redirection page links, backward links, and hypertext anchors to automatically construct synsets of Wikipedia pages. The Java-based Wikipedia Library[1] is used to retrieve these links.

4.1.1 Bootstrapping Set Creation

We automatically build two bootstrapping sets following the ideas from the previous OLLIE OIE system (30) and also from TRIPLEX (33). These sets involve sentences that are extracted from Wikipedia text, expressing dynamic facts connected to spatio-temporal contexts, or static spatio-temporal facts. We use the original Wikipedia sentences in the creation of bootstrapping sets, and also simplified versions of these sentences. The use of syntactic information for sentence simplification is explained on Section 4.2. Specifically regarding the bootstrapping sets:

- We use the YAGO facts connected to a spatial or a temporal context, together with Wikipedia pages, to construct a bootstrapping set of sentences that express dynamic facts connected to spatio-temporal contexts.
- We process KBs (e.g., the Wikipedia pages and their relevant infobox values), together with Wikipedia pages, to construct the bootstrapping set of sentences for static spatio-temporal facts.

[1] https://dkpro.github.io/dkpro-jwpl/

Custom templates (i.e., regular expressions) are used to recognize infobox values, the objects of facts, or the spatio-temporal contexts present in text. In the specific case of temporal expressions appearing in the text, they are normalized and classified by the NLP pipeline. These normalized values are mapped to particular classes (e.g., instant, partial, and interval) and we rely on the Joda-Time library[1] while matching temporal expression with text.

In dynamic facts, the same subject can appear with multiple objects, or the same object can appear with multiple subjects, according to the spatial and/or the temporal context. For example, *Michael Phelps* won six gold medals in Athens and he won five medals in Rio de Janeiro; *Angelina Jolie* married actor Billy *Bob Thornton* in 2000 and she married *Brad Pitt* in 2004. We discovered around 700,000 instances in the set of YAGO facts that are connected to temporal contexts. YAGO has therefore a good coverage of dynamic facts connected to temporal contexts, although it is very poor in terms of facts connected to spatial contexts. We nonetheless used the `isLocatedIn` relation in YAGO to find the spatial contexts of objects within dynamic facts. For example, *Katri Mattsson* played for *PK-35 Vantaa* in Finland and she played for *LSK Kvinner FK* in Norway. We discovered around 1,600 dynamic facts connected to a spatial context, and we also have a total of 800 facts with both spatial and temporal contexts. These instances constitute the source data of our bootstrapping method for the extraction of dynamic facts.

Next, the sentence extractor matches phrases from the text of the Wikipedia pages with the corresponding dynamic facts and also with their spatio-temporal contexts. We simplify complex sentences

[1] http://www.joda.org/joda-time/

that include coordinated constituents and/or dependent clauses. We consider templates generated from both the original and simplified versions during sentence extraction. If there exist dependency paths connecting an object of a fact, the spatio-temporal context, and the synset of the Wikipedia page in a sentence, then the sentence is extracted and added to the bootstrapping set. For example, given the page for *Dennis Bergkamp*, the sentence extractor matches a fact connected to the temporal context (i.e., `<Bergkamp; Played for; Ajax> (temporal_context: 1986)`) with the sentence *Bergkamp played for Ajax in 1986*. Notice that there exists a dependency path associating the object *Ajax*, the temporal context *1986*, and the synset word *Bergkamp* in Figure 6, according to the dependency relations. Thus, the sentence could be added to the bootstrapping set of dynamic facts. As another example, given the page for *Bhutan*, the extractor matches the triple associated to the temporal context `<Bhutan; population; 770,000;> (Temporal context: [2015])` with the sentence *Bhutan had a population of 770,000 People in 2015*. Notice that there exists a dependency path connecting the object token `770,000`, the temporal context `2015`, and the synset word `Bhutan` in Figure 7. Thus, the sentence *Bhutan had a population of 770,000 People in 2015* could be added to the bootstrapping set of dynamic facts.

In the case of the bootstrapping set for static spatio-temporal facts, we perform a similar process over the collection of Wikipedia pages, if their type is *Location*, or if the type of their infobox values is either *Location* or *Date*. A sentence is added to the bootstrapping set if there exists a dependency path between an infobox value and the synset of the Wikipedia page in a sentence. For example, given the page for *Flacq District*, the extractor matches the infobox value `297.9` km^2 with the sentence *Flacq District has an area of 297.9* km^2. Since, there exists a dependency path between the infobox value `297.9` km^2 and the synset `Flacq District`, the sentence is added to the bootstrapping set of static facts.

We also apply a constraint on the length of the dependency path between a synset member and an infobox value (an object) to reduce bootstrapping errors. This constraint sets the maximum length of the path to 8, a value that was determined experimentally by examining the quality of both bootstrapping sets.

We categorize the sentences in both bootstrapping sets into semantic groups according to their corresponding infobox names or properties (e.g., *birth date*). In the case of verb-mediated triples, there are frequently noisy sentences in the groups, which do not express the core meaning of triples. For example, the triple `<Obama; birthplace; Honolulu>` is found by the original TRIPLEX extractor in the sentence *Obama was born in Honolulu*, and also in the sentence *Obama visited Honolulu.* The first sentence is a true positive and the second one is a false positive (noise) in the bootstrapping set of the property *birthplace*. In the example, the word *birthplace* is a property name and both *visited* and *born in* are called relations. Relations are extracted from sentences that express a fact (i.e., sentences that contain the subject and object of triples) and they include all words in the dependency path between a subject and an object. Most relations are verb phrases and verbs do indeed play a significant role in conveying the core meaning of facts in sentences. Thus, we need to remove noisy sentences associated with verb-mediated triples from the bootstrapping sets, before the step of template extraction.

In previous work, Intxaurrondo et al. (19) proposed several methods to remove noisy sentences automatically from a bootstrapping set. We use their heuristics in TRIPLEX-ST.

The first heuristic computes the Triple-PMI in between the entities (i.e., the subject and the object) of a fact and its corresponding labels, as extracted from a sentence. The Triple-PMI is explained in Equation 3.2. Consequently, noisy sentences are detected and removed from the bootstrapping sets. For example, the sentence *Gene Siskel was an American film critic and journalist for the Chicago Tribune* is excluded from the bootstrapping set of the property *residence*.

Intxaurrondo et al. (19) also proposed a method to compute the centroid of all labels for each property in a bootstrapping set. They believe that the noisy labels of a property are far from the centroid of the cluster of labels for that property. We remove stop words from labels and also apply a stemming algorithm. Each dimension of the vector for a label corresponds to a word, where the occurrence frequency of each word is used as a feature. A vector of the form $\vec{V}_{i_n} = (W_{1_n}, W_{2_n}, \ldots, W_{K_n})$ is thus used to represent a label n for a property i, and we consider K to be the number of distinct words in the labels of property i, and W_{j_n} to be the number of appearances of word j $(1 \leq j \leq K)$ in label n of property i. We then use the formula in Equation 4.1 to calculate the centroid $\vec{P}_i$ of all labels for property i:

$$\vec{P}_i = \left(\frac{\sum\limits_{\forall n \in labels_i} \vec{V}_{i_n}}{|labels_i|} \right) \tag{4.1}$$

In the equation, $labels_i$ is the set of all labels for property i, with $1 \leq i \leq M$ and where M is the number of properties in the bootstrapping sets. The standard cosine similarity metric compares the centroid of all labels for property i and the vector of label n for property i. We keep the top 85% of the most similar labels to the centroid for each property, and discard the rest. For example, we compute the centroid of labels for the property *birthplace*. The label *born in*, appearing in the sentence *Samuel Loyd, born in Philadelphia and raised in New York, was an American chess player*, is the most similar label to the centroid of labels for the property *birthplace*. On the contrary, the label *was the King of Aragon of*, which is extracted from the sentence *Peter the Great was the King of Aragon of Valencia and Count of Barcelona from 1276 to his death*, is far from the centroid of labels for the property *birthplace* and is thus removed from the bootstrapping set.

4.1.2 Inferring Extraction Templates

After removing noise from the bootstrapping sets, the next step is to infer templates from the sentences in both bootstrapping sets (i.e., the ones for static and dynamic facts). These templates show how the spatio-temporal contexts of triples, or how static spatio-temporal facts, can be expressed in textual resources, and they will later be used to extract new facts from text.

When considering dynamic facts, the templates include the shortest dependency paths that connect the synset member (i.e., a subject), an infobox value (i.e., an object), and also the phrases that correspond to the spatial and/or temporal contexts of the fact. When considering static spatio-temporal facts, the templates include the shortest dependency path between a synset member and an infobox value. For example, Figure 5 shows the template extracted from a sentence associating the object *Catholic churches*, the spatial context *Dublin* , and the synset *Patrick Byrne*.

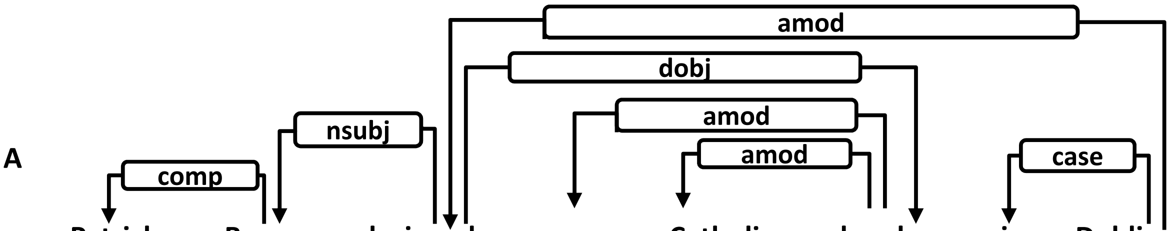
The word tokens involved in the dependency paths are annotated by POS tags, named entity types, and WordNet synsets. VerbNet is also used to add syntactic frames and semantic restrictions for verbs in the dependency path of the template. Moreover, a type is attached to the template. This type is determined by the type of the Wikipedia page where the sentence occurred. TRIPLEX-ST only uses templates whose types are the same as their synset member types. The dependencies between synset members and infobox values are also added to the template. If there is an adjective modifier, a verbal modifier, or a noun in this dependency path, we add them as lexical constraints to the template. For example, *populous* is an adjective modifier that is added as a lexical constraint to the template that is generated from the sentence *Anaheim is the 10th most populous city in California*.

Since there is a mapping between infobox names and properties in the considered KBs, we add the domain of the corresponding property in the KBs to the templates. For example, the domain of the property *nationality* is a *Location*, and the domain of the property *population rank* is an *Integer*. Moreover, we add the domain of spatio-temporal contexts to the templates, in case the templates involve contexts. The domain of a spatial context is a *Location*, and the domain of a temporal context is a *Date*. These constraints help in avoiding errors while extracting new facts.

Synset members of the Wikipedia page name may happen before or after infobox values in sentences. If there exists a dependency path between these values, independently of their position, the relevant template is extracted. For example, the synset member may occur before the infobox value, as shown in Figure 6. In this case, the word *Bergkamp* is the synset member and the word *Ajax* is the infobox value.

As another example, the synset member may also occur before the infobox value, as shown in the sentence *Bhutan had a population of 770,000 People in 2015*. In this case, the word *Bhutan* is the synset member and the token *770,000* is the infobox value (see Figure 7).

We also extend the dependency path between a Wikipedia synset member and an infobox value (object) in the bootstrapping sets if there exists a noun phrase related to one of the words in the path, and if it occurs before or after the dependency path. For example, the dependency path between *Bhutan*



	Patrick	Byrne	designed	many	Catholic	churches	in	Dublin
B	NOUN	NOUN	VERB	ADJ	ADJ	NOUN	ADP	NOUN
C	PERSON	PERSON	O	O	O	O	O	LOC
D	O	O	O	O	O	O	O	O
E	SUBJ	O	O	O	OBJ	OBJ	O	O
F	O	O	O	O	O	O	O	O
G	O	O	O	O	O	O	O	LOC

Infobox name or property: Created	**Infobox value : Catholic churches**
Range of population: -	**Synset member: Patrick Byrne**
Synset member type: PERSON	**Spatial context: LOCATION**
Temporal context: -	**VerbNet Frame: NP V NP**

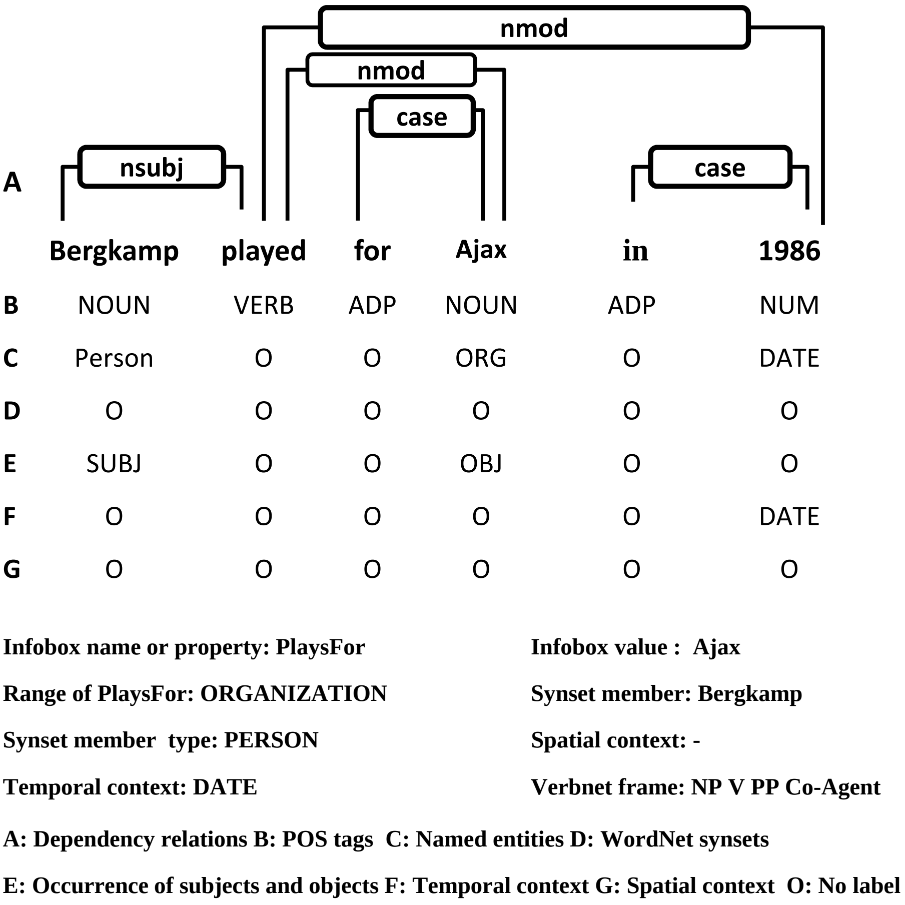
A: Dependency relations B: POS tags C: Named entities D: WordNet synsets LOC: LOCATION

E: Occurrence of subjects and objects F: Temporal context G: Spatial context O: No label NUM: Number

Figure 5. An example template resulting from a sentence annotated by the NLP pipeline, for the property *Created*.

and *770,000* is expanded by adding the noun phrase *people* in the sentence *Bhutan had a population of 770,000 people in 2015*.

In order to generalize the templates, the POS tags are first converted to universal coarse-grained POS tags (38). Moreover, we use VerbNet to obtain syntactic frames and semantic restrictions of verbs in dependency paths of templates. Similar templates are merged together if they have the same named



A	Bergkamp	played	for	Ajax	in	1986
B	NOUN	VERB	ADP	NOUN	ADP	NUM
C	Person	O	O	ORG	O	DATE
D	O	O	O	O	O	O
E	SUBJ	O	O	OBJ	O	O
F	O	O	O	O	O	DATE
G	O	O	O	O	O	O

Infobox name or property: PlaysFor	**Infobox value : Ajax**
Range of PlaysFor: ORGANIZATION	**Synset member: Bergkamp**
Synset member type: PERSON	**Spatial context: -**
Temporal context: DATE	**Verbnet frame: NP V PP Co-Agent**

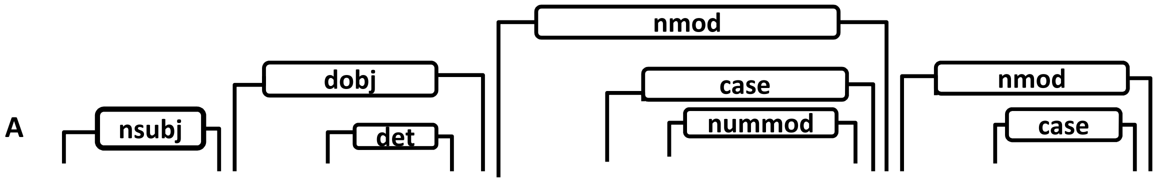
A: Dependency relations B: POS tags C: Named entities D: WordNet synsets

E: Occurrence of subjects and objects F: Temporal context G: Spatial context O: No label

Figure 6. An example template resulting from a sentence annotated by the NLP pipeline, for the property *PlaysFor*.

entity types, WordNet synsets, infobox names or properties, syntactic frames and semantic restrictions of verbs, temporal context types, spatial context types, and template types. We keep one template and a list of lexical constraints for the merged templates.

If there exist neighboring words (i.e., words appearing in consecutive positions in the text) that have the same named entity type in the dependency paths, these words will be considered as a single noun



	Bhutan	had	a	population	of	770,000	People	in	2015
B	NOUN	VERB	DET	NOUN	ADP	NUM	NOUN	ADP	NUM
C	LOC	O	O	O	O	NUMBER	O	O	DATE
D	LOC	O	O	O	O	O	O	O	O
E	SUBJ	O	O	O	O	OBJ	O	O	O
F	O	O	O	O	O	O	O	O	DATE
G	O	O	O	O	O	O	O	O	O

Infobox name or property: Population Infobox value : 770,000

Range of population: Integer number Synset member: Bhutan

Synset member type: LOCATION Spatial context: -

Temporal context: DATE VerbNet Frame: NP V NP

A: Dependency relations B: POS tags C: Named entities D: WordNet synsets LOC: LOCATION

E: Occurrence of subjects and objects F: Temporal context G: Spatial context O: No label NUM: Number

Figure 7. An example template resulting from a sentence annotated by the NLP pipeline, for the property *Population*.

phrase. A module in the pipeline detects these words and merges them and their tags. For instance, the words *Seattle, Washington, U.S.A.*, are considered as a single *Location* phrase. Moreover, we use a noun phrase chunker to search dependency paths and merge words that belong to the same noun phrase chunk.

4.1.3 Template Matching

When extracting information from a new document, the NLP pipeline first annotates an input sentence. We also apply our sentence simplification method on complex sentences (see Section 4.2). We consider the original sentences and their simplified versions during template matching. Then, we match the dependency parse of the sentence with the dependency paths of templates to identify the candidate subjects and objects. Afterwards, infobox names (i.e., properties) of templates are assigned to a candidate subject and a candidate object, derived from matching templates with subject types, object types, dependency types, WordNet synsets, POS tags, syntactic frames and semantic restrictions of verbs in dependency paths, temporal context types, spatial context types, and named entity annotations. If multiple templates match a sentence, we use the most frequent templates and combine their extraction results.

If the pipeline recognizes a temporal context and/or a spatial context for a fact, and if the type of the recognized context is matched with the type of the context of templates, then the recognized context is also attached to the triple. If the spatio-temporal context of a template is not exactly matched, the pipeline extracts a triple without the corresponding context. If there are lexical constraints in a template, we use Jiang and Conrath's approach (22) to calculate the semantic similarity between the words in the dependency path and members of the lexical constraint list, during template matching. We consider a similarity threshold of 0.7 in our experiments. For instance, TRIPLEX-ST detects the candidate subject *David Beckham* and the candidate object *LA Galaxy* after matching the dependency path of the sentence *David Beckham played for LA Galaxy in 2007* with the template in Figure 6. Afterwards, TRIPLEX-ST annotates the dependency path between the candidate subject, the candidate object, and spatio-temporal expressions. Since all annotations of the dependency path are matched with the template in Figure 6, the dynamic fact `<David Beckham; played for; LA Galaxy> (temporal_context: 2007)` is captured from the sentence. Notice that TRIPLEX-ST merges the words *David Beckham* and their tags before template matching. A similar merging is also performed on the words *LA Galaxy*.

As another example, the candidate subject `Iran` and the candidate object `65 million` are recognized after matching the dependency path of the sentence *Iran had a population of 65 million people in October 2010* with the template in Figure 7. The spatio-temporal expressions in the sentence are recognized by the NLP pipeline. Then, the dependency path between the candidate subject, the candidate object, and spatio-temporal expressions are annotated by TRIPLEX-ST. Since all annotations of the dependency path are matched with the template in Figure 7, the dynamic fact `<Iran; had the population; 65 million> (Temporal context: [October : 2010])` is extracted from the sentence.

We also use a priori information about infobox names (properties) and templates to assess the correctness of the extracted triples. If there is a specific domain (i.e., *Person*, *Organization*, *Location*, or *Date*) for a template and if the object type of a triple is unknown, the pipeline adjusts the confidence score of the extracted triple through a previously trained confidence function.

For the confidence function, we specifically pre-trained a logistic regression classifier using 500 manually labeled dynamic facts extracted from Wikipedia pages, taking inspiration on ideas from OLLIE (30). We use a single function to assess the correctness of dynamic facts together with their spatio-temporal contexts. The confidence score corresponds to the probability computed by the classifier. The set of features used by the classifier in the case of triples with spatio-temporal contexts is shown in

Features for the confidence classifier	**Dynamic**	**Static**
Temporal context type	✓	
Spatial context type	✓	
An indicator that if checks the length of the dependency path between the spatio/temporal context and the object is <10 words	✓	
An indicator that checks if the length of dependency path between the spatio/temporal context and the subject is <10 words	✓	
Existence of prepositions before the spatio-temporal context	✓	
Existence of a coordinating conjunction between the object and the spatio-temporal context	✓	
Existence of a coordinating conjunction between the subject and the spatio-temporal context	✓	
Frequency of the extraction template	✓	✓
Existence of particular lexical features	✓	✓
Range of properties	✓	✓
Semantic subject type	✓	✓
Semantic object type	✓	✓

TABLE VI

FEATURES IN THE CLASSIFIERS THAT ARE USED FOR ASSIGNING A CONFIDENCE SCORE TO EXTRACTED FACTS.

Table VI, including features derived from syntactic dependency relations, named entities, and lexical constraints. The set of features for static spatio-temporal triples is also shown in Table VI.

4.2 Handling Complex Sentences

The performance of NLP and IE tools is often deteriorated on complex sentences that include coordinated constituents and/or dependent clauses. Thus, rewriting and simplifying complex sentences can improve the accuracy of OIE tools.

In TRIPLEX-ST, we propose to use a syntactic simplification method as a pre-processing step to improve the extraction performance. We identify complex sentences and convert them into sequences of shorter and structurally simpler sentences. Additionally, we apply our syntactic simplification method on complex sentences of the bootstrapping sets. We process both the original sentences and their simplified versions, during the stage of inferring extraction templates, and also during template matching.

4.2.1 Sentences with Dependent Clauses

Complex sentences often contain at least one dependent clause and one independent clause. In turn, an independent clause contains a subject and a verb phrase, and it does not contain any dependent clauses. A dependent clause either modifies an independent clause or serves as another component of a sentence. Noun clauses, relative clauses, and adverbial clauses are different types of dependent clauses. For example, the sentence in Figure 8 includes two dependent clauses, namely *After finishing runner-up seven times in 2002* and *winning eight consecutive titles in the 400m dash*. A dependent clause may not contain an explicit subject and it can inherit its subject from its independent clause. One of our sentence simplification goals is to find the subjects of dependent clauses and then rewrite them as simple sentences.

We use the following dependency relations to detect dependent clauses in complex sentences: adverbial clause modifiers, relative clause modifiers, clausal complements, and clausal modifiers of a noun. Afterwards, we use the constituent parse tree of a sentence to find the smallest subtree that represents the sentence and includes a dependent clause. We then identify the boundaries of the dependent clause based upon the leaves of the subtree. We use the following dependency relations to detect dependent clauses in complex sentences: adverbial clause modifiers, relative clause modifiers, clausal comple-

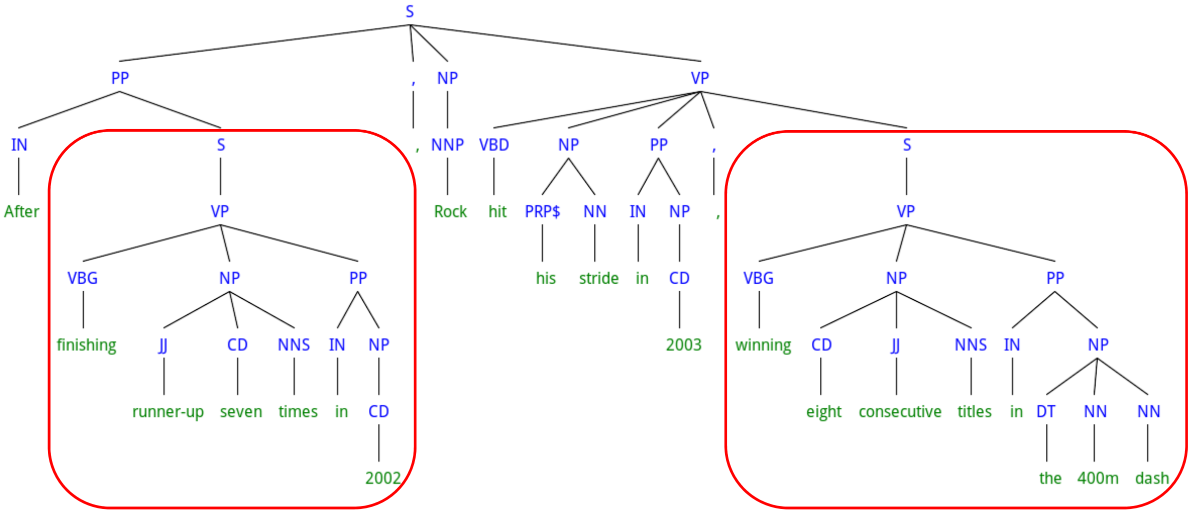


Figure 8. Two dependent clauses in the constituent parse tree for a given example sentence.

ments, and clausal modifiers of a noun. Afterwards, we use the constituent parse tree of a sentence to find the smallest subtree that represents the sentence and includes a dependent clause. We then identify the boundaries of the dependent clause based upon the leaves of the subtree. The constituent parse tree and dependency paths are used to construct a set of candidate subjects for each dependent clause. The coreference resolution module of the NLP pipeline may substitute pronouns of sentences prior to identifying subjects. A candidate subject is typically a noun phrase or a pronoun that is identified by using the constituent parse tree. A candidate subject is valid if there exists a dependency path between the candidate subject and the verb phrase of a dependent clause. Candidate subjects may occur before or after a dependent clause in complex sentences, as shown in Figure 8. For example, the set of candidate subjects, in Figure 8 includes *Rock*, *stride*, and *eight consecutive titles* for the dependent clause *finishing runner-up seven times in 2002*. Next, we use a pre-trained classifier to recognize the correct subject of a dependent clause from the set of candidate subjects. The classifier is trained by using a distant supervision method, as will be explained in Section 4.2.3. For example, *Rock* is the identified subject for the two dependent clauses in Figure 8. Finally, we capture the two sentences (e.g., *Rock finishing runner-up seven times in 2002* and *Rock winning eight consecutive titles in the 400m dash*) from the dependence clauses in Figure 8. From the simplified sentences, TRIPLEX-ST is then able to extract the triples `<Rock; finished; runner-up seven times> (temporal_context: 2002)` and `<Rock; winning; eight consecutive titles>`.

4.2.2 Compound Sentences

A compound sentence consists of multiple independent clauses with no dependent clauses, joined together using conjunctions and/or punctuation. Coordinating conjunctions are words that join phrases, clauses, and sentences, connecting noun and verb phrases that have a similar grammatical structure in sentences, called *conjoints*. In Figure 9, there exist two noun phrase conjoints i.e., *Saudi Arabia (1990)* and *Afghanistan (2007)*. Moreover, two or more verb phrases can also be connected by a coordinating conjunction. In Figure 10, there are two verb phrase conjoints (i.e., *won election to the Ohio Senate in 2008* and *began his first term in 2009*) that are joined by a coordinating conjunction.

The dependency parser detects coordinating conjunctions and conjoints in sentences by using the dependency relations *conjunct* and *coordination*. Then, the boundaries of each conjoint can be identified by processing the constituent parse tree of a sentence. Conjoints that are noun phrases need to be semantically and syntactically similar. If a sentence includes noun phrase conjoints, we rewrite the sentence into a set of simpler sentences. Each simple sentence only contains one noun phrase conjoint. For example, the sentence in Figure 9 can be written into the following simple sentences: *McHale*

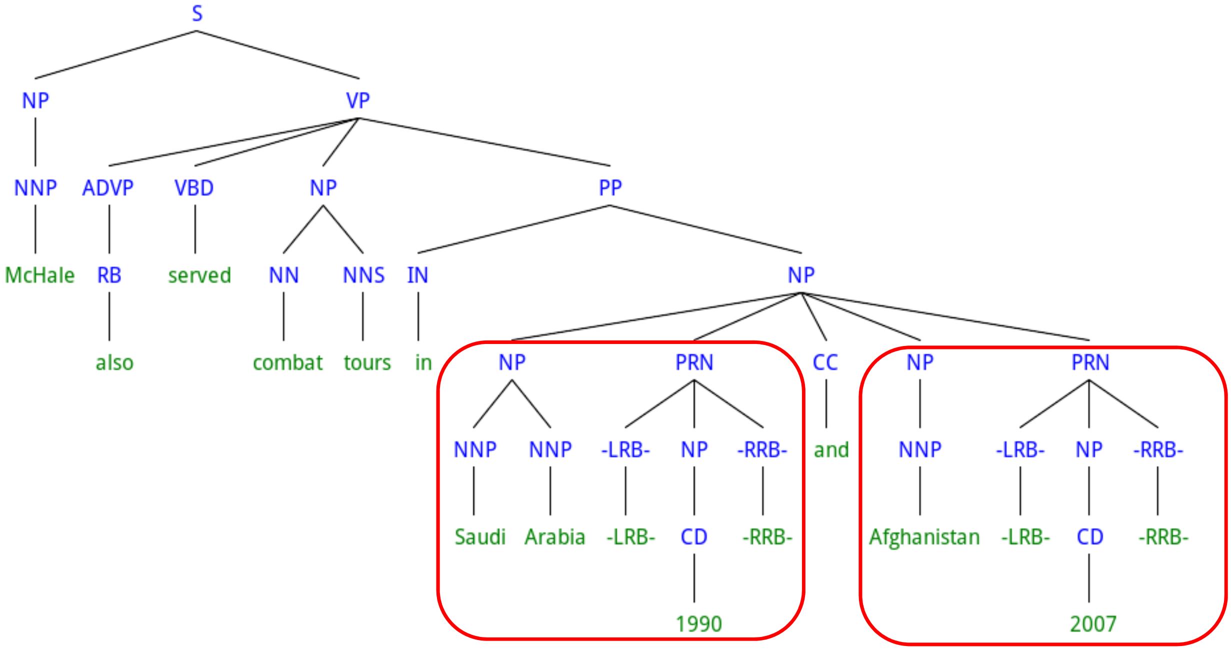


Figure 9. Two noun phrase conjoints in the constituent parse tree for a given example sentence.

also served combat tours in Saudi Arabia (1990) and *McHale also served combat tours in Afghanistan (2007)*. From the simplified sentences, TRIPLEX-ST can extract two triples: `<McHale; served; combat tours> (spatial_context: Saudi Arabia; temporal_context: 1990)` and `<McHale; served; combat tours> (spatial_context: Afghanistan; temporal_context: 2007)`.

If there exists a verb phrase conjoint without an explicit subject in a compound sentence, we need to find the subject and boundary of the verb phrase before rewriting the sentence. As mentioned in Section 4.2.1, the set of candidate subjects of verb phrase conjoints can be constructed automatically.

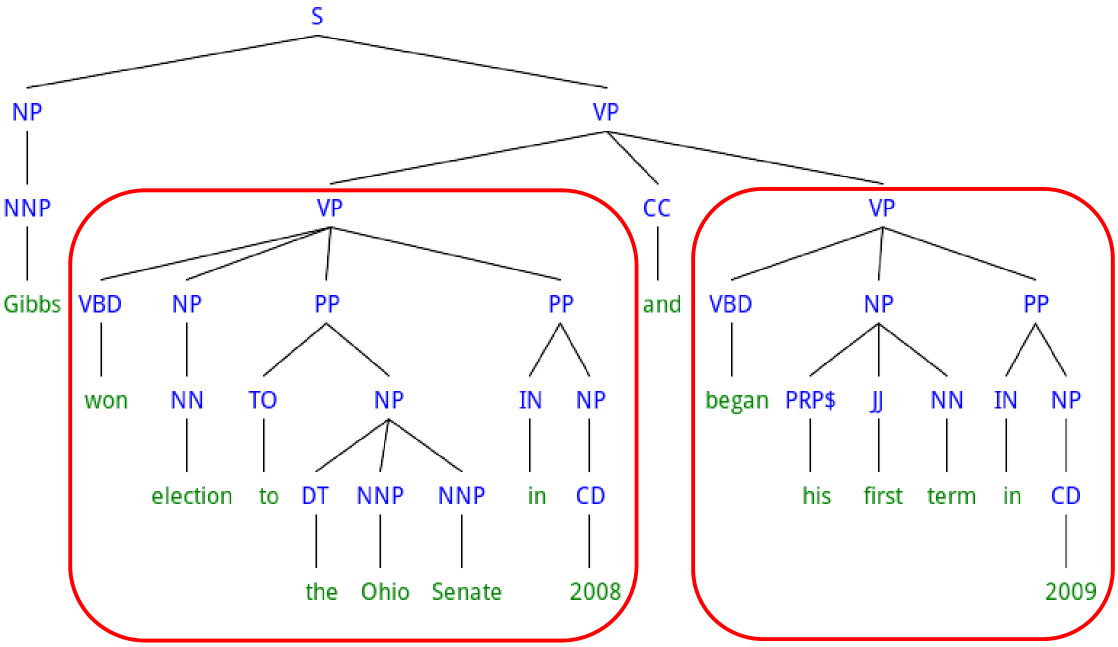


Figure 10. Two verb phrase conjoints in the constituent parse tree for a given example sentence.

Next, we detect the correct subjects for verb phrase conjoints based on a pre-trained classifier, as will be explained in Section 4.2.3. Finally, we rewrite each verb phrase conjoint and its corresponding subject as a separate sentence. For example, the sentences *Gibbs won election to the Ohio Senate in 2008* and *Gibbs began his first term in 2009* are extracted after simplifying the sentence in Figure 10. From the simplified sentences, TRIPLEX-ST can extract two triples: `<Gibbs; won; election>` `(temporal_ context: 2008)` and `<Gibbs; began; his first term;>` `(temporal_ context: 2009)`.

4.2.3 Identifying Subjects of Re-written Clauses

We pre-trained a single logistic regression classifier to detect the correct subjects of dependent clauses and verb phrase conjoints, by using automatically labeled sentences (i.e., complex sentences that describe facts) collected from Wikipedia pages. We collected approximately 25,800 sentences, including complex and compound sentences, from Wikipedia. We first extract dependent clauses, verb phrase conjoints, and candidate subjects from the sentences. We then classify each dependent clause and verb phrase conjoint together with candidate subjects, as positive or negative instances. First, we use the dependency relations corresponding to adverbial clause modifiers, relative clause modifiers, clausal complements, coordinates, conjunctions, and clausal modifiers, to recognize dependent clauses and verb phrase conjoints in sentences. Next, we detect the boundaries of each dependent clause and verb phrase conjoint based on the constituent parse tree. Moreover, we use the constituent parse tree and dependency paths to construct a set of candidate subjects for each dependent clause or verb phrase conjoint. A candidate subject is typically a noun phrase or a pronoun that is identified in the constituent parse tree. A candidate subject is valid if there exists a dependency path between the candidate subject and the verb phrase of the dependent clause or the verb phrase conjoint. We rewrite each dependent clause and its candidate subjects as a simple sentence. For example, the dependent clause *finishing runner-up seven times in 2002* in Figure 8 and its candidate subjects can be written into the following simple sentences: *Rock finishing runner-up seven times in 2002*, *Stride finishing runner-up seven times in 2002*, and *Eight consecutive titles finishing runner-up seven times in 2002*. We similarly rewrite each verb phrase conjoint and its candidate subjects as a simple sentence. Facts are then extracted from simplified sentences. For example, the triples `<Rock; finished; runner-up seven times>`, `<Stride;`

finished; runner-up seven times>, and <Eight consecutive titles; finished; runner-up seven times> are extracted from the simplified sentences.

We use the existing knowledge bases and triple-PMI shown in Equation 3.2 to evaluate extracted triples from each dependent clause and its candidate subjects (7). A triple is correct if its corresponding fact has been found in one of the considered knowledge bases, or if its PMI score is above the threshold of 10^{-2}, which was determined experimentally. If a fact is judged as correct, its corresponding dependent clause and candidate subject are classified as a positive instance. Otherwise, dependent clauses and their candidate subjects are classified as negative instances. We similarly classify positive and negative instances for verb phrase conjoints and their candidate subjects. Finally, we use the positive and negative instances to train the classifier. The set of features used to detect subjects of dependent clauses and verb phrase conjoints is shown in Table VII.

4.3 Evaluation

To evaluate TRIPLEX-ST, we first used a comprehensive set of experiments to access the extraction of spatio-temporal contexts. Afterwards, we compared the output of TRIPLEX-ST, in terms of the extraction of static triples, against static spatio-temporal triples produced by the state-of-the-art systems like OLLIE (30) and Stanford OIE (1).

We mostly leverage the automated evaluation approach proposed by Bronzi et al. (7) to assess TRIPLEX-ST, as explained in 3.2. We extended their approach to evaluate facts associated with contexts. The automatic evaluation procedure uses existing knowledge bases and a Triple-PMI metric to verify whether a fact is correctly extracted or not. In a separate set of experiments, using a manual evaluation procedure and a small corpus, a human judge also verified each extracted fact.

Features for finding a candidate subject
Whether the candidate subject is before or after the dependent clause/verb phrase conjoint
Named entity type of candidate subject
Whether a Wh-pronoun (that, who, when, whom, etc.) is after the dependent clause/verb phrase conjoint
Whether there exists a dependent clause or a verb phrase conjoint
POS tags of candidate subject
Whether there exists a preposition exactly before a candidate subject
Whether a Wh-pronoun is before the dependent clause/verb phrase conjoint
The next node after the candidate subject in preorder traversal of constituent parse tree has constituent label VP
Number of leaf nodes between the candidate subject and the first word of a dependent clause/verb phrase conjoint in the constituent parse tree
Number of leaf nodes between the candidate subject and the last word of a dependent clause/verb phrase conjoint in the constituent parse tree
Whether there exist a singular or plural agreement between the candidate subject and the verb of a dependent clause/verb phrase conjoint

TABLE VII

FEATURES IN THE CLASSIFIER THAT IS USED FOR IDENTIFYING SUBJECTS OF DEPENDENT CLAUSES AND VERB PHRASE CONJOINTS.

4.3.1 The Evaluation Procedure

An evaluation dataset was created by taking 50,000 random sentences from Wikipedia that have not been used for bootstrapping. All extracted facts, gathered by different OIE systems (i.e., TRIPLEX-ST, OLLIE, and Stanford OIE), are collected in the context of the evaluation set. All extracted facts, gathered by different OIE systems and extended in order to also consider possible spatio-temporal contexts, are collected in the context of the evaluation set.

Considering the sentence *Laurent signed a one-year contract with Southend United,* the sets S and O of recognized entities include *Laurent*, *a one-year contract*, and *Southend United.* In our case, we specifically focus on static spatio-temporal facts. The subject type or object type of these static facts is either *Location* or *Date*, and this rule is used to filter the results of the Cartesian product. Moreover, we use Boolean ORs to consider different renderings of dates and location levels (i.e., city, state, and country) for each query when we compute the Triple PMI to validate facts. In order to retrieve location levels, we use the GeoNames[1] service, which contains administrative divisions associated with places. We also use HeidelTime to recognize and normalize entities. Moreover, we rely on the Joda-Time library for normalizing temporal expressions and matching them to the knowledge base properties.

We further select 50 sentences from the dataset of 50,000 sentences, and a human judge extracts all correct facts from this small collection. This small dataset constitutes our manually annotated gold standard corpus, which we used for manual evaluation and also for measuring the agreement between the manual and automated evaluation procedures. The agreement is defined as the ratio between the number of facts where the human and automatic evaluators agree, and the total number of facts (7).

We also extend the automatic approach proposed by Bronzi et al. to evaluate the spatio-temporal contexts of facts. In this case, we only use YAGO instead of also considering Freebase, Wikidata, and DBpedia, since many of its facts are associated with the corresponding spatial and temporal contexts.

A dynamic fact and its spatio-temporal context is correct if the fact is found in the YAGO knowledge base or if there exists a significant association between its entities (subjects and objects), its relation,

[1] http://www.geonames.org/

and its corresponding spatio-temporal context. We adapt the triple PMI function, as defined in Equation Equation 3.2, to compute the PMI score for a dynamic fact and its spatio-temporal context. The Triple-SP-PMI measures the likelihood of observing the fact and its corresponding spatio-temporal context given that we observe its subject (s) and its object (o), independently of its relation (r) and its spatio-temporal context (c):

$$\text{Triple-SP-PMI}(s, r, o, c) = \frac{\text{Count}(s \ \wedge \ r \ \wedge \ \ o \wedge c)}{\text{Count}(s \ \wedge \ o)} \tag{4.2}$$

We use Boolean ORs in order to submit multiple queries into the Lucene index when using the PMI based evaluation method. The queries for building the gold standard involve a subject, an object, a relation, and a context. We use the OR operator to consider different variations for each query, specifically using different renderings of dates and location levels (i.e., city, state, and country). In particular, a fact and its corresponding spatio-temporal context is deemed correct if its PMI score is above the threshold value of 0.2, which was determined experimentally. We use four sets S, O, P, and C, that are respectively the set of identified subjects, objects, relations, and contexts. These sets are used to generate all possible facts and their temporal contexts. Finally, we use Equations Equation 3.1 and Equation 3.3 to estimate precision and recall for TRIPLEX-ST, in the case of dynamic facts.

4.3.2 Results for the Extraction of Dynamic Facts Connected to Spatio-Temporal Contexts

We first evaluate TRIPLEX-ST when extracting dynamic facts associated with spatio-temporal contexts. Triples with spatio-temporal contexts can be divided into three groups according to the types of their corresponding contexts. We evaluate each group individually, using the automatic evaluation procedure and also with basis on the manually labeled dataset. Table VIII shows the results in terms of precision, recall and F1 (harmonic mean of precision and recall) when considering only the original sentences, and when considering also their simplified versions. The agreement between both evaluation procedures is 78% for original sentences and 83% when considering the original sentence plus the simplified sentences. Table IX shows the results for the extraction of triples associated to spatio-temporal contexts in terms of precision, recall and F1, for the original TRIPLEX-ST, and its different configurations. When the bootstrapping sets include noisy sentences and TRIPLEX-ST infers templates from all sentences, the precision decreases significantly, as shown in Table IX. Similarly, when TRIPLEX-ST does not use VerbNet annotations, the precision slightly decreases.

The performance of OIE systems is known to degrade when processing complex sentences containing dependent clauses and/or coordinated constituents. Thus, we improve the extraction results by simplifying complex sentences and rewriting them into several simple sentences before extracting facts, as shown in Table VIII and Table IX.

Recall that TRIPLEX-ST assigns a confidence score to each extraction. Figure 11 shows the variation in the F1 score when considering different threshold values for the confidence scores of triples associated with spatio-temporal contexts. In our tests, we achieved the best result when this confidence score is above the threshold value of 0.3.

We use the smaller manually annotated dataset to analyze the types of errors made by TRIPLEX-ST, and the automated evaluation procedure, as shown in Table X and Table XI. The errors are classified into two groups: false positives and false negatives. Table X details the false positive errors while Table XI shows the percentage of triples missed by TRIPLEX-ST.

		Automatic evaluation			**Manual evaluation**		
	Context	Prec	Rec	F1	Prec	Rec	F1
Original sentences	Temporal	0.66	0.4	0.49	0.7	0.42	0.52
	Spatial	0.55	0.25	0.34	0.6	0.27	0.37
	Spatial and/or temporal	0.65	0.36	0.46	0.69	0.38	0.49
Simplified sentences	Temporal	0.69	0.47	0.55	0.7	0.49	0.55
	Spatial	0.62	0.3	0.4	0.65	0.38	0.47
	Spatial and/or temporal	0.69	0.43	0.52	0.71	0.45	0.55

TABLE VIII

RESULTS FOR THE EXTRACTION OF TRIPLES ASSOCIATED WITH SPATIO-TEMPORAL CONTEXTS.

The total number of triples having dynamic temporal and/or spatial contexts in YAGO is 709,012. The total number of triples involving dynamic spatio-temporal contexts that are extracted by Triplex-ST, from the 50,000 sentences, is 77,948 from a maximum possible value of 140,964 triples in the automatic evaluation procedure.

Approximately 33% of the errors that are detected by the automatic procedure are in fact correct triples extracted by Triplex-ST. This stems from the fact that there are few documents for queries that contain the subjects, objects, and the spatial and/or temporal contexts of these triples, thus impacting the computation of the Triple-SP-PMI scores. In these cases, applying the same PMI threshold as the one applied for prominent subjects proves to be ineffective. For example, the spatio-temporal context of the triple `<Greg Howes; played for; Rochester Raging Rhinos> (temporal_context:`

	Prec	Rec	F1
TRIPLEX-ST	0.65	0.36	0.46
TRIPLEX-ST (without filtering noisy sentences)	0.2	0.35	0.25
TRIPLEX-ST (without VerbNet)	0.6	0.36	0.45
TRIPLEX-ST (without WordNet)	0.55	0.29	0.37
TRIPLEX-ST (without coreference resolution)	0.6	0.24	0.34

TABLE IX

RESULTS FOR THE EXTRACTION OF TRIPLES ASSOCIATED TO SPATIO-TEMPORAL CONTEXTS, FOR DIFFERENT CONFIGURATIONS OF TRIPLEX-ST.

Percentage	**Incorrect Extractions**
33%	False positive triples
30%	Wrong types by NER
9%	Clauses with relative pronouns *which*, *that*, etc.
28%	Other errors

TABLE X

PERCENTAGE OF ERRORS IN THE EXTRACTIONS PRODUCED BY TRIPLEX-ST WHEN CONSIDERING SPATIO-TEMPORAL CONTEXTS.

`[2005])` extracted from the sentence *Greg Howes played for the Rochester Raging Rhinos in 2005 before returning to the Seattle Sounders in 2007*, is judged incorrect due to a low Triple-SP-PMI score.

Similarly, 30% of the errors are the result of wrong semantic types assigned by the NER system for the spatial contexts of triples. For instance, the semantic type of *Burnley* should be *Organization*, but the NER system cannot recognize the correct type in the sentence *Arthur Bell played two matches in the Football League for Burnley in the 1902-03 season*.

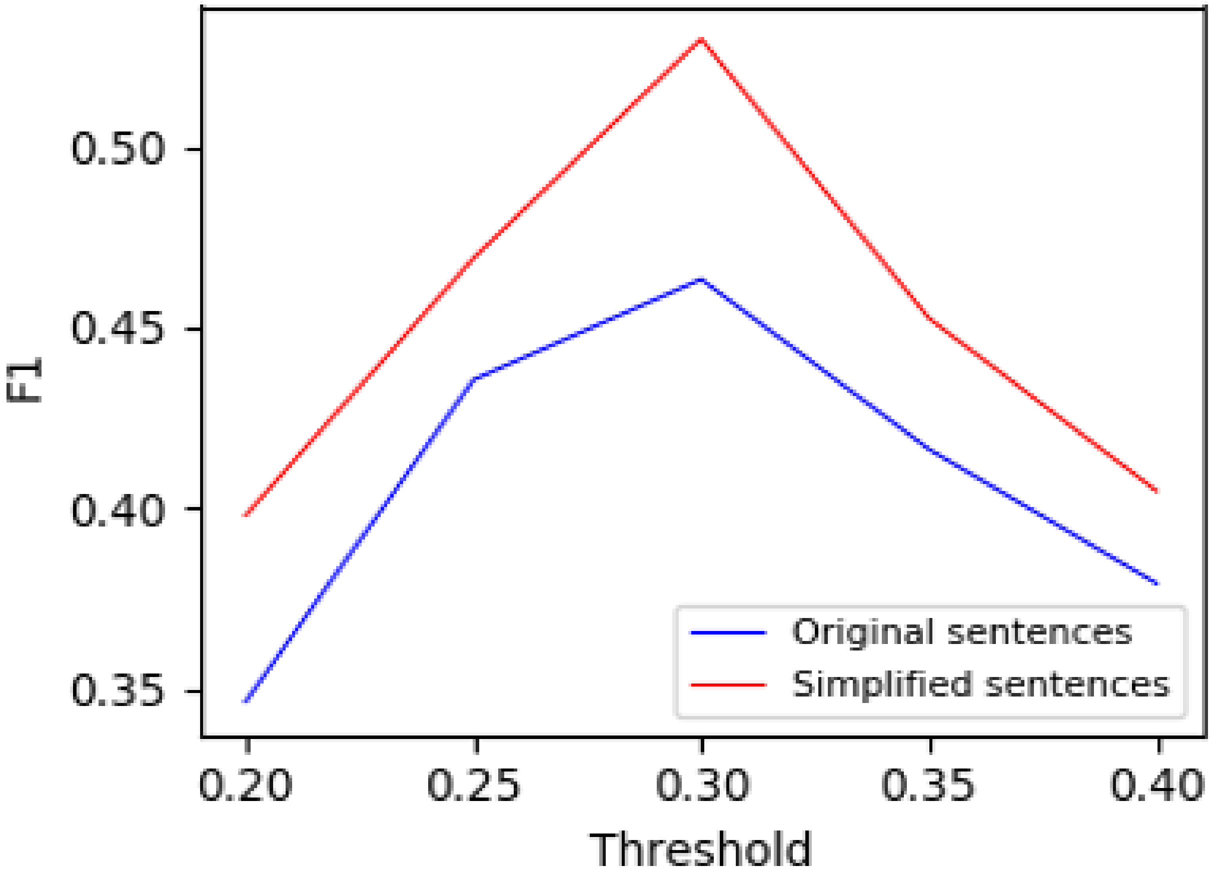


Figure 11. Results for the extraction of triples associated with spatio-temporal contexts, using different thresholds for confidence scores.

Also, 9% of the errors are related to triples extracted from dependent clauses including relative pronouns (e.g., *which* or *that*). For example, TRIPLEX-ST cannot identify the correct subject of the clause *was found on the body of Stephen Foster in January 1864* in the sentence *Bob Hilliard was inspired to write the song based on a scrap of paper with the words “Dear friends and gentle hearts" that was found on the body of Stephen Foster in January 1864 when he was discovered in a New York*

Percentage	**Missed Extractions**
17%	Constituent parser
9%	Absence of coordinating conjunctions in sentences
15%	Clauses with relative pronouns *which*, *that*, etc.
8%	Parenthetical phrases in sentences
24%	Wrong types by NER
27%	Other errors

TABLE XI

PERCENTAGE OF TRIPLES MISSED BY TRIPLEX-ST WHEN CONSIDERING DYNAMIC FACTS WITH SPATIO-TEMPORAL CONTEXTS.

hotel room due to existence of the dependent clause with passive voice *was found on the body of Stephen Foster in January 1864* and also the relative pronoun *that* in the sentence. Thus, it cannot extract correctly the triple <a scrap of paper; was found on; the body of Stephen Foster> (Temporal_ context: [January 1864)] from the sentence.

Finally, 28% of the errors happen due to other problems (e.g., in the dependency parser, or the POS tagger).

The percentage of missed triples that are lost in recall is 17% when there are errors in constituent parse trees of sentences. For instance, TRIPLEX-ST cannot extract the temporal context of the triple <They; followed next with; The Road to Ellenside> (temporal_context: [2006]) from the sentence *They followed next with Tenure in 2002, the double album Dilapidated Beauty in 2003, and The Road to Ellenside in 2006.* Since the boundary of the noun phrase conjoint *The Road to Ellenside in 2006* is not identified correctly by the constituent parse tree of the sentence, the pipeline cannot extract the triple.

Also, 9% of the errors happen due to absence of coordinating conjunctions (e.g., *and*) between noun phrases in sentences. For instance, TRIPLEX-ST cannot extract the triples `<Niyi Ogunlana; has played for; Katsina United F.C. of Katsina> (temporal_context: [2001-[2003])` and `<Niyi Ogunlana; has played for; Kwara United F.C. of Ilorin> (temporal_context: [2004-2006])` from the sentence *Niyi Ogunlana has played for Kwara Stars F.C. of Ilorin (1997-2001), Katsina United F.C. of Katsina (2001-2003), Kwara United F.C. of Ilorin (2004-2006).* Since there is no coordinating conjunction *and* between noun phrases in the sentence, the pipeline cannot detect noun phrase conjoints *Katsina United F.C. of Katsina (2001-2003)* and *Kwara United F.C. of Ilorin (2004-2006).*

We also notice that 15% of the errors are related to sentences with relative pronouns (e.g., *which* or *that*). TRIPLEX-ST cannot identify correctly the subjects of triples expressed in this kind of sentences. For example, TRIPLEX-ST extracts the triple `<Nippon Animation; re-animated; 3000 Leagues> (temporal_context: [1999])`, but it cannot extract the triple `<song; was included in; her album> (temporal_context: 1999)` from the sentence *Nippon Animation also re-animated 3000 Leagues as a feature-length film in 1999, with a theme song performed by Scottish pop superstar Sheena Easton ("Carry a Dream"), which was included in her album called Home in 1999 that was only released in Japan.*

Approximately 8% of the missed triples are related to parenthetical phrases. For example, TRIPLEX-ST could not extract the temporal contexts of triples in the sentence *Persi Diaconis returned to school (City College of New York for his undergraduate work graduating in 1971 and then a Phd in Mathe-*

matical Statistics from Harvard University in 1974), learned to read Feller, and became a mathematical probabilist.

Also, 24% of the errors are due to wrong semantic types assigned by the NER system for the spatial contexts of triples. For example, the NLP pipeline assigns an incorrect semantic type to the phrase *Rio Claro*, labeling it as a *Location* instead of an *Organization*. Consequently, TRIPLEX-ST missed the triple `<Pedro Henrique Martins; transferred; Rio Claro> (temporal_context: 01/2010)` from the sentence *Soon after Pedro Henrique Martins transferred to Rio Claro in January 2010.*

Finally, 27% of the errors happen due to other problems (e.g. in the dependency parser or the POS tagger).

4.3.3 Results for Static Spatio-Temporal Triples

When considering static spatio-temporal triples, TRIPLEX-ST and Stanford OIE (1) are compared individually and then evaluated in combination. TRIPLEX-ST is also compared individually against OLLIE (30). We use substring matches in order to merge the results of Stanford OIE and TRIPLEX-ST for each sentence. Whenever the subject, object, and relation phrases of a triple, produced by one of these systems, is respectively a substring of the subject, object, and relation of a triple produced by the other system, the triples are considered equivalent (i.e., only one triple is produced, with the longest phrases for subject, object, and relation). Table XIII shows results in terms of precision, recall, and F1 (harmonic mean of precision and recall), when extracting static spatio-temporal triples. The agreement between both evaluation procedures is 70%. The total number of spatio-temporal triples that is extracted by TRIPLEX-ST, from the 50,000 sentences and out of a maximum possible value of

Relation Type	**Number of Extractions**
`<Person; born; Date>`	1,850
`<Person; born; Location>`	1,436
`<Person; died; Date>`	1,300
`<Person; died; Location>`	1,200
`<Person; residence; Location>`	400
`<Person; served; Location>`	49
`<Organization; established; Date>`	400
`<Organization; headquarters; Location>`	100
`<Organization; owner; Location>`	90
`<Organization; foundation; Location>`	75
`<Location; region; Location>`	200
`<Location; population; Number>`	70
`<Location; discovered; Date>`	40
`<Location; area; Number>`	30
`<Location; built; Date>`	15
`<Location; abolished; Date>`	15
`<Unknown; term start; Date>`	500
`<Unknown; conflict; Location>`	30
`<Unknown; released; Date>`	20
`<Unknown; venue; Location>`	20

TABLE XII

THE MOST COMMON TYPES OF STATIC SPATIO-TEMPORAL RELATIONS EXTRACTED BY TRIPLEX-ST.

37,124 triples (i.e., the denominator in Equation 3.3) is 14,178. When combining TRIPLEX-ST with Stanford OIE, the number of extracted triples raised to 15,592. Table XII shows the most common types of spatio-temporal relations that are extracted from the 50,000 Wikipedia sentences.

TRIPLEX-ST assigns a confidence score to each extracted triple. In our experiments, the extracted triples are only considered if their confidence scores are above the threshold of 0.2. Table XIII shows

		Automatic evaluation			Manual evaluation		
	System	Prec	Rec	F1	Prec	Rec	F1
Original sentences	TRIPLEX-ST	0.5	0.41	0.45	0.53	0.43	0.47
	OLLIE	0.31	0.27	0.28	0.32	0.3	0.31
	Stanford OIE	0.3	0.34	0.31	0.33	0.36	0.34
	TRIPLEX-ST + Stanford OIE	0.4	0.43	0.41	0.42	0.47	0.44
Simplified sentences	TRIPLEX-ST	0.54	0.48	0.5	0.56	0.49	0.52
	OLLIE	0.4	0.38	0.39	0.42	0.4	0.41
	Stanford OIE	0.41	0.39	0.4	0.43	0.42	0.42
	TRIPLEX-ST + Stanford OIE	0.48	0.5	0.49	0.5	0.51	0.5

TABLE XIII

RESULTS FOR EXTRACTING STATIC SPATIO-TEMPORAL TRIPLES.

better results for TRIPLEX-ST when using the manual evaluation procedure, since the extracted facts with very low PMI scores are considered as false in the automatic evaluation. However, these facts are often evaluated as true positives by a human judge.

We also analyze the errors made by TRIPLEX-ST in the gold standard dataset that is manually annotated. These can be classified into two groups: false positives and false negatives. Table XIV shows the types of errors that are associated with the triples in the gold standard that are not extracted by TRIPLEX-ST (i.e., the false negatives), while Tables Table XV details the false positives.

The percentage of missed triples that are lost in recall is 15% when the range of a property in the considered KBs is a literal and the property has no specific range. The pipeline requires more reliable methods to extract these kinds of triples.

Percentage	Missed Extractions
15%	Properties with no specific ranges
20%	No semantic types for objects
18%	Properties with specific ranges
47%	Other errors

TABLE XIV

PERCENTAGE OF STATIC SPATIO-TEMPORAL TRIPLES THAT WERE MISSED BY TRIPLEX-ST, DUE TO PARTICULAR PROBLEMS.

The percentage of triples with a low confidence score is 20%, and most of these triples are filtered out because the procedure for entity recognition cannot assign the specific semantic type for the objects of these triples. If there is no semantic type for the object of a triple, the confidence score of the triples is penalized.

The percentage of missed triples due to unclear semantic properties is 18%. These triples do not have the semantic type for their objects and, simultaneously, the properties of these triples have specific ranges. The pipeline (i.e., the final logistic regression classifier) filters out these triples, by assigning them a low confidence.

Finally, 47% of the errors happen due to incorrect grammatical relations, coreference resolution errors, and improper lexical constraints of templates.

We also estimate that 30% of the errors (false positives) in the automated evaluation procedure are indeed correct triples. These triples have subjects that are not commonly found (i.e., there are few documents about them) and thus their triple PMI scores are very low. Using the same PMI threshold as that which is applied for prominent subjects proved to be ineffective.

Similarly, 13% of the errors are the result of extracting templates from noisy sentences during the bootstrapping process. The distribution of noisy sentences in the bootstrapping set is not the same for all properties, and applying the same threshold is not effective for particular properties (e.g., capital city). A higher threshold is necessary to filter these noisy sentences.

Over-generalized templates cause 9% of missed triples. During template generalization, POS tags are substituted by universal POS tags (38). Since some templates only extract triples for proper nouns, common nouns or personal pronouns, generalizing and merging these templates does not produce good results.

Finally, 48% of the errors happen due to other problems in the dependency parser, the POS tagger, the coreference resolution modules, the named entity recognizer, or the noun phrase chunker. Incorrect NLP annotations in the templates can generate wrong triples.

Table XIII also shows that sentence simplification improves the extraction results of other OIE systems by rewriting complex sentences that include coordinated constituents and/or dependent clauses into a set of simple sentences.

We can also examine the output of TRIPLEX-ST with respect to the types of triples that are extracted and how the relations are expressed, as shown in Table XVI. As shown in previous studies (30), noun-mediated relations indeed correspond to a significant amount of the relations that can be extracted. The TRIPLEX-ST system seems to be equally capable of extracting noun-mediated and verb-mediated relations encoding spatio-temporal information.

Percentage	Incorrect Extractions
30%	False positive triples
13%	Templates from noisy sentences
9%	Over-generalized templates
48%	Other errors

TABLE XV

PERCENTAGE OF ERRORS IN THE EXTRACTIONS PRODUCED BY TRIPLEX-ST, WHEN CONSIDERING STATIC SPATIO-TEMPORAL TRIPLES.

	Distribution	Triple category
Noun-mediated	3%	Noun phrases
	4%	Adjectives
	9%	Parenthetical phrases
	6%	Appositions
	6%	Templates with lexicon
	9%	Conjunctions
Verb-mediated	63%	Verb-mediated triples

TABLE XVI

DISTRIBUTION FOR TYPES OF CORRECTLY EXTRACTED TRIPLES.

4.4 Conclusions

We presented TRIPLEX-ST, a novel OIE system for the collection of static spatio-temporal facts and dynamic facts connected to spatial and/or temporal contexts. A preliminary version of TRIPLEX-ST was presented as a poster at ACM SIGSPATIAL 2016 (35). Here, we further detail the proposed method and present, for the first time, the experimental evaluation results. TRIPLEX-ST uses rich linguistic annotations together with information available in Wikipedia and in other knowledge bases. Our experimental results attest to the effectiveness of the proposed approach. For instance, TRIPLEX-ST has inferred a total of 40,018 templates for extracting facts associated with spatial and/or temporal contexts from a large Wikipedia dataset. Additionally, we improve the results of TRIPLEX-ST through restructuring complex sentences, capturing facts from dependent clauses and/or coordinated constituents.

Overall, with an automated evaluation procedure, we estimate an F1 score of 0.52 for the extraction of dynamic facts, and an F1 score of 0.5 for the extraction of static spatio-temporal facts. Our tests have also confirmed that TRIPLEX-ST could outperform previous systems (e.g., OLLIE or Stanford OIE) on the extraction of static facts, while at the same time extending OIE in the direction of considering dynamic information.

CHAPTER 5

FUTURE WORK

A comprehensive knowledge base is essential for many different applications, such as textual inference (27), entity linking and disambiguation (40), semantic search (28), and question answering (15). To achieve this goal, a large number of knowledge bases have been constructed, such as Google's Knowledge Graph (44), DBpedia (2), YAGO (4) BabelNet (36) and Probase (56). Since factual knowledge is highly ephemeral, it is essential to populate and enrich existing knowledge bases instantaneously and automatically with information scattered throughout a tremendous amount of electronic documents, without human intervention.

Future work can focus on further advancing OIE methods in order to support the expansion of information in knowledge bases.

The remainder of this chapter is organized as follows: In Section 5.1, we describe future work related to the extraction of spatio-temporal triples from textual resources. Finally, Section 5.2 presents the idea of automatically assessing triples extracted by OIE methods.

5.1 Extraction of Spatio-temporal Information

There are also many opportunities to enhance the information extraction procedure from TRIPLEX-ST. Improvements in the different NLP components can, for instance, lead to better results. We also note that although our method relies on a complex NLP pipeline and on lexical resources such as WordNet or VerbNet, it would be fairly easy to adapt TRIPLEX-ST to languages other than English. For instance both HeidelTime and Stanford Core NLP already support different languages. We can also enrich the bootstrapping sets by using a self-training algorithm (51) to learn further templates based on the initial bootstrapping sets.

Specifically in the context of spatio-temporal fact extraction, future experiments can consider issues such as: (i) evaluating the impact of adding further constraints to the extraction templates, which are specific to these domains (e.g., instead of coarse-grained types such as *Location* or *Date*, we could consider types such as *City*, *Month*, or *Year* in the lexical constraints); and (ii) enriching the set of templates that can deal with spatio-temporal contexts by combining them with similar templates than only deal with subjects, objects, and predicates.

Besides improvements in the quality of the results, there are also many possibilities for future applications, particularly if we consider that the information extraction results are going to be used to enrich existing knowledge bases. The reconciliation of facts is nonetheless an important challenge, as inconsistencies can arise between the extracted facts and those already present in existing knowledge bases. More than just performing entity linking (18) or place reference resolution (41) over the subjects, objects, or spatial contexts of triples, as currently made by TRIPLEX-ST, it is necessary to access the correctness of extracted facts and remove the noise, derive reasonable new facts, and reconcile all the available information (e.g., adding new spatial and/or temporal information may require inferring valid contexts from multiple observations, possibly with multiple granularities).

5.2 Predicting the Validity of Extracted Triples

A massive body of text is available on the web and this presents an extraordinary opportunity for OIE systems to extract factual information of interest to many different applications. Moreover, the correctness of the extracted triples should be carefully assessed before using these triples. An OIE assessor takes as input a list of candidate triples and ranks them so that correct triples are prioritized. One important assumption is that triples occurring more frequently are more likely to be correct. However, there are triples that occur very infrequently and that are nonetheless also correct.

An OIE assessor usually ranks extracted triples based on the types of their subjects and objects. Moreover, the semantic relations between subjects and objects are considered while ranking triples. For example, the triple <Bill Gates; Founded; Microsoft> has correct types for its subject and object. More formally, a list of extracted triples is denoted as ET = <$\texttt{subject}_1$, $\texttt{relation}_1$, $\texttt{object}_1$>, ..., <$\texttt{subject}_m$, $\texttt{relation}_m$, $\texttt{object}_m$>. An extracted triple is correct if and only if the $\texttt{relation}_i$ holds between the $\texttt{subject}_i$ and the $\texttt{object}_i$ in the real world. For example the triple, <subject; Headquartered; object> is correct if there exists an organization, subject, that is in fact headquartered in a location, object.

Challenges:

- How to assess triples based upon the context (i.e., the sequence of surrounding words) of semantic relations between their subjects and objects?
- How to rank triples based upon features such as their occurrence frequency, or existence of correct types for their subjects and objects for a specific semantic relation?

CHAPTER 6

CONCLUSIONS

My PhD research focuses on the area of Open-domain Information Extraction (OIE). We proposed TRIPLEX, an information extractor that complements previous efforts, concentrating on noun-mediated triples related to nouns, adjectives, and appositions. TRIPLEX automatically constructs templates expressing noun-mediated triples from a bootstrapping set. The bootstrapping set was constructed without manual intervention by creating templates that include syntactic, semantic, and lexical constraints. Our experimental study indicated that TRIPLEX is a promising approach for extracting noun-mediated triples.

We also advance OIE methods in order to enrich information present in knowledge bases. We perform the realization of experiments focusing on the following problems: extraction of spatio-temporal information and improvement of OIE results through the usage of sentence re-structuring methods.

We present TRIPLEX-ST, a novel OIE system for the collection of static spatio-temporal facts and dynamic facts connected to spatial and/or temporal contexts. We use rich linguistic annotations together with information available in Wikipedia and in other knowledge bases. Our experimental results attest to the effectiveness of the proposed approach. Our tests have also confirmed that TRIPLEX-ST could outperform previous systems (e.g., OLLIE or Stanford OIE) on the extraction of static facts, while at the same time extending OIE in the direction of considering dynamic information. Additionally, we improve the results of TRIPLEX-ST through restructuring complex sentences, capturing facts from dependent clauses and/or coordinated constituents. Our sentence simplification improves the extraction results of other OIE systems (e.g., Stanford OIE (1)) by rewriting complex sentences into a set of simple sentences. Finally, We extend the automated evaluation procedure of Bronzi et al. (7) to assess noun-mediated triples and triples associated to spatio-temporal contexts.

CITED LITERATURE

1. Gabor Angeli, Melvin Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 344–354, 2015.

2. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *International Semantic Web Conference (ISWC)*, pages 722–735, 2007.

3. Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open Information Extraction for the Web. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, 2007.

4. Joanna Biega, Erdal Kuzey, and Fabian M. Suchanek. Inside YAGO2s: A Transparent Information Extraction Architecture. In *International World Wide Web Conference (WWW)*, pages 325–328, 2013.

5. Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1247–1250, 2008.

6. Stefan Bott and Horacio Saggion. An Unsupervised Alignment Algorithm for Text Simplification Corpus Construction. In *Workshop on Monolingual Text-To-Text Generation (Text-to-Text)*, pages 20–26, 2011.

7. Mirko Bronzi, Zhaochen Guo, Filipe Mesquita, Denilson Barbosa, and Paolo Merialdo. Automatic Evaluation of Relation Extraction Systems on Large-scale. In *NAACL-HLT Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 19–24, 2012.

8. Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. Survey of Temporal Information Retrieval and Related Applications. *ACM Compututing Surveys*, 47(2):15:1–15:41, 2014.

9. William Coster and David Kauchak. Learning to Simplify Sentences Using Wikipedia. In *Workshop on Monolingual Text-to-text Generation (Text-to-Text)*, pages 1–9, 2011.

10. Marie-Catherine De Marneffe and Christopher D Manning. Stanford Typed Dependencies Manual, 2008.

11. Luciano Del Corro and Rainer Gemulla. ClausIE: Clause-based Open Information Extraction. In *International World Wide Web Conference (WWW)*, pages 355–366, 2013.

12. Leon Derczynski and Robert Gaizauskas. Information Retrieval for Temporal Bounding. In *ACM SIGIR Conference on the Theory of Information Retrieval (ICTIR)*, pages 129–130, 2013.

13. Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open Information Extraction from the Web. *Communications of the ACM*, 51(12):68–74, 2008.

14. Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying Relations for Open Information Extraction. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1535–1545, 2011.

15. David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building Watson: An Overview of the DeepQA Project. *AI magazine (AAAI)*, 31(3):59–79, 2010.

16. G. Garrido, A. Peñas, B. Cabaleiro, and Á. Rodrigo. Temporally Anchored Relation Extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 107–116, 2012.

17. Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3161–3165, 2013.

18. Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 782–792, 2011.

19. Ander Intxaurrondo, Mihai Surdeanu, Oier Lopez de Lacalle, and Eneko Agirre. Removing Noisy Mentions for Distant Supervision. In *Conference of the Spanish Society for Natural Language Processing (SEPLN)*, pages 41–48, 2013.

20. Heng Ji, Taylor Cassidy, Qi Li, and Suzanne Tamang. Tackling representation, annotation and classification challenges for temporal knowledge base population. *Knowledge and Information Systems*, 41(3):611–646, 2014.

21. Heng Ji and Ralph Grishman. Knowledge Base Population: Successful Approaches and Challenges. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1148–1158, 2011.

22. J.J. Jiang and D.W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *International Conference Research on Computational Linguistics (ROCLING)*, pages 19–33, 1997.

23. Christopher B. Jones and Ross S. Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3):219–228, 2008.

24. Dan Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson, 2014.

25. A. Khan, M. Vasardani, and S. Winter. Extracting Spatial Information from Place Descriptions. In *ACM SIGSPATIAL Workshop on Computational Models of Place*, pages 62–69, 2013.

26. Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40, 2008.

27. Ni Lao, Tom Mitchell, and William W. Cohen. Random Walk Inference and Learning in a Large Scale Knowledge Base. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 529–539, 2011.

28. Yuangui Lei, Victoria Uren, and Enrico Motta. Semsearch: A Search Engine for the Semantic Web. In *International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 238–245. 2006.

29. Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, 1993.

30. Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open Language Learning for Information Extraction. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 523–534, 2012.

31. Alexey Mazalov, Bruno Martins, and David Matos. Spatial Role Labeling with Convolutional Neural Networks. In *ACM SIGSPATIAL Workshop on Geographic Information Retrieval (GIR)*, pages 1–7, 2015.

32. Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant Supervision for Relation Extraction Without Labeled Data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1003–1011, 2009.

33. Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In *The Semantic Web: ESWC Satellite Events, Revised Selected Papers*, volume 9341, pages 230–243. 2015.

34. Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives. In *ESWC Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD)*, 2015.

35. Seyed Iman Mirrezaei, Bruno Martins, and Isabel F. Cruz. A Distantly Supervised Method for Extracting Spatio-Temporal Information from Text. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS)*, 2016.

36. Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence (AIJ)*, 193:217–250, 2012.

37. Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. A Sentence Simplification System for Improving Relation Extraction. In *Conference on Computational Linguistics: System Demonstrations (COLING)*, pages 170–174, 2016.

38. Slav Petrov, Dipanjan Das, and Ryan McDonald. A Universal Part-of-speech Tagset. In *International Conference on Language Resources and Evaluation (LREC)*, pages 2089–2096, 2011.

39. James Pustejovsky, Parisa Kordjamshidi, Marie-Francine Moens, Aaron Levine, Seth Dworman, and Zachary Yocum. SemEval-2015 Task 8: SpaceEval. In *International Workshop on Semantic Evaluation (SemEval)*, pages 884–894, 2015.

40. Delip Rao, Paul McNamee, and Mark Dredze. Entity Linking: Finding Extracted Entities in a Knowledge Base. In *Multi-source, Multilingual Information Extraction and Summarization (MULTISOURCE-MULITLI)*, volume 64, pages 93–115. 2013.

41. João Santos, Ivo Anastácio, and Bruno Martins. Using Machine Learning Methods for Disambiguating Place References in Textual Documents. *GeoJournal*, 80(3):375–392, 2015.

42. Jordan Schmidek and Denilson Barbosa. Improving Open Relation Extraction via Sentence Re-Structuring. In *International Conference on Language Resources and Evaluation (LREC)*, pages 3720–3723, 2014.

43. Matthew Shardlow. A Survey of Automated Text Simplification. *Journal of Advanced Computer Science and Applications (IJACSA)*, pages 58–70, 2014.

44. A. Singhal. Introducing the Knowledge Graph: Things, Not Strings. Official Google Blog, May 2012.

45. Jannik Strötgen and Michael Gertz. A baseline temporal tagger for all languages. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 541–547, 2015.

46. Fabian M. Suchanek, James Fan, Raphael Hoffmann, Sebastian Riedel, and Partha Talukdar. Advances in Automated Knowledge Base Construction. *SIGMOD Records*, 42(2), 2013.

47. Peter D. Turney. *Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL.* 2001.

48. Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James F. Allen, and James Pustejovsky. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *International Workshop on Semantic Evaluation (SemEval)*, pages 1–9, 2013.

49. Alakananda Vempala and Eduardo Blanco. Complementing Semantic Roles with Temporally Anchored Spatial Knowledge: Crowdsourced Annotations and Experiments. In *National Conference on Artificial Intelligence (AAAI)*, pages 2652–2658, 2016.

50. David Vickrey and Daphne Koller. Sentence Simplification for Semantic Role Labeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 344–352, 2008.

51. Duc-Thuan Vo and Ebrahim Bagheri. Self-training on Refined Clause Patterns for Relation Extraction . *Information Processing and Management*, in press, 2017.

52. Denny Vrandečić and Markus Krötzsch. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, 57:78–85, 2014.

53. Jan Oliver Wallgrün, Alexander Klippel, and Timothy Baldwin. Building a Corpus of Spatial Relational Expressions Extracted from Web Documents. In *ACM SIGSPATIAL Workshop on Geographic Information Retrieval (GIR)*, pages 1–8, 2014.

54. Yafang Wang, Maximilian Dylla, Marc Spaniol, and Gerhard Weikum. Coupling Label Propagation and Constraints for Temporal Fact Extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 233–237, 2012.

55. Fei Wu and Daniel S. Weld. Open Information Extraction Using Wikipedia. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 118–127, 2010.

56. Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A Probabilistic Taxonomy for Text Understanding. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 481–492, 2012.

57. Clarissa Xavier and Vera Lima. Boosting Open Information Extraction with Noun-Based Relations. In *International Conference on Language Resources and Evaluation (LREC)*, pages 96–100, 2014.

58. Wei Xu, Chris Callison-Burch, and Courtney Napoles. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics (TACL)*, 3:283–297, 2015.

59. Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1785–1794, 2015.

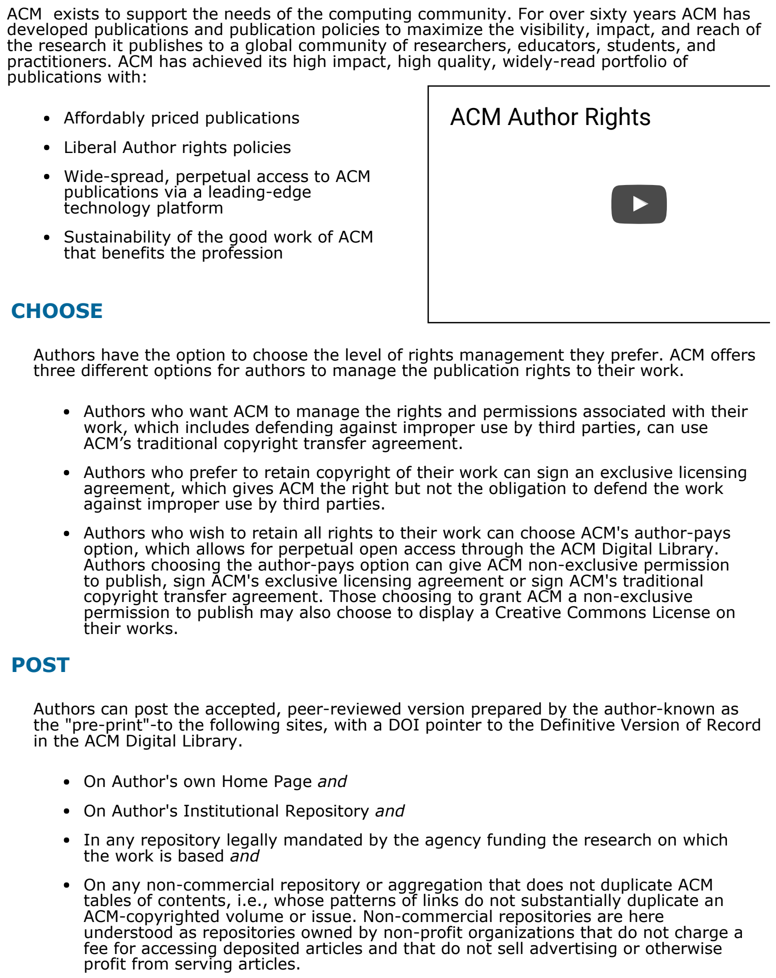
60. Mohamed Yahya, Steven Euijong Whang, Rahul Gupta, and Alon Halevy. Renoun: Fact Extraction for Nominal Attributes. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 325–335, 2014.

61. Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation Classification via Convolutional Deep Neural Network. In *Conference on Computational Linguistics (COLING)*, pages 2335–2344, 2014.

APPENDICES

Appendix A

COPYRIGHT AGREEMENTS



ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:

- Affordably priced publications
- Liberal Author rights policies
- Wide-spread, perpetual access to ACM publications via a leading-edge technology platform
- Sustainability of the good work of ACM that benefits the profession

ACM Author Rights

CHOOSE

Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.

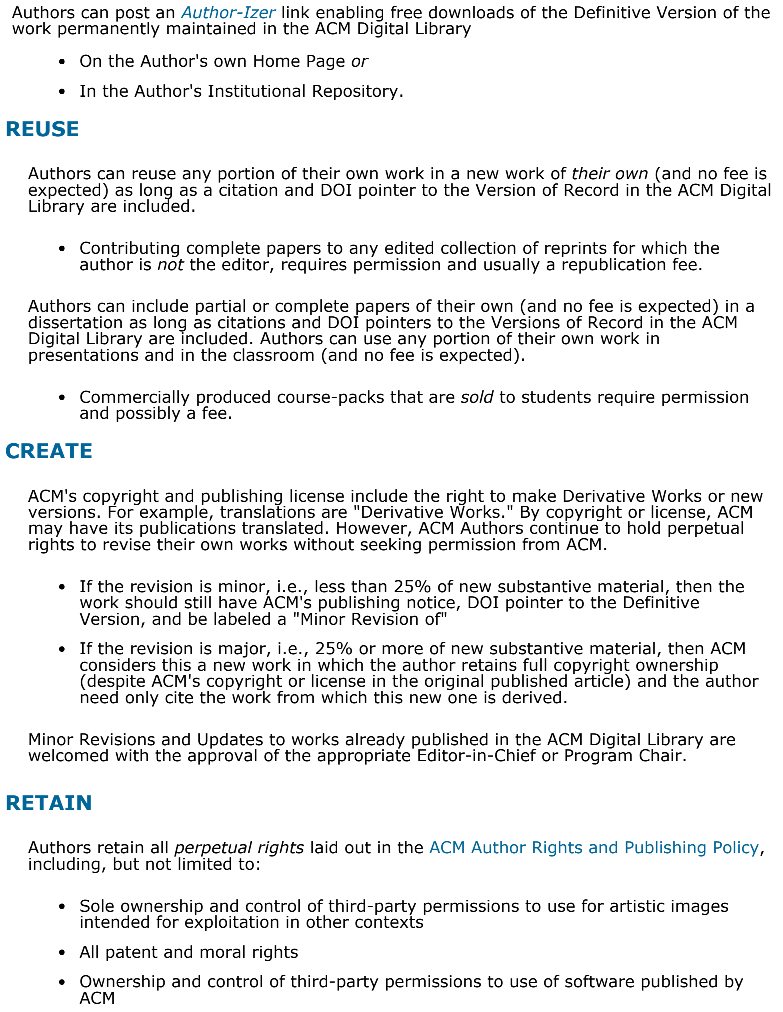
- Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM's traditional copyright transfer agreement.
- Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties.
- Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement. Those choosing to grant ACM a non-exclusive permission to publish may also choose to display a Creative Commons License on their works.

POST

Authors can post the accepted, peer-reviewed version prepared by the author-known as the "pre-print"-to the following sites, with a DOI pointer to the Definitive Version of Record in the ACM Digital Library.

- On Author's own Home Page *and*
- On Author's Institutional Repository *and*
- In any repository legally mandated by the agency funding the research on which the work is based *and*
- On any non-commercial repository or aggregation that does not duplicate ACM tables of contents, i.e., whose patterns of links do not substantially duplicate an ACM-copyrighted volume or issue. Non-commercial repositories are here understood as repositories owned by non-profit organizations that do not charge a fee for accessing deposited articles and that do not sell advertising or otherwise profit from serving articles.

Appendix A (Continued)



Authors can post an *Author-Izer* link enabling free downloads of the Definitive Version of the work permanently maintained in the ACM Digital Library

- On the Author's own Home Page *or*
- In the Author's Institutional Repository.

REUSE

Authors can reuse any portion of their own work in a new work of *their own* (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is *not* the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

- Commercially produced course-packs that are *sold* to students require permission and possibly a fee.

CREATE

ACM's copyright and publishing license include the right to make Derivative Works or new versions. For example, translations are "Derivative Works." By copyright or license, ACM may have its publications translated. However, ACM Authors continue to hold perpetual rights to revise their own works without seeking permission from ACM.

- If the revision is minor, i.e., less than 25% of new substantive material, then the work should still have ACM's publishing notice, DOI pointer to the Definitive Version, and be labeled a "Minor Revision of"
- If the revision is major, i.e., 25% or more of new substantive material, then ACM considers this a new work in which the author retains full copyright ownership (despite ACM's copyright or license in the original published article) and the author need only cite the work from which this new one is derived.

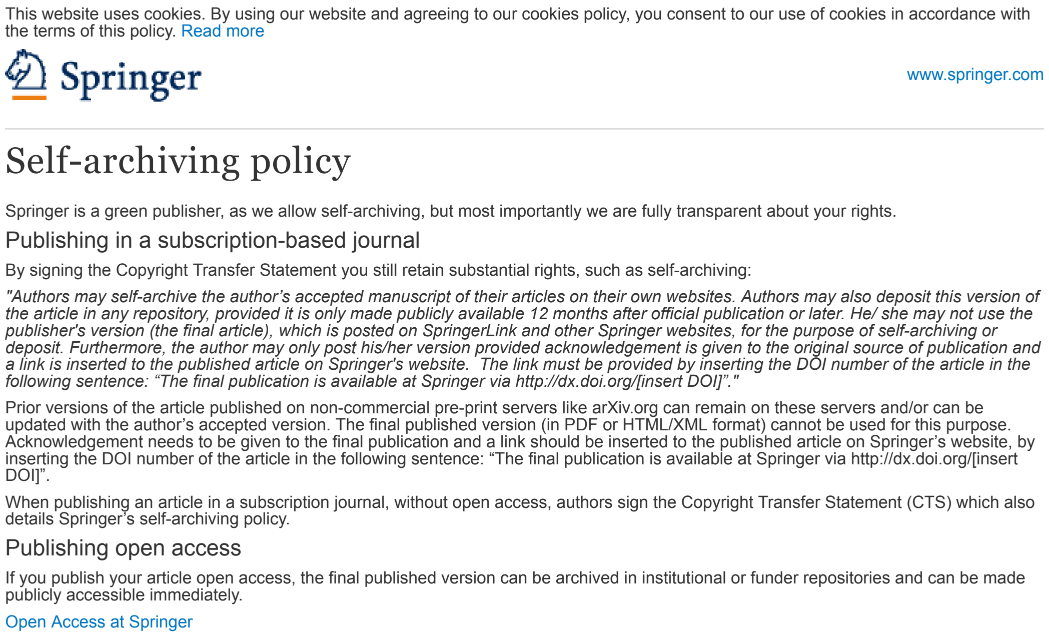
Minor Revisions and Updates to works already published in the ACM Digital Library are welcomed with the approval of the appropriate Editor-in-Chief or Program Chair.

RETAIN

Authors retain all *perpetual rights* laid out in the ACM Author Rights and Publishing Policy, including, but not limited to:

- Sole ownership and control of third-party permissions to use for artistic images intended for exploitation in other contexts
- All patent and moral rights
- Ownership and control of third-party permissions to use of software published by ACM

Appendix A (Continued)



This website uses cookies. By using our website and agreeing to our cookies policy, you consent to our use of cookies in accordance with the terms of this policy. Read more

www.springer.com

Self-archiving policy

Springer is a green publisher, as we allow self-archiving, but most importantly we are fully transparent about your rights.

Publishing in a subscription-based journal

By signing the Copyright Transfer Statement you still retain substantial rights, such as self-archiving:

"Authors may self-archive the author's accepted manuscript of their articles on their own websites. Authors may also deposit this version of the article in any repository, provided it is only made publicly available 12 months after official publication or later. He/ she may not use the publisher's version (the final article), which is posted on SpringerLink and other Springer websites, for the purpose of self-archiving or deposit. Furthermore, the author may only post his/her version provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be provided by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]"."

Prior versions of the article published on non-commercial pre-print servers like arXiv.org can remain on these servers and/or can be updated with the author's accepted version. The final published version (in PDF or HTML/XML format) cannot be used for this purpose. Acknowledgement needs to be given to the final publication and a link should be inserted to the published article on Springer's website, by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]".

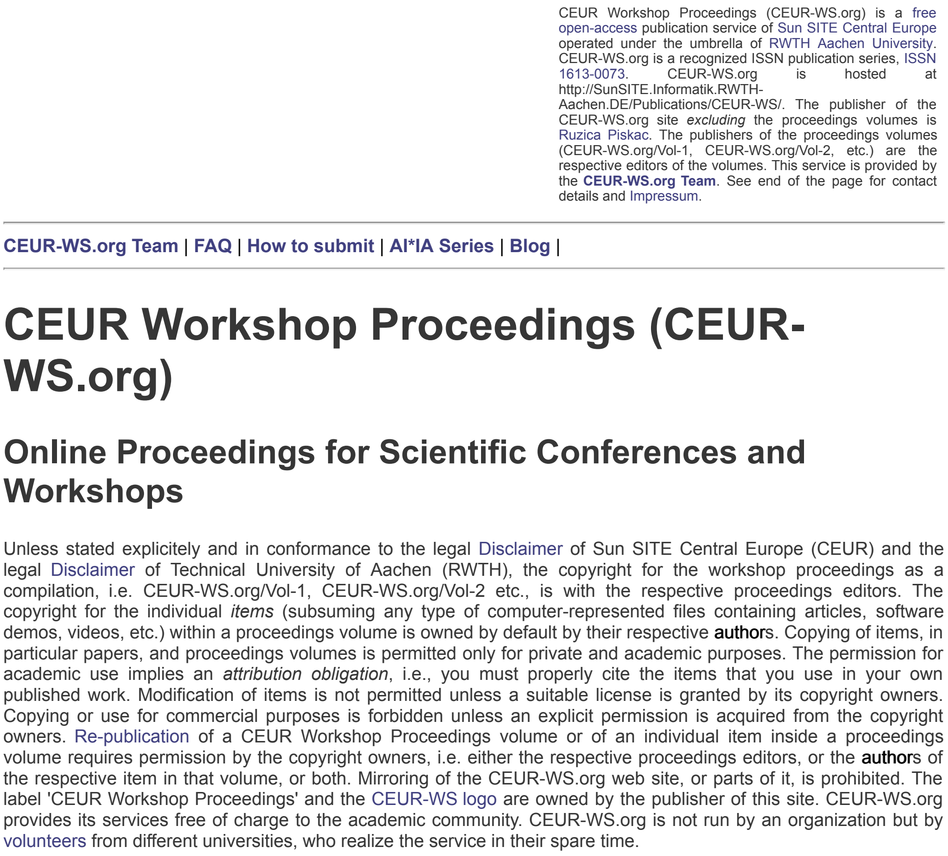
When publishing an article in a subscription journal, without open access, authors sign the Copyright Transfer Statement (CTS) which also details Springer's self-archiving policy.

Publishing open access

If you publish your article open access, the final published version can be archived in institutional or funder repositories and can be made publicly accessible immediately.

Open Access at Springer

Appendix A (Continued)



CEUR Workshop Proceedings (CEUR-WS.org) is a free open-access publication service of Sun SITE Central Europe operated under the umbrella of RWTH Aachen University. CEUR-WS.org is a recognized ISSN publication series, ISSN 1613-0073. CEUR-WS.org is hosted at http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/. The publisher of the CEUR-WS.org site *excluding* the proceedings volumes is Ruzica Piskac. The publishers of the proceedings volumes (CEUR-WS.org/Vol-1, CEUR-WS.org/Vol-2, etc.) are the respective editors of the volumes. This service is provided by the **CEUR-WS.org Team**. See end of the page for contact details and Impressum.

CEUR-WS.org Team | **FAQ** | **How to submit** | **AI*IA Series** | **Blog** |

CEUR Workshop Proceedings (CEUR-WS.org)

Online Proceedings for Scientific Conferences and Workshops

Unless stated explicitely and in conformance to the legal Disclaimer of Sun SITE Central Europe (CEUR) and the legal Disclaimer of Technical University of Aachen (RWTH), the copyright for the workshop proceedings as a compilation, i.e. CEUR-WS.org/Vol-1, CEUR-WS.org/Vol-2 etc., is with the respective proceedings editors. The copyright for the individual *items* (subsuming any type of computer-represented files containing articles, software demos, videos, etc.) within a proceedings volume is owned by default by their respective authors. Copying of items, in particular papers, and proceedings volumes is permitted only for private and academic purposes. The permission for academic use implies an *attribution obligation*, i.e., you must properly cite the items that you use in your own published work. Modification of items is not permitted unless a suitable license is granted by its copyright owners. Copying or use for commercial purposes is forbidden unless an explicit permission is acquired from the copyright owners. Re-publication of a CEUR Workshop Proceedings volume or of an individual item inside a proceedings volume requires permission by the copyright owners, i.e. either the respective proceedings editors, or the authors of the respective item in that volume, or both. Mirroring of the CEUR-WS.org web site, or parts of it, is prohibited. The label 'CEUR Workshop Proceedings' and the CEUR-WS logo are owned by the publisher of this site. CEUR-WS.org provides its services free of charge to the academic community. CEUR-WS.org is not run by an organization but by volunteers from different universities, who realize the service in their spare time.

VITA

Seyed Iman Mirrezaei was born in Tehran, Iran. He completed his master's degree in Computer Engineering at *Sharif University of Technology*, Tehran, Iran, in 2007 following which he worked as a *J2EE developer* in Tehran, Iran. Later, he joined *University of Illinois at Chicago* for his PhD's degree in Computer Science in 2011, where he has been completed his PhD's degree under supervision of Dr. Isabel F. Cruz, Professor of Computer Science.

A.1 Publications

1. S. I. Mirrezaei, B. Martins, I. F. Cruz, *A Distantly Supervised Method for Extracting Spatio-Temporal Information from Text*, ACM SIGSPATIAL, 2016.

2. S. I. Mirrezaei, B. Martins, I. F. Cruz, *The Triplex Approach for Recognizing Semantic Relations from Noun Phrases*, invited to the ESWC Workshops Post Proceedings with Best Papers, Springer, 2015.

3. S. I. Mirrezaei, B. Martins, I. F. Cruz, *The Triplex Approach for Recognizing Semantic Relations from Noun Phrases*, ESWC Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD), 2015.

4. I. F. Cruz, V. R. Ganesh, and S. I. Mirrezaei, *Semantic Extraction of Geographic Data from Web Tables for Big Data Integration*, ACM SIGSPATIAL Workshop on Geographic Information Retrieval (GIR), 2013.

5. S. I. Mirrezaei, J. Shahparian, M. Ghodsi, *A Topology-aware Load Balancing Algorithm for P2P Systems*, Conference on Digital Information Management (ICDIM), 2009.

6. S. I. Mirrezaei, J. Shahparian, M. Ghodsi, *RAQNet: A Topology-aware Overlay Network*, Autonomous Infrastructure Management and Security Conference (AIMS), 2007.