Compact Open Information Extraction: Methods, Corpora, Analysis

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
M. Sc. Kiril Gashteovski
aus Veles, Nordmazedonien

Mannheim, 2020

Dekan: Dr. Bernd Lübcke, Universität Mannheim
Referent: Prof. Dr. Rainer Gemulla, Universität Mannheim
Korreferent: Prof. Dr. Simone Paolo Ponzetto, Universität Mannheim
Tag der mündlichen Prüfung: 17.03.2021

To my loving wife Biljana Gashteovska

"I am preaching the message that, with apparently only one life to live on this earth, you ought to try to make significant contributions to humanity rather than just get along through life comfortably—that the life of trying to achieve excellence in some area is in itself a worthy goal for your life. It has been observed the true gain is in the struggle and not in the achievement—a life without a struggle on your part to make yourself excellent is hardly a life worth living."

Richard Hamming
"The Art of Doing Science and Engineering: Learning to Learn"

Acknowledgements

My personal story begins in the small village of Smojmirovo, located in the mountainous region of Maleshevo in North Macedonia, where I spent the first three years of my life. It was a long journey from there to earning a PhD degree in Germany. This journey, of course, was not going to be possible without the help and influence of many individuals. I was fortunate enough to have these people in my life. This part of the thesis, therefore, is about them.

First, I would like to thank my PhD advisor Prof. Rainer Gemulla for giving me the opportunity to pursue this intellectual growth and adventure. Under his mentorship, I learned how to think more critically, write clearer and speak better, which, in my opinion, are the major pillars of education. With such profound intellectual impact, Rainer remains one of the most influential individuals in my life, for which I am in eternal debt to him.

None of the published papers that are part of this thesis would have been possible without the help of all other collaborators. Therefore, I would like to thank Luciano del Corro, Sven Hertling, Samuel Broscheit, Anne Lauscher (who also translated the abstract of this thesis), Bhushan Kotnis, Sebastian Wanner and Christian Meilicke for being such great collaborators, and for being available when it counted.

The whole endeavour of this thesis was going to be much harder (probably even impossible) without a few laughs and motivational support. I want to thank all colleagues from the Data and Web Science group at the University of Mannheim for being a great bunch of people. I always enjoyed the semestrial BBQs, the annual Christmass parties, the daily lunch discussions with the people from our PI1 chair, the occasional birthday gatherings and the Friday NLP lunch with the cheerful group of Prof. Simone Paolo Ponzetto and Prof. Goran Glavaš. One special thanks goes to Martina Hey, for being such a good person and always helping me out navigating the complex administration life in Germany. I also want to thank Kaustubh Beedkar, for being an infinite source of motivation and encouragement during the first year of my PhD. One special thanks goes to NEC Labs Europe, where I am currently employed. Without the support of NEC Labs and—in particular—Mathias Niepert, the fifth chapter of this thesis would have been truly impossible to finish.

Looking further back, in many ways my intellectual path towards this thesis was also accompanied and encouraged by different teachers and mentors. Prof. Stefan Stefanov from

the South-West University “Neofit Rilski” in Bulgaria is one of the most inspiring teachers I’ve ever seen. Such love for clear and articulate teaching is rare, and I was fortunate enough to benefit from his lectures in linear programming and numerical methods. This was the first turning point in my education when I started to get interested into computer science more deeply. One thanks goes to Prof. Ivan Trenchev, who trusted me enough to involve me into a research project on bioinformatics, which was the first time I was exposed to a research environment. The combination of this project and the lectures of Prof. Stefanov inspired me to choose to study the topic of operations research for my master studies at the Aix-Marseille University in Marseille, France. I also want to thank Prof. Georgi Tuparov from the South-West University of Bulgaria for always challenging my opinions and for the great way of teaching his courses by combination of technical knowledge mixed with life stories and analogies, some of which I still quote occasionally. I also want to thank Prof. Yann Vaxès from the Aix-Marseille University, Prof. Thomas Guyet from Institut Agro in Rennes, France and Dr. René Quiniou from INRIA in Rennes, France for encouraging me to pursue a research-oriented career.

After the master studies, my first employment was in a small company—OmegaCube—which was located in Marseille, France. This was the first time I was exposed to a data science project, and, in turn, changed the course of my career for good. After this experience, I knew that there is no turning back for me, that everything I do next will be related in some particular aspect of data science. Therefore, I want to thank Guillaume Seigneuret and Xavier Roger-Machart for giving me the first opportunity to get into data science and for all the incredible amount of freedom I experienced during this project. The skills I learned during this project were essential for my future career.

Next, I want to thank my family for all the moral support throughout my journey. My family members consider the completion of this thesis as a great family victory, since none of my parents—nor anyone before them—ever set foot in a university. My father Marjan (Max) Gashteovski was the person who always nurtured my passion for knowledge ever since I was a kid. As a child, I used to admire his passion for reading, and he successfully “infected” me with it. My mother, Snezhanka Gashteovska, always took care while I was a child that I have quiet time for studying and also thought me the value of being disciplined with my studies.

Finally, I want to thank my loving wife, Biljana Gashteovska, for being the most supportive partner I could ever wish for. Her emotional support meant the world to me during the tough times of the writing of this thesis. I was fortunate enough to meet her, fall in love with her and marry her—all during the time of this thesis. For all her support and love, I dedicate this thesis to her.

Abstract

Most existing data is stored in unstructured textual formats, which makes their subsequent processing by computers more difficult. The Open Information Extraction (OpenIE) paradigm aims at structuring the knowledge that is contained in text into more machine readable formats. An OpenIE system (usually) extracts triples—*("subject"; "relation"; "object")*—from natural language text in an unsupervised manner, without having predefined relations. OpenIE extractions are used for improving deeper language-understanding tasks, including KB population, link prediction and text comprehension.

A common problem for such systems is that they often extract triples which contain unnecessarily detailed constituents. For instance, the phrases *"the great Richard Feynman"* and *"Richard Feynman"* have the same meaning, but the first phrase contains redundant words—*"the"* and *"great"*—that do not alter the meaning of the head phrase *"Richard Feynman"*. Such redundant words pose difficulties for using OpenIE in downstream tasks, such as linking entities for KB population. In this thesis, we propose MinIE, an OpenIE system which aims to remove words from the triples that are considered to be overly-specific without damaging the triple's semantics. The methods proposed in MinIE are domain-independent and could in principle be integrated into any other OpenIE system.

OpenIE extractions are most useful when they are available in large quantities. Our second contribution, therefore, is OPIEC, which is the largest publicly available OpenIE corpus to date (containing 341M triples). OPIEC was constructed from the entire English Wikipedia and it contains the links found in the Wikipedia articles, thus reducing ambiguity in certain cases. Such OpenIE triples with unambiguous arguments are useful for bootstrapping OpenIE extractors as well as for downstream tasks such as KB population.

Our final contribution is an analysis of OPIEC. Such analysis is difficult to perform due to the openness and ambiguity of OpenIE extractions. Therefore, we compared the content of OPIEC with reference KBs (DBpedia and YAGO), which are not ambiguous and are also constructed from Wikipedia. Our analysis is (mostly) manual and reveals findings about semantic relatedness between OpenIE corpora and KBs, which are important for downstream tasks such as KB population (e.g., the study suggests that most knowledge found in OpenIE triples is relevant for the current KBs and it is not present in the KBs).

Kurzfassung

Der Großteil der existierenden Daten liegt in unstrukturierten textuellen Formaten vor, was die anschließende rechnergestützte Verarbeitung erschwert. Das Open Information Extraction (OpenIE)-Paradigma zielt daher darauf ab, das Wissen, welches im Text enthalten ist, in ein maschinenlesbares Format zu strukturieren. Hierbei extrahiert ein OpenIE-System (üblicherweise) Tripel—*("Subjekt"; "Relation"; "Objekt")*—aus natürlichsprachigem Text in unüberwachter Art, ohne dabei auf vordefinierte Relationen zurückzugreifen. Die resultierenden OpenIE-Extraktionen werden dazu verwendet, um Aufgaben des tieferen Sprachverstehens zu verbessern, z.B. zur Population von Knowledge Bases (KBs), zum Vorhersagen von Links und zum Textverstehen. Ein bekanntes Problem solcher Systeme ist jedoch, dass sie oft Tripel extrahieren, welche unnötige, detaillierte Bestandteile enthalten. Zum Beispiel haben die beiden Ausdrücke *"the great Richard Feynman"* und *"Richard Feynman"* die gleiche Bedeutung, aber der erste Ausdruck enthält redundante Wörter—*"the"* und *"great"*—, die Bedeutung des Head-Ausdrucks "Richard Feynman" nicht verändern. Solche redundanten Wörter stellen die Verwendung von OpenIE in Downstream-Aufgaben, wie z.B. beim Verknüpfen von Entitäten für die Population von KBs, vor Schwierigkeiten.

In dieser Thesis schlagen wir MinIE vor: Ein OpenIE-System, welches zum Ziel hat, Wörter, die als überspezifisch angesehen werden, aus den Tripeln zu entfernen, ohne dabei die Semantik des Triples zu beschädigen. Die mit MinIE vorgeschlagenen Methoden sind domänenunabhängig und können prinzipiell in jedes andere OpenIE-System integriert werden.

OpenIE-Tripel sind am nützlichsten, wenn sie in großen Mengen vorhanden sind. Unser zweiter Beitrag ist daher OPIEC, welches mit 341 Mio. Tripeln das derzeit größte öffentlich verfügbare OpenIE-Korpus ist. OPIEC wurde aus der englischsprachigen Wikipedia erzeugt und enthält die Verknüpfungen aus Wikipedia, womit Ambiguität in manchen Fällen reduziert wird. OpenIE-Tripel mit eindeutigen Argumenten sind nützlich, um OpenIE-Extraktoren zu bootstrappen und um Downstream-Aufgaben, wie z.B. die Population von KBs, besser zu unterstützen.

Unser finaler Beitrag ist eine Analyse von OPIEC. Aufgrund der Offenheit und Ambiguität von OpenIE-Extraktionen ist eine solche Analyse schwierig durchzuführen. Daher

vergleichen wir den Inhalt von OPIEC mit Referenz-KBs (DBpedia und YAGO), welche nicht uneindeutig sind und auch aus Wikipedia konstruiert wurden. Unsere (hauptsächlich) manuelle Analyse offenbart semantische Zusammenhänge zwischen OpenIE-Korpora und KBs, welche wichtig für Downstream-Anwendungen, wie z.B. Population von KBs, sind. So schlagen die Ergebnisse der Studie beispielsweise vor, dass das meiste Wissen, welches in OpenIE-Triplen vorhanden ist, relevant für aktuelle KBs ist, aber nicht in diesen gefunden werden kann.

Table of contents

Chapter 1

Introduction

Open Information Extraction (OpenIE) [Banko et al. 2007] is the task of extracting information from natural language text data into machine-readable format in an unsupervised, domain-independent manner. In contrast to traditional IE systems, OpenIE systems do not require an upfront specification of the target schema—e.g., target relations—or access to background knowledge; e.g., a Knowledge Base (KB). Instead, extractions are (usually) represented in the form of surface subject-relation-object triples. Consider the input sentence *"Bill Gates, who is the co-founder of Microsoft, lives in Seattle"*. An OpenIE system should extract the following extractions: *("Bill Gates"; "is co-founder of"; "Microsoft")* and *("Bill Gates"; "lives in"; "Seattle")*. OpenIE extractions serve as an input for deeper natural language understanding tasks such as relation extraction [Riedel et al. 2013; Petroni et al. 2015], automated knowledge base construction [Dong et al. 2014], question answering [Fader et al. 2014], word analogy [Stanovsky et al. 2015], information retrieval [Löser et al. 2011; Kadry and Dietz 2017] and knowledge base population [Lin et al. 2020].

One common problem of OpenIE systems is that they extract triples which are considered to be *overly-specific* [Fader et al. 2011]. An OpenIE triple is considered as *overly-specific* if it contains words such that, if removed, the semantics of the triple remains unchanged. Consider the OpenIE triple *("The great Michael Jordan"; "grew up in"; "Wilmington")*. This triple is overly specific, because if we remove the words *"the great"* from the subject, the triple would not lose its meaning (*"the great"* is merely a detail about the entity *"Michael Jordan"*). Once we remove such overly-specific detailed words, we get a triple which we consider to be *more compact*. Thus, the triple *("Michael Jordan"; "grew up in"; "Wilmington")* is more compact (for more elaborate discussion on compactness, refer to Section 2.1.2). Overly-specific triples may pose difficulties for their use in downstream tasks. For example, Lin et al. [2020] report that such lack of compactness in the extractions produced by some OpenIE systems posed difficulties for linking the entities of OpenIE triples to a KB, which in

turn causes problems for the task of KB population. Therefore, aiming for compactness of OpenIE extractions could result into producing extractions which are potentially more useful for downstream tasks.

As a first contribution of this thesis, we propose MinIE [Gashteovski et al. 2017], an OpenIE system that aims at producing more compact OpenIE extractions. With MinIE, we propose methods for producing more compact OpenIE extractions that are generated by prior OpenIE methods. In particular, MinIE is built on top of ClausIE [Del Corro and Gemulla 2013], which is an OpenIE system that extracts OpenIE tuples with high precision and recall. ClausIE, however, produces overly-specific extractions. To address this issue, MinIE uses the extractions from ClausIE as an input, and subsequently processes them for compactness. Our experimental study shows that the methods for compactness proposed in MinIE do not hurt the precision of the extractions significantly. Even though MinIE is based on ClausIE, the methods for compactness proposed by MinIE are domain independent and can be, in principle, applied to other OpenIE systems. Subsequent work demonstrated the usefulness of the extractions produced by MinIE w.r.t. other downstream tasks, including KB population [Lin et al. 2020], fact salience [Ponza et al. 2018] as well as for specializing the compactness methods to specific domain [Lauscher et al. 2019]. The details about the methods for compactness in OpenIE are discussed in Chapter 3.

Once OpenIE systems are applied on large text corpora, they can produce massive amounts of OpenIE triples [Gashteovski et al. 2019]. Such large OpenIE corpora are used for many downstream tasks, including question answering [Yan et al. 2018], automated knowledge base construction [Dong et al. 2014] and open link prediction [Broscheit et al. 2020]. Moreover, OpenIE corpora are used for more human-centric tasks as well, including text summarization with salient facts [Ponza et al. 2018; Sheng and Xu 2019; Sheng et al. 2020] or explainability of entity-ranking for information retrieval [Kadry and Dietz 2017]. For these reasons, it is important to have large publicly-available OpenIE corpora, which can be used for many different downstream tasks. Therefore, the second contribution of this thesis is OPIEC [Gashteovski et al. 2019], the largest publicly-available OpenIE corpus to date, which contains more than 341 million OpenIE triples. Subsequent work showed that OPIEC is useful resource for downstream tasks such as open link prediction [Broscheit et al. 2020] and entity aspect linking [Nanni et al. 2019]

OPIEC was constructed by running a version of MinIE on the textual part of the articles of the entire English Wikipedia. To make the corpus less ambiguous, we retained the original links found in the text in the Wikipedia articles. Thus, some of the triples have disambiguated arguments. Reducing such ambiguity in OpenIE triples is useful for downstream tasks such as open link prediction [Broscheit et al. 2020] and knowledge base unification [Delli Bovi

et al. 2015a]. Because much of the OpenIE triples are noisy, we created two less-noisy subcorpora: OPIEC-Clean and OPIEC-Linked. In OPIEC-Clean we kept the OpenIE triples that contain arguments which are either entities or concepts and in OPIEC-Linked we kept the triples which contain disambiguated arguments on both sides. The details about the construction of OPIEC, as well as its statistics, are discussed in Chapter 4.

Because large OpenIE corpora are used in different downstream tasks [Mausam 2016; Gashteovski et al. 2019], it is important to have an intrinsic in-depth semantic analysis of such corpora, which is not dependent on a particular downstream task. Such semantic analysis can provide insights about the information content of the OpenIE corpus. Therefore, the third contribution of this thesis is a semantic analysis of such large OpenIE corpora; namely, a semantic analysis of OPIEC. Large OpenIE corpora, however, can be quite ambiguous, because their extractions are merely surface patterns, which makes their semantic analysis difficult. For example, the phrase *"Michael Jordan"*—which could be an argument in an OpenIE triple—refers to 13 people in Wikipedia. Similarly, the open relations in the OpenIE triples are strings that do not have precise semantics. Such ambiguity in OpenIE extractions makes the semantic analysis of large OpenIE corpora difficult. For the purpose of the semantic analysis, we used OPIEC-Linked to reduce the ambiguity. We then compared the OPIEC-Linked corpus with the DBpedia KB, because KBs are resources that contain triples which are not ambiguous; i.e., both the relations and the arguments are semantically precise concepts. Moreover, OpenIE corpora are often used in combination with KBs for improving the performance of different downstream tasks, which is another reason why such semantic analysis that is not dependent on a particular downstream task is important.

When the arguments are disambiguated, OpenIE corpora are aligned with KBs for performing downstream tasks, such as KB population [Lin et al. 2020], slot filling [Angeli et al. 2015] or even for learning extraction rules for improving OpenIE systems themselves through the distant supervision assumption [Weld et al. 2009; Wu and Weld 2010; Mausam et al. 2012; Yahya et al. 2014; Saha et al. 2017]. Such alignments are usually measured w.r.t. the downstream task at hand [Angeli et al. 2015; Lockard et al. 2019] and do not provide in-depth semantic analysis of the OpenIE corpus. In Chapter 5, we perform both automated and manual semantic analyses of OPIEC w.r.t. reference KBs. These analyses are not dependent on a particular downstream task. We found that it is difficult to map open relation to KB relation due to high ambiguity and that it is safer for such mappings to be done on instance level. For the distant-supervision assumption in OpenIE, we found that it holds in general, though the OpenIE triples are usually more specific than the KB facts. We also observed that while many OpenIE triples can be expressed with a single KB fact, they often cannot be fully expressed. We found, however, that the use of KB formulas significantly

improves the expressibility of an OpenIE triple w.r.t. reference KB. Next, we observed that the information contained in most OpenIE triples that are relevant for the KB are not present in the KB. This shows the potential of knowledge that is contained in OpenIE corpora which can be harnessed for the relevant KBs. Finally, we made experiments on the transferibility of our semantic analysis. Our experiments suggest that our findings generally transfer over to other OpenIE systems and are not limited only to MinIE.

The rest of the thesis is structured as follows: in Chapter 2 we discuss some preliminaries for understanding the rest of the thesis as well as related work; Chapter 3 discusses methods for compact OpenIE with MinIE; In Chapter 4 we discuss OPIEC—an OpenIE corpus—and perform in-depth data profiling; In Chapter 5 we analyse how the OpenIE extractions of OPIEC are aligned with structured KBs; Finally, in Chapter 6 we conclude the thesis and speculate about possible future directions of research.

Chapter 2

Preliminaries and Related Work

In this chapter, we discuss related work for OpenIE as well as some preliminaries that are necessary for understanding the rest of the thesis. We first discuss Open Information Extraction (OpenIE) in general (Section 2.1). In particular, we discuss common extraction formats, compact OpenIE, methodologies for constructing OpenIE systems, OpenIE systems in different natural languages and methods for evaluation. Next, we discuss publicly available OpenIE corpora, their main properties and how they are used (Section 2.2). Finally, we discuss the concept of Knowledge Bases (KBs)—as well as several publicly available KBs—and how they play a role within the context of OpenIE (Section 2.3).

2.1 Open Information Extraction

As discussed in Chapter 1, OpenIE is the task of extracting relations and their arguments from natural language sentence in unsupervised manner. For instance, given an input sentence *"Barack Obama, who served as President of the United States, was born in Honolulu"*, an OpenIE system should extract the following triples: *("Barack Obama"; "served as"; "President of the United States")* and *("Barack Obama"; "was born in"; "Honolulu")*.

Most commonly, OpenIE systems extract schemaless triples from an input sentence. In principle, OpenIE representations represent knowledge that is found in natural language sentences into structured machine-readable form. Contrary to traditional information extraction pipelines, OpenIE systems do not require predefined schemas. Standard IE systems are limited by the predefined schemas, which makes them unable to extract information that goes beyond the schemas. On the other hand, OpenIE systems, in principle, are able to extract any form of relation between two entities, which makes them scalable w.r.t. the diversity of natural language.

OpenIE extractions are useful for numerous downstream tasks, including question answering [Yan et al. 2018; Khot et al. 2017; Fader et al. 2013], information retrieval [Kadry and Dietz 2017; Löser et al. 2011], slot filling [Yu et al. 2017; Angeli et al. 2015; Soderland et al. 2015a;b; 2013], event schema induction [Balasubramanian et al. 2013], text summarization [Ponza et al. 2018], knowledge base population [Lin et al. 2020; Wolfe et al. 2017] entity aspect linking [Nanni et al. 2019], link prediction [Gupta et al. 2019] and open link prediction [Broscheit et al. 2020].

2.1.1 Extraction Formats

OpenIE Triples

Most OpenIE systems output their extractions as triples having the form of *("subject"; "relation"; "object")*. The main reason for such output format is because this particular structure makes the OpenIE outputs useful for downstream semantic tasks, such as text comprehension tasks [Stanovsky et al. 2015], knowledge base population [Lin et al. 2020], link prediction [Gupta et al. 2019] and open link prediction [Broscheit et al. 2020]. OpenIE triples, however, may be accompanied with further context in the form of semantic annotations, which help in capturing a more precise meaning of the information contained in the triple.

Contextual Semantic Annotations

Even though OpenIE triples are used in many downstream tasks and can (usually) represent a coherent piece of information, still sometimes it is hard to represent complex information within a single OpenIE triple. For these reasons, some OpenIE systems introduced semantic annotations, which provide further contextual information that accompanies the OpenIE triple.

One of the earliest such OpenIE system is OLLIE [Mausam et al. 2012], which introduced the notions of *Clausal modifier* and *attribution* in the context of OpenIE extractions. The clausal modifier annotation supplies context for the extraction, which is provided by a dependent clause. The *attribution* is the provider of the information that is contained in the extraction (if found in the sentence). Consider the input sentence *"Angela Merkel believes that between 60% and 70% of the population will be infected by coronavirus if no action is taken"*. From that sentence, OLLIE extracts:

> *("between 60% and 70% of the population"; "will be infected by"; "coronavirus")*
>
> *Clause Modifier: "if no action is taken"*

Attribution: "Angela Merkel believes"

Other OpenIE systems also exploited the annotations for clausal modifier (e.g. RelNoun [Pal and Mausam 2016] and CALMIE [Saha and Mausam 2018]) and attribution (e.g. MinIE [Gashteovski et al. 2017]). In similar spirit, contextual annotations could represent other types of semantic annotation about the OpenIE triple, including polarity, modality [Gashteovski et al. 2017], space, time [Christensen et al. 2011], condition and contrast [Cetto et al. 2018]. In certain scenarios, such more complex representations of OpenIE extractions are useful for downstream tasks, such as question answering [Bhutani and Jagadish 2019].

N-ary Extractions

To capture more complicated information content from a sentence, other OpenIE systems structure their outputs into n-ary tuples (e.g. KrakeN [Akbik and Löser 2012], ClausIE [Del Corro and Gemulla 2013] and OpenIE 4 [Mausam 2016]). The goal of such representations is to be able to represent information which cannot be easily represented with binary relation (i.e. a triple). For example, suppose we want to extract information from the following sentence: *"NZ Natural is brand of bottled water collected in New Zealand"*. We could represent the information from this sentence either as a triple or an n-ary tuple:

("NZ Natural"; "is brand of bottled water collected in"; "New Zealand")

("NZ Natural"; "is brand of"; "bottled water"; "collected in"; "New Zealand")

In the first extraction (the triple), the entities *NZ Natural* and *New Zealand* are related with a rather complex relation. Such relations are not compact, because they contain many different detailed concepts and predicates (e.g. the relation in the triple involves the concepts for *brand* and *bottled water* as well as predicates indicating *is-a* relation and *something being collected in a location*). In the second case (the n-ary tuple), the same entities are still related (*NZ Natural* and *New Zealand*), though the information representation is more expressible. This representation enables to separate the different concepts and the different predicates from the complex relation into separate chunks.

Nested Extractions

Another way to structure more complex information of natural language sentences is by representing the OpenIE output as nested extractions. In fact, such systems extract *triples* from the sentences, though an argument might be a whole triple itself (hence, nested structures). Consider the sentence *"After celebrating the elections victory, Angela Merkel gave a speech"*. An OpenIE system might extract the following nested extractions:

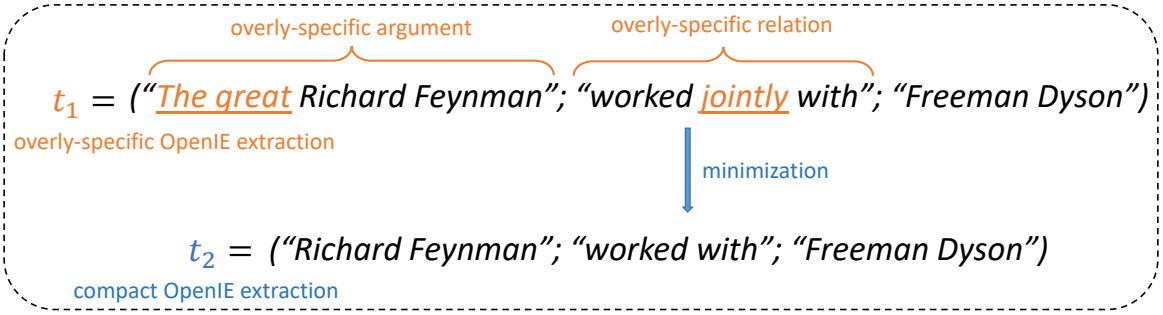


Fig. 2.1 Example of overly-specific OpenIE extraction and its corresponding compact OpenIE extraction

t_1: *(“Angela Merkel”; “be celebrating”; “elections victory”)*

t_2: *(“Angela Merkel”; “gave”; “speech”)*

t_3: *(t_2; “after”; t_3)*

Nested OpenIE systems include CSD-IE [Bast and Haussmann 2013], NestIE [Bhutani et al. 2016] and Graphene [Cetto et al. 2018]. Bhutani and Jagadish [2019] suggest that such nested OpenIE extractions are useful for question answering.

2.1.2 Compact OpenIE

A common problem for OpenIE triples is that they can often be overly-specific.[1] An OpenIE triple is considered as *overly-specific* if it comprises a relation (or an argument), which contains words that, when removed, do not change the semantics of the triple (i.e. they are redundant w.r.t. the semantics of the triple).

Consider the example shown on Figure 2.1. The OpenIE triple t_1—*(“The great Richard Feynman”; “worked jointly with”; “Freeman Dyson”)*—is overly-specific, because it contains an overly-specific argument (*“The great Richard Feynman”*) and an overly-specific relation (*“worked jointly with”*). For the argument, the words *“The great”* are merely details of the noun phrase, which do not change the meaning of the phrase if they are removed (i.e. *“the great Richard Feynman”* has virtually the same meaning as *“Richard Feynman”*). In the same manner, the open relation *“worked jointly with”* has the same meaning as the open relation *“worked with”*.

The overly-specific OpenIE triples can be reduced to *compact OpenIE extractions* through the process of *minimization* (Figure 2.1). In our working example on Figure 2.1, the overly-

[1] For simplicity, in this section we discuss only the case for OpenIE triples. Note that the claims in this section hold true for other OpenIE extraction formats as well (e.g. for n-ary OpenIE extractions).

specific words (*"the great"* from the subject and *"jointly"* from the relation) are dropped (i.e. the triple is *minimized*). The resulting OpenIE triple t_2—*("Richard Feynman"; "worked with"; "Freeman Dyson")*— is *compact* w.r.t. the original overly-specific triple (t_1), because we eliminated words from t_1 without changing the triple's meaning. Consequently, we can compare compactness (i.e. one triple might be more compact than another). Consider the OpenIE triple t_3 = *("Richard Feynman"; "worked jointly with"; "Freeman Dyson")*. The OpenIE triple t_3 is more compact than t_1, but less compact than t_2.

MinIE [Gashteovski et al. 2017] is an OpenIE system that aims to produce more compact extractions. In Chapter 3, we discuss in details how MinIE provides methods for minimizing OpenIE extractions.

2.1.3 Methodologies for Constructing OpenIE Systems

The methodologies for constructing OpenIE systems could be split into three major categories: methods using hand-crafted rules, bootstrapping methods and neural models. In what follows, we briefly discuss methods for OpenIE from each of these categories of methods, as well as their strengths and weaknesses.

Methods using Hand-Crafted Rules

Such methods usually exploit syntactic or semantic information of the input sentence, which is subsequently used to hand-craft extraction rules [Del Corro and Gemulla 2013; Cetto et al. 2018]. The advantages of these approaches are that they are domain independent and they use knowledge from linguistics which is relatively simple to implement. The drawback of these methods, however, is that they require laborious process of designing and testing of such linguistic rules. Moreover, such methods rely on prior natural language processing systems that produce the syntactic structure needed for making the extractions. Consequently, any error produced by the syntactic parser propagates further in the OpenIE pipeline, which leads to an error in the OpenIE extraction itself.

Early methods were exploiting shallower syntactic information (e.g. POS tags and noun phrase chunks) [Fader et al. 2011]. Other work explored the use of constructing extraction rules with deeper syntactic information such as constituency [Van Durme and Schubert 2008] and dependency parsing [Gamallo et al. 2012; Del Corro and Gemulla 2013; Gashteovski et al. 2017]. Compared to the methods that rely on POS tag sequences, methods based on dependency parse trees were shown to improve recall, because they capture deeper dependencies between the entities within the sentence. More recently, some methods [Cetto

et al. 2018] exploit Rhetorical Structure Theory [Mann and Thompson 1988] to improve the precision and semantically enrich the extractions.

Besides the syntactic rules for OpenIE, other systems exploit semantic information as well. Such semantic information informs the extraction rules. For example, SRL-IE [Christensen et al. 2011] uses Semantic Role Labeling (SRL) as a preprocessing step, where the SRL predicates and arguments correspond to open relations and OpenIE arguments respectively. Other approaches use entity-centric rules for extracting information, such as Named-Entity Recognition (NER) tags [Gashteovski et al. 2017] or disambiguation links for the entities [Moro and Navigli 2013; 2012; Delli Bovi et al. 2015b; Bovi et al. 2015; Gashteovski et al. 2019].

Bootstrapping Methods

Bootstrapping methods learn extraction rules—usually added in addition to already existing hand-crafted rules—through the use of the distant supervision assumption (DSA) [Mintz et al. 2009]. The DSA was originally used for standard relation extraction tasks. In particular, such methods typically make use of high-confidence extractions from prior methods—extracted from a large corpus of textual documents—, which serve as ground truth. Next, they search in the corpus for sentences that contain the same argument pairs as the ground truth OpenIE triples. The assumption is that the selected sentences express the same information as the ground truth triples. This suggests that the open pattern (e.g. the shortest path in the dependency parse tree between the entities of the triple) is an extraction pattern that can extract a triple. These patterns are stored as positive data, which are subsequently used for learning extraction rules.

The advantage of such methods is that, unlike the OpenIE systems that use hand-crafted rules, they do not require the time-consuming construction and testing of hand-crafted rules. Rather, they aim at learning such rules automatically. One drawback of such methods, however, is that they cannot learn sufficiently effective extraction rules if the used corpus is small. If, however, the corpus is too large, then there is a risk of concept drift. Other limitation is the quality of the seed extractions themselves. If the seed extractions are noisy, then the learned rules are going to be noisy as well. Moreover, if the seed extractions are not linguistically diverse enough, then the learned rules might be too homogeneous, thus hurting recall points of the OpenIE system.

One of the first such OpenIE systems is OLLIE [Mausam et al. 2012]. OLLIE uses high-confidence extractions from the OpenIE system ReVerb [Fader et al. 2011] as seed extractions, which are extracted from the large corpus ClueWeb09 [Callan et al. 2009]. Besides OLLIE, other OpenIE systems follow similar strategies for learning extraction patterns. Weld et al.

[2009] and Wu and Weld [2010] use Wikipedia as a source for distant supervision in order to learn dependency parse extraction patterns. ReNoun [Yahya et al. 2014] uses user queries of a search engine to learn extraction patterns, which target OpenIE triples with noun-mediated relations (e.g. *("Justin Trudeau"; "prime minister"; "Canada")*). NestIE [Bhutani et al. 2016] learns extraction patterns for nested OpenIE extractions (explained in section 2.1.1) and BONIE [Saha et al. 2017] learns extraction patterns that are tailored for numerical OpenIE extractions (e.g. *("Rhine"; "has length of"; "1,230 km")*). While most of these methods focus on learning dependency parse patterns, Gotti and Langlais [2019] follow the same strategy, though use sequence patterns of lemmas for distant supervision. The previously described methods work on the sentence level, though Jiang et al. [2017] (MetaPAD) and Zhu et al. [2019] (ReMine) proposed distantly-supervised frameworks for OpenIE that take into account the context provided by corpus statistics.

Neural Models

The prior methods discussed so far—methods using hand-crafted rules and bootstrapping methods—have been the dominant approaches for constructing OpenIE systems. In recent years, however, researchers turned to the idea of training neural models for OpenIE. Such models are trained directly from the the input text and do not rely on complex feature selection. They avoid, therefore, the drawback of error propagation in the OpenIE extractions produced by prior feature processing (e.g., errors produced by the dependency parser of the input sentence). Other advantage is that they do not require generating hand-crafted rules, which is usually very time consuming process.

Such methods, however, suffer from several drawbacks. First, they are biased towards the domain of the corpus that they are trained on. Second, when such methods produce an error, it is not clear why they produced such extraction, thus making the improvements of the models harder compared to rule-based methods. Finally, the major bottleneck for such approaches is the lack of large amounts of high-quality training data for the OpenIE task. Generating golden OpenIE data usually requires manual expert labor, because the task is not trivial enough for crowd-sourcing.

To construct training data for training a neural model for OpenIE, several approaches were proposed. Stanovsky et al. [2018] use datasets from other tasks—Question-Answer Driven Semantic Role Labeling (QA-SRL) [He et al. 2015] and Question-Answer Meaning Representation (QAMR) [Michael et al. 2018]— which are then automatically converted to OpenIE training and test data. Roy et al. [2019] report that they found mistakes in this particular dataset (e.g. some open relations were not covered), mostly because its original goal is to serve as training data for other tasks. Therefore, they manually corrected the dataset

from the noise and used this corrected dataset as training and test data. Cui et al. [2018] extract triples from Wikipedia with OpenIE 4 and keep the triples with high confidence scores (> 0.9) as positive training data. Similarly, Zhan and Zhao [2020] used the same dataset, though they also kept low-confidence triples, because they found that particular kinds of low-confidence triples are also correctly extracted (e.g. they found that many triples which have a personal pronoun as an argument gets low confidence value, even though many of them are correctly extracted). The strategy proposed by Cui et al. [2018] was followed by Kolluru et al. [2020], though besides OpenIE 4, they combine extractions from different OpenIE systems. Other approaches used large crowd-sourced dataset (SAOKE) of almost 50,000 sentence-extraction pairs in Chinese language [Sun et al. 2018a;b].

To train neural models for OpenIE, researchers are treating OpenIE either as a sequence tagging or as a sequence generation problem. For example, Stanovsky et al. [2018] formulate the problem of OpenIE as sequence tagging problem. To this end, they propose RNN architecture, which was motivated by prior work of semantic role labeling [Zhou and Xu 2015; He et al. 2017]. Roy et al. [2019] also treat OpenIE as a sequence tagging problem. Zhan and Zhao [2020] propose a span-selection model for OpenIE, which is a version of a sequence labeling problem. Other line of work treats OpenIE as a sequence generation problem [Cui et al. 2018; Sun et al. 2018a;b; Kolluru et al. 2020].

2.1.4 OpenIE Systems in Different Languages

Most of the currently available OpenIE systems are constructed to extract information from natural language sentences that are written in English. Besides the OpenIE systems for English, there has been increased interest in constructing cross-lingual and multi-lingual OpenIE systems [Faruqui and Kumar 2015; Gamallo and Garcia 2015; Zhang et al. 2017; Harting et al. 2020; Ro et al. 2020] as well as OpenIE systems tailored for specific languages other than English, including German [Falke et al. 2016; Bassa et al. 2018], Portuguese [de Oliveira et al. 2017; Sena et al. 2017; Glauber et al. 2018], Italian [Guarasci et al. 2019; 2020], Chinese [Qiu and Zhang 2014; Jia et al. 2018a; Wang et al. 2019a], Korean [Nam et al. 2015], Indonesian [Romadhony et al. 2018] and Persian [Rahat and Talebpour 2018b;a; Rahat et al. 2018; Saheb-Nassagh et al. 2020].

In this thesis, we focus on compact Open Information Extraction for the English language. Our methods are partially informed by linguistic properties of the English language and may not apply to other languages. We believe that investigating methods for compact OpenIE in other languages is interesting direction for future work.

2.1.5 Evaluation

OpenIE systems are evaluated either intrinsically or extrinsically. In this section we discuss both approaches.

Intrinsic Evaluations

In intrinsic evaluations, the experiments are performed either manually or automatically. In manual intrinsic evaluation, an input sentence and its OpenIE extractions are shown to human labelers. Then, they annotate each extraction w.r.t. the input sentence as *correctly extracted* or *incorrectly extracted* according to an annotation guideline (for example guideline, see Appendix A). Such evaluations are ideal in terms of quality, because the judgment of the correctness of OpenIE extractions require deep semantic understanding of both the input sentence and the corresponding extraction. Such complex semantic understanding can be achieved best by humans. For this reason, we used (mostly) manual intrinsic evaluations in this thesis.

As for automatic intrinsic evaluation, several benchmarks were proposed, where the OpenIE systems are compared against extractions that are considered as ground truth [Stanovsky and Dagan 2016; Schneider et al. 2017; Léchelle et al. 2019; Bhardwaj et al. 2019]. The development of such benchmarks is important for OpenIE, because it allows for fast and scalable evaluation of OpenIE systems. Such benchmarks, however, use automatic scorers, which penalize certain properties (e.g. too long extractions) and reward others (e.g. exact match of all words in the golden data). This makes them not suitable for evaluating OpenIE systems in certain scenarios. For instance, the OpenIE benchmark proposed by Stanovsky and Dagan [2016] treats an OpenIE extraction as correctly extracted if the heads of each constituent match the ones of the gold extraction. This is not suitable for this thesis because the benchmark does not account for minimization (which does not change grammatical heads). Moreover, if an OpenIE extraction is simply not found in the golden extractions, the benchmark considers the extraction as *incorrectly extracted*, even if this is not the case. Consequently, this leads to penalizing high-recall OpenIE systems. For these reasons, we are not using the automatic benchmarks in this thesis.

Extrinsic Evaluations

In extrinsic evaluations, OpenIE systems are measured w.r.t. a downstream task. For example, Lin et al. [2020] evaluate OpenIE systems w.r.t. the KB population task. They propose a KB completion system (KBPearl), which uses OpenIE extraction in the pipeline. To evaluate the OpenIE systems w.r.t. the task, the authors switched different OpenIE extractors and report

the performance of KBPearl with each OpenIE extractor individually. Similar evaluation strategies are used for other downstream tasks as well, including event schema induction, text comprehension [Mausam 2016] and fact salience [Ponza et al. 2018; Sheng et al. 2020].

Such evaluations are important for showing the usefulness of OpenIE w.r.t. a certain downstream task. They say nothing, however, about the intrinsic properties of an OpenIE system. In this thesis, we are more focused towards the intrinsic properties of our proposed OpenIE system (MinIE), therefore we do not discuss such extrinsic evaluations in details.

2.2 OpenIE Corpora

OpenIE extractions are useful when they are extracted from large amounts of natural language text data. Such large-scale OpenIE corpora—also known as *open knowledge bases* in some literature [Galárraga et al. 2014; Gupta et al. 2019; Broscheit et al. 2020]—are important for many downstream tasks, such as word embeddings generation [Stanovsky et al. 2015], question answering [Khot et al. 2017] and fact retrieval [Löser et al. 2011]. Because the extracted information is usually represented in the form of triples, such corpora are particularly useful for KB-related tasks, including automated KB construction [Dong et al. 2014], KB extension [Dutta et al. 2015], KB population [Lin et al. 2020] and slot filling [Angeli et al. 2015]. For such tasks, it is important that the OpenIE corpus is large, not too noisy and that the arguments of the triples are correctly disambiguated.

One of the first and widely-used OpenIE resource is the ReVerb corpus [Fader et al. 2011], which consists of high-confidence extractions produced with the ReVerb OpenIE system from the ClueWeb09 corpus [Callan et al. 2009]. In subsequent work, Lin et al. [2012] released a subset of the ReVerb corpus with automatically disambiguated arguments. The PATTY [Nakashole et al. 2012], WiseNet [Moro and Navigli 2012], WiseNet 2.0 [Moro and Navigli 2013], and DefIE [Delli Bovi et al. 2015b] corpora additionally organize open relations in relational synsets and then structure the relational synsets into relational taxonomies. Finally, KB-Unify [Delli Bovi et al. 2015a] integrates multiple different OpenIE corpora into a single resource. Recently, Gashteovski et al. [2019] released OPIEC — an OpenIE corpus constructed from the entire English Wikipedia —, which contains OpenIE triples with disambiguated arguments as well as syntactic and semantic annotations.

Containing more than 341M triples, OPIEC is the largest OpenIE corpus to date. OPIEC contains two subcorpora:

- OPIEC-Clean: contains only triples with arguments that are at least one of the following: 1) recognized named entities; 2) concepts for which at least one Wikipedia page exists; 3) entities (or concepts) that are linked to Wikipedia.

- OPIEC-Linked: contains only triples with arguments that are either entities or concepts that are linked to Wikipedia.

More details about OPIEC and its sub-corpora (OPIEC-Clean and OPIEC-Linked) are discussed in Chapter 4. In Chapter 5 we analyze OPIEC w.r.t. existing knowledge bases that were constructed from the same resource (Wikipedia).

2.3 Knowledge Bases

Contrary to OpenIE corpora, Knowledge Bases (KBs) contain triples with disambiguated elements. In KBs, both the arguments and relations are not surface patterns, but canonicalized entities and relations. Thus, they are are not ambiguous (i.e., a KB triple has precise unambiguous semantics). Consider the following KB triple from the DBpedia [Auer et al. 2007] knowledge base: (Michael Jordan; dbo:birthPlace; Brooklyn). For each argument, we have a direct unique link. This eliminates potential ambiguity, because we know exactly which person and which place the triple is referring to. As for the relation, DBpedia shows a variety of information which makes the meaning of the relation unambiguous. For example, for the relation dbo:birthPlace, we know that the domain (i.e. the subject type) is of type "person", the range (i.e. the object type) is of type "place" and there is also a comment in DBpedia for this relation: *"where the person was born"*. All this information indicates that this relation semantics is about a particular person being born in a particular place.

Such large knowledge bases are usually constructed from semi-structured data, such as Wikipedia infoboxes. For example, DBpedia [Auer et al. 2007] was constructed by extracting information from the infoboxes of the English Wikipedia articles. Suchanek et al. [2007] extracted information from the lexical database WordNet [Miller 1995] and integrated it with the information extracted from Wikipedia infoboxes in a single KB (YAGO). Subsequently, in YAGO2 [Hoffart et al. 2013], adds temporal and spatial information to the triples. Besides the information collected from the English Wikipedia articles, in YAGO3, Mahdisoltani et al. [2013] make use of the infoboxes from Wikipedia articles from other languages. Similarly, BabelNet [Navigli and Ponzetto 2010] is a KB constructed from Wikipedia and WordNet. Such canonical KBs are used in many downstream applications, including recommender systems [Zhang et al. 2016], question answering [Huang et al. 2019], information retrieval [Liu et al. 2018] and computer vision [Fang et al. 2017].

As discussed previously, both the OpenIE extractions and the KB facts are usually represented in the form of *(subject; relation; object)*-triples, which makes them easily comparable resources. Contrary to KBs, OpenIE triples are consisted of surface patterns, which makes them ambiguous. To analyze the information content of compact OpenIE

triples, we analyzed their information content w.r.t. a KB (discussed in Chapter 5). Such analysis is important because it provides insights to the semantics of a large OpenIE corpus and its relationship with a reference KB.

Chapter 3

MinIE: Minimizing Facts in Open Information Extraction

3.1 Introduction

A common problem for OpenIE systems is extracting triples which have overly specific arguments and relations. Consider the sentence *"Pinocchio believes that the hero Superman was not actually born on beautiful Krypton."*, and the corresponding extractions of various systems in Table 3.1, extractions 1–6. Although most of the extractions are correct, they are often overly specific in that their constituents contain specific modifiers or even complete clauses (for more detailed discussion, refer to Section 2.1.2). Such extractions severely limit the usefulness of OpenIE results (e.g., they pose difficulties for linking the entities of the OpenIE triples to KBs for KB population [Lin et al. 2020]). The main goals of OpenIE should be (i) to provide useful, compact extractions and (ii) to produce extractions with high precision and recall. The key challenge in OpenIE is how to achieve both of these goals simultaneously. In fact, most of the available systems (often implicitly) focus on either compactness (e.g. ReVerb [Fader et al. 2011]) or precision/recall (e.g. ClausIE [Del Corro and Gemulla 2013]).

We propose MinIE [Gashteovski et al. 2017], an OpenIE system that aims to address and trade-off both goals. MinIE is built on top of ClausIE, a state-of-the-art OpenIE system that achieves high precision and recall, but often produces overly-specific extractions. To generate more useful and semantically richer extractions, MinIE (i) provides semantic annotations for each extraction, (ii) minimizes overly-specific constituents, and (iii) produces additional extractions that capture implicit relations. Table 3.1 shows the output of (variants of) MinIE

Input sentence

"Pinocchio believes that the hero Superman was not actually born on beautiful Krypton."

OpenIE extractions

OpenIE system	#	Subject	Relation	Object
OLLIE	*1*	*("Pinocchio";*	*"believes that";*	*"the hero [...] beautiful Krypton")*
	2	*("Superman";*	*"was not actually born on";*	*"beautiful Krypton")*
	3	*("Superman";*	*"was not actually born on beau. Krypton in";*	*"the hero")*
ClausIE	*4*	*("Pinocchio";*	*"believes";*	*"that the hero [...] beautiful K.")*
	5	*("the hero Superman";*	*"was not born";*	*"on beautiful Krypton")*
	6	*("the hero Superman";*	*"was not born";*	*"on beautiful Krypton actually")*
Stanford OIE		*No extractions*		
MinIE-C(omplete)	*7*	*("Superman";*	*"was born actually on";*	*"beautiful Krypton")*
		A.: fact. (– [not], CT), attrib. (Pinocchio, +, PS [believes])		
	8	*("Superman";*	*"was born on";*	*"beautiful Krypton")*
		A.: fact. (– [not], CT), attrib. (Pinocchio, +, PS [believes])		
	9	*("Superman";*	*"is";*	*"hero")*
		A.: fact. (+, CT)		
MinIE-S(afe)	*10*	*("Superman";*	*"was born on";*	*"beautiful Krypton")*
		A.: fact. (– [not], CT), attrib. (Pinocchio, +, PS [believes]), relation (was actually born on)		
	11	*("Superman";*	*"is";*	*"hero")*
		A.: fact. (+, CT)		
MinIE-D(ictionary)	*12*	*("Superman";*	*"was born on";*	*"Krypton")*
		A.: fact. (– [not], CT), attrib. (Pinocchio, +, PS [bel.]), rel. (was act. born on), argument (beau. K.)		
MinIE-A(ggressive)	*13*	*("Superman";*	*"is";*	*"hero")*
		A.: fact. (+, CT)		

A annotation; + positive polarity, – negative polarity; *PS* possibility, *CT* certainty; *fact.* factuality; *attrib.* attribution;

Table 3.1 Example extractions and annotations from various OpenIE systems

for the example sentence. Note that MinIE's extractions are significantly more compact but retain correctness.

MinIE's semantic annotations represent information about polarity, modality, attribution, and quantities. The idea of using annotations has already been explored by OLLIE [Mausam et al. 2012] for capturing the context of an extraction. MinIE follows OLLIE, but adds semantic annotations that make the extraction *itself* more compact and useful (as opposed to capturing context). For example, MinIE detects negations in the relation, removes them from the extraction, and adds a "negative polarity" (-) annotation. In fact, MinIE treats surface relations such as *"was born on"* and *"was not born on"* as equivalent up to polarity. The absence of negative evidence is a major concern for relation extraction and knowledge base construction tasks – e.g., addressed by using a local closed world assumption [Dong et al. 2014] or negative sampling [Riedel et al. 2013; Petroni et al. 2015] – and MinIE's annotations can help to alleviate this problem.

In addition to the semantic annotations, MinIE minimizes its extractions by identifying and removing parts that are considered overly specific. In general, such minimization is

inherently limited in scope due to the absence of domain knowledge. Thus MinIE does not and cannot correctly minimize all its extractions in all cases. Instead, MinIE supports multiple minimization modes, which differ in their aggressiveness and effectively control the usefulness-precision trade-off. In particular, MinIE's complete mode (MinIE-C) does not perform any minimizations (except for pruning triples which have whole clauses as an object). MinIE's safe mode (MinIE-S) only performs minimizations that are considered universally safe. MinIE's dictionary mode (MinIE-D) makes use of corpus-level statistics to inform the minimization process. Finally, MinIE's aggressive mode (MinIE-A) only keeps parts that are considered universally necessary. The use of corpus-level statistics by MinIE-D is inspired by the pruning techniques of ReVerb, although we use these statistics for minimization instead of pruning (see Section 3.2). Table 3.1 shows the output of MinIE's various modes. We conducted an experimental study with several real-world datasets and found that the various modes of MinIE produced much shorter extractions than most prior systems, while simultaneously achieving competitive or higher precision (depending on the mode being used). MinIE sometimes fell behind prior systems in terms of the total number of extractions. We found that in almost all of these cases, MinIE became competitive once redundant extractions were removed.

The rest of the chapter is organized as follows: in Section 3.2 we discuss related work; Section 3.3 shows an overview of MinIE; In 3.4 we discuss how MinIE obtains meaningful input extractions for minimization; Section 3.5 explains the semantic annotations provided by MinIE; In Section 3.6 we discuss methods for minimization of OpenIE facts; Section 3.7 discusses an extension of MinIE—MinIE-SpaTe—, which produces further semantic annotations of OpenIE extractions (namely, for space and time); The experimental study is presented in Section 3.8; Next, in Section 3.9 we discuss how MinIE was subsequently used for improving other downstream tasks; Finally, Section 3.10 concludes the chapter.

3.2 Related Work

A general challenge in OpenIE is to avoid both uninformative and overly-specific extractions. ReVerb [Fader et al. 2011] proposed to avoid overly-specific relations by making use of *lexical constraints*: relations that occur infrequently in a large corpus were considered overly-specific and pruned. MinIE's dictionary mode also makes use of the corpus frequency of constituents. In contrast to ReVerb, MinIE uses frequency to inform minimization (instead of pruning) and applies it to relations and arguments as well. Perhaps the closest system in spirit to MinIE is Stanford OIE [Angeli et al. 2015], which uses aggressive minimization. Stanford OIE removes all subconstituents connected by certain typed dependencies (e.g.,

amod).[1] For some dependencies (e.g., *prep* or *dobj*), it uses a frequency constraint along the lines of ReVerb. MinIE differs from Stanford OIE in that it (i) separates out polarity, modality, attribution, and quantities; (ii) uses a different, more principled (and more precise) approach to minimization.

Annotated OpenIE extractions were introduced by OLLIE [Mausam et al. 2012], which uses two types of annotations: *attribution* (the supplier of information) and *clause modifier* (a clause modifying the triple). MinIE extends OLLIE's attribution by additional semantic annotations for polarity, modality, and quantities. Such annotations are not provided by prior OpenIE systems. CSD-IE [Bast and Haussmann 2013] introduced the notion of nested facts (termed "minimal" in their work) and produce extractions with "pointers" to other extractions. NestIE [Bhutani et al. 2016] takes up this idea. OLLIE's clause modifier has a similar purpose. MinIE currently does not handle nested extractions.

Another line of research explores the integration of background knowledge into OpenIE [Nakashole et al. 2012; Moro and Navigli 2012; 2013]. In general, OpenIE systems should use background knowledge when available, but remain open when not. MinIE currently does not use background knowledge, although it allows providing domain-dependent dictionaries.

3.3 Overview

The goal of MinIE is to provide minimized, semantically annotated OpenIE extractions. While the techniques employed here can potentially be integrated into any OpenIE system, we built MinIE on top of ClausIE. We chose ClausIE because (i) it separates the identification of the extractions from the generation of propositions, (ii) it detects clause types, which are also useful for MinIE, and (iii) it is an OpenIE system with high precision and recall.

As ClausIE, MinIE focuses on extractions obtained from individual clauses (with the exception of attributions; see Section 3.5.3). Each clause consists of one subject (S), one verb (V) and alternatively an indirect object (O_i), a direct object (O), a complement (C) and one or more adverbials (A). ClausIE identifies the clause type, which indicates which constituents are obligatory or optional from a syntactic point of view. Quirk et al. [1985] identified seven clause types for English: SV, SVA, SVC, SVO, SVOO, SVOA, and SVOC, where letters refer to obligatory constituents and each clause can be accompanied by additional optional adverbial(s).

[1] For more details of the Stanford typed dependencies, refer to Stanford's typed dependencies manual [De Marneffe and Manning 2008]

The overview of the high-level architecture of MinIE is shown on Figure 3.1. In general, MinIE consists of three major phases:

(1) First, each sentence is preprocessed in order to get the necessary NLP annotations that are used for subsequent processing (e.g., dependency parse tree, POS tags and NER tags). Then, the input sentence along with its NLP annotations is run through ClausIE and a separate extractor for implicit facts (Section 3.4.2). Then, MinIE rewrites ClausIE's extractions to make relations more informative (Section 3.4.1). We refer to the resulting extractions as *input extractions*.

(2) Next, MinIE detects semantic information about polarity (Section 3.5.1), modality (Section 3.5.2), attribution (Section 3.5.3), and quantities (Section 3.5.4). MinIE represents such information with semantic annotations.

(3) To further minimize the resulting *annotated extractions*, MinIE provides various minimization modes (Section 3.6) with increasing levels of aggressiveness: MinIE-C(omplete), MinIE-S(afe), MinIE-D(ictionary), and MinIE-A(ggressive). The modes differ in the amount of minimizations being applied. The result of this phase is a *minimized (compact) extraction*.

Finally, MinIE outputs each minimized extraction along with its annotations. Semantic annotations (such as polarity) are crucial to correctly represent the extraction, whereas other annotations (such as original relation) provide additional information about the minimization process.

3.4 Input Extractions

We first describe how MinIE obtains meaningful input extractions.

3.4.1 Enriching Relations

As mentioned before, MinIE uses ClausIE as its underlying OpenIE system. The relations extracted by ClausIE consist of only verbs and negation particles (cf. Table 3.1). Fader et al. [2011] argue that such approach can lead to uninformative relations. For example, from the sentence *"Faust made a deal with the Devil"*, ClausIE extracts the OpenIE triple *("Faust"; "made"; "a deal with the Devil")*, whereas the extraction *("Faust"; "made a deal with"; "the Devil")* has more informative relation and a shorter argument for the object. Indeed, the relation *"made"* is highly polysemous (the word *make* has 49 synsets in WordNet), whereas

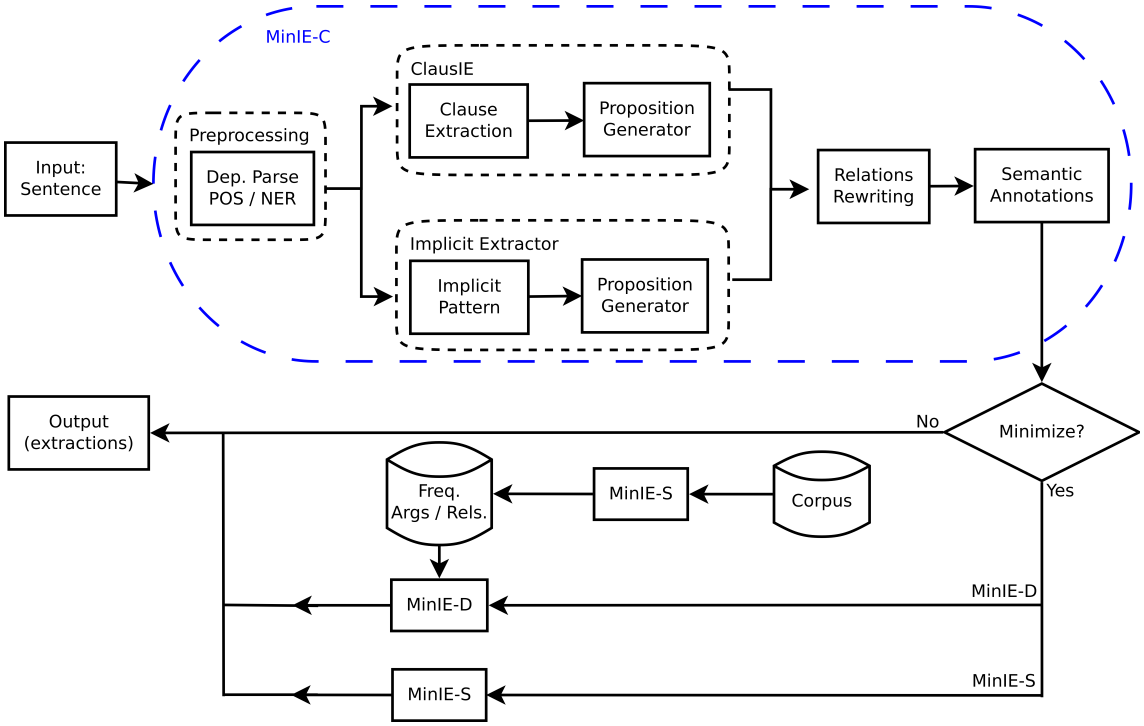


Fig. 3.1 Overview of MinIE

the relation *"made a deal with"* is not. MinIE aims to produce informative relations by deciding which constituents of the input sentence should be pushed from the object into the relation. Our goal is to retain only one of the constituents of the input clause in the argument of the extraction whenever possible, while simultaneously retaining coherence. In particular, our approach uses the clause types detected by ClausIE to ensure that MinIE never removes obligatory constituents from a clause (which would lead to incoherent extractions); it instead may opt to move such constituents to the relation. Our approach is inspired by the syntactic patterns of ReVerb—which is similar to our handling of the SVA and SVO clause types—but, in contrast, applies to all clause types. Note that the relations produced in this step may sometimes be considered overly specific; such relations will be minimized further in subsequent steps.

SVA

For extractions generated by the SVA (Subject-Verb-Adverbial) clause type, if the adverbial is a prepositional complement, we push the preposition into the relation. For example, we rewrite the OpenIE triple *("Superman"; "lives"; "in Metropolis")* to *("Superman"; "lives in"; "Metropolis")*. This allows us to distinguish the relation *"live in"* from other relations such as *"live during"*, *"live until"*, *"live through"*, and so on.

SVO$_i$O, SVOC

In the extractions obtained from the clause types SVO$_i$O (Subject-Verb-Object-Object) and SVOC (Subject-Verb-Object-Complement), we generally push the indirect object (SVO$_i$O) or direct object (SVOC) into the relation. In both cases, the verb requires two additional constituents: we use the first one to enrich the relation and the second one as an argument. For example, we rewrite *("Superman"; "declared"; "the city safe")* to *("Superman"; "declared the city"; "safe")*. As this example indicates, this rewrite is somewhat unsatisfying; further exploration is an interesting direction for future work.

SVOA

For the OpenIE extractions generated by the clause type SVOA (Subject-Verb-Object-Adverbial), if the adverbial consists of a single adverb, we push it to the relation and use the object as an argument. This approach retains coherence because such adverbials are "fluent", i.e., they do not have a fixed position. Otherwise, we proceed as in SVOC, but additionally push the starting preposition (if present) of the adverbial to the relation. For example, *("Ana"; "turned"; "the light off")* becomes *("Ana"; "turned off"; "the light")*, and *("The doorman"; "leads"; "visitors to their destination")* becomes *("The doorman"; "leads visitors to"; "their destination")*.

Optional adverbials

If the clause contains optional adverbials, ClausIE creates one extraction without any optional adverbial and one additional extraction per optional adverbial. The former extractions are processed as above. The latter extractions are treated as if the adverbial were obligatory. For example, the extraction *("Faust"; "made"; "a deal with the Devil")* becomes *("Faust"; "made a deal with"; "the Devil")*. Here the actual clause type is SVO, but we process it as if it were SVOA.

Infinitive forms

If the argument starts with a to-infinitive verb, we move it to the relation. For example, *("Superman"; "needs"; "to defeat Lex Luthor")* becomes *("Superman"; "needs to defeat"; "Lex Luthor")*.

Prepositional phrases with NERs

If the object is a prepositional phrase, such that the prepositional attachment is a named entity and the head noun is not, then the head noun and the preposition are pushed to the relation. Consider the triple *("Satya Nadella"; "is"; "CEO of Microsoft")*, where *"Microsoft"* is named entity of type ORGANIZATION and *"CEO"* is not a named entity. MinIE rewrites this triple to *("Satya Nadella"; "is CEO of"; "Microsoft")*. Such rewrites are helpful for reducing the triple's arguments to recognized named entities, which in turn makes them more useful for downstream tasks such as KB population [Lin et al. 2020].

3.4.2 Implicit Extractions

In some cases, ClausIE produces non-verb-mediated extractions from appositions and possessives. We refer to these extractions as *implicit extractions*. MinIE makes use of additional implicit extractors. In particular, we use the patterns of FINET [Del Corro et al. 2015] to detect explicit type mentions. For example, if the sentence contains "*president Barack Obama*", we obtain *("Barack Obama"; "is"; "president")*. We also include certain patterns involving named entities: pattern *ORG IN LOC* for extraction *("ORG"; "is IN"; "LOC")*; pattern *"Mr." PER* for *("PER"; "is"; "male")* (similarly, *"Ms."* or *"Mrs."*); and pattern *ORG POS? NP PER* for *("PER"; "is NP of"; "ORG")* from RelNoun [Pal and Mausam 2016]. Apart from providing additional high-quality extractions, we use implicit extractions as a signal for minimization (Section 3.6.3). The extractors above have thus been included both to increase recall and to be able to provide more effective minimizations. Table 3.2 lists the implicit extraction patterns in more details along with examples.

3.5 Semantic Annotations

Once input extractions have been created, MinIE detects information about polarity (Section 3.5.1), modality (Section 3.5.2), attribution (Section 3.5.3) and quantities (Section 3.5.4). This information is represented by using semantic annotations. Our focus is on simple, rule-based methods that are both domain-independent and (considered) safe to use in that they do not harm the accuracy of the extraction.

MinIE annotates each extraction with information about its *factuality*. Following Saurí and Pustejovsky [2012], we represent the factuality of an extraction with two pieces of information: polarity ("+" or "-") and modality ("*CT*" or "*PS*"; for "*certainty*" or "*possibility*", respectively). Table 3.3 lists some examples.

#	Pattern	Extraction pattern	Example extraction
1	*The Joie de Vivre store is in Cambridge, Massachusetts.*		
	$[L_1 \mid O] \xrightarrow{appos} L_2$	$([L_1 \mid O],$ *be in,* $L_2)$	*(Cambridge; is in; Massachusetts)*
2	*John Roberts, a Catholic, is Chief Justice of the U.S.*		
	$NP_1 \xrightarrow{appos} NP_2$	$(NP_1;$ *be;* $NP_2)$	*(John Roberts; is; Catholic)*
3	*WPP acquired Les Ouvriers du Paradis in Paris.*		
	O IN L	*(O; be IN; L)*	*(Les Ouvriers du Paradis; be in; Paris)*
4	*Mrs. Vigdís Finnbogadóttir was the world's first democratically directly elected female president.*		
	(Mr. | [Mrs.|Ms.]) P	*(P; be; [male|female])*	*(Vigdís Finnbogadóttir; be; female)*
5	*Google's founders Larry Page and Sergey Brin met at Stanford.*		
	O (POS)? NP P_1	$(P_i;$ *be NP of;* $O);\ i \in \overline{1,n}$	*(Larry Page; be founders of; Google)*
	$([, \mid$ *and* $\mid$ *or*$]\ P_i)_{2 \le i \le n}$		*(Sergey Brin; be founders of; Google)*
6	*Richard Nixon among other presidents, obscured his foreign policy from public view.*		
	P among (other) NP	*(P; be; NP)*	*(Richard Nixon; be; presidents)*
7	*They bar the use of amplifiers on platforms and entering nonpublic areas like tracks and tunnels.*		
	NP_1 *(such as* $\mid$ *like)*	$(NP_i;$ *be;* $NP_1);\ i \in \overline{2,n}$	*(tracks; be; nonpublic areas)*
	NP_2 $([, \mid$ *and* $\mid$ *or*$]\ NP_i)_{2 \le i \le n}$		*(tunnels; be; nonpublic areas)*
8	*Four big tech companies, including Google and Facebook, are open-sourcing deep learning libraries.*		
	NP_1 *(including* $\mid$ *especially)*	$(NP_i;$ *be;* $NP_1);\ i \in \overline{2,n}$	*(Google; be; big tech companies)*
	NP_2 $([, \mid$ *and* $\mid$ *or*$]\ NP_i)_{2 \le i \le n}$		*(Facebook; be; big tech companies)*
9	*Barack Obama, Justin Trudeau and other world leaders celebrate the life of Kofi Annan.*		
	NP_1 $([, \mid$ *and* $\mid$ *or*$]\ NP_i)^*_{1 \le i \le n}$	$(NP_i;$ *be;* $NP_{n+1};\ );\ i \in \overline{1,n}$	*(Barack Obama; be; world leaders)*
	other NP_{n+1}		*(Justin Trudeau; be; world leaders)*
10	*Alcazar is a landmark of the Spanish city of Seville.*		
	JP?[city | town] of L	*(L; be; JP? [city|town])*	*(Seville; be; Spanish city)*
11	*Gov. George E. Pataki announced on Friday that he would create a new redevelopment authority.*		
	NP P	*(P; be; NP)*	*(George E. Pataki; be; Gov.)*

Table 3.2 Implicit extraction patterns (O = organization, L = location, P = person, NP = noun phrase, IN = preposition, JP = adjective phrase, POS = possessive)

3.5.1 Polarity

The *polarity* indicates whether or not a triple occurred in negated form. In order to assign a polarity value to a triple, we aim to detect whether the relation indicates a negative polarity. If so, we assign negative polarity to the whole triple. We detect negations using a small lexicon of negation words (e.g., *no, not, never, none*). If a word from the lexicon is detected, it is dropped from the relation and the triple is annotated with negative polarity (-) and the negation word. In Table 3.3, the extractions from sentences 2 and 4 are annotated as negative. Annotating such extractions with negative polarity is useful for populating KBs with negative

Sentence	Factuality
Superman does live in Metropolis.	(+, CT)
Superman does not live in Metropolis.	(– [not], CT)
Superman does probably live in Metropolis.	(+, PS [probably])
Superman probably does not live in Metropolis.	(– [not], PS [probably])

Table 3.3 Factuality examples. MinIE extracts triple *(Superman; does live in; Metropolis)* from each sentence but the factuality annotations differ.

facts [Arnaout et al. 2020], which is information that most of the current KBs are missing [Auer et al. 2007; Vrandečić and Krötzsch 2014; Pellissier Tanon et al. 2020].

We found that this simple approach successfully spots many negations present in the input relations. Note that whenever a negation is present but not detected, MinIE still produces correct results because such negations are retained in the triple. For example, if a negation occurs in the subject or argument of the extraction, MinIE does not detect it. E.g., from sentence *"No people were hurt in the fire"*, MinIE extracts *("Q_1 people"; "were hurt in"; "fire")* with quantity Q_1=*no* (see Section 3.5.4). This extraction is correct, though it can be further minimized to *("people"; "were hurt in"; "fire")* with a negative polarity annotation. We consider such advanced minimizations too dangerous to use.

MinIE does not deal with double negations, although in many cases it manages to capture the whole concept of the extraction. For instance, from the sentence *"John won't do Jane no good."*, MinIE extracts the following annotated triple:

("John"; "do"; "Jane Q_1 good");

Factuality=(–, PS); $Q_1 = no$.

Generally, negation detection is a hard problem and involves questions such as negation scope resolution, focus detection, and double negation [Blanco and Moldovan 2011]. MinIE does not address these problems, but restricts attention to the simple, safe cases.

3.5.2 Modality

The *modality* indicates whether the triple is a *certainty* (CT) or a *possibility* (PS) according to the clause in which it occurs. We proceed similarly as for the detection of negations and consider a triple certain unless we find evidence of possibility.

To find such evidence, MinIE searches the relation for (1) modal verbs such as *may* or *can*, (2) possibility-indicating words, and (3) certain infinitive verb phrases. For (2) and (3),

Input sentences:			
s_1	*"Donald Trump said that Barack Obama may have been born in Kenya."*		
s_2	*"Donald Trump did not say that Barack Obama may have been born in Kenya."*		
	Extraction	Factuality	Attribution
s_1	(Barack Obama; have been born in; Kenya)	(+, PS)	(Donald Trump, +, CT)
s_2	(Barack Obama; have been born in; Kenya)	(+, PS)	(Donald Trump, –, CT)

Table 3.4 Factuality examples: the factuality of the attribution is independent from the factuality of the extraction

we make use of a small domain-independent lexicon. Our lexicon is based on the lexicon of Saurí and Pustejovsky [2012] and the words in the corresponding WordNet synsets. It mainly contains adverbs such as *probably, possibly, maybe, likely* and infinitive verb phrases such as *is going to*, *is planning to*, or *intends to*. Whenever words indicating possibility are detected, we remove these words from the triple and annotate the triple as possible (PS) along with the words just removed. For example, sentences 3 and 4 in Table 3.3 are annotated PS with the possibility-indicating word *probably*.

3.5.3 Attribution

The *attribution* of a triple is the supplier of information given in the input sentence, if any. We adapt our attribution annotation from the notion of *source* of Saurí and Pustejovsky [2012], i.e., the attribution consists of a supplier of information (as in OLLIE) and an additional factuality (polarity and modality). The factuality of the attribution is independent from the factuality of the extracted triple; it indicates whether the supplier expresses a negation or a possibility. Table 3.4 illustrates an example: the factuality of the subordinate clause *"Barack Obama may have been born in Kenya"* is not dependent from the factuality of the clauses *"Donald Trump said"* and *"Donald Trump did not say"*.

We extract attributions in two ways: from subordinate clauses and from *"according to"* patterns.

Subordinate clauses

MinIE searches for extractions that contain entire clauses as arguments. We then compare the relation against a domain-independent dictionary of relations indicating attributions (e.g., *say* or *believe*).[2] If we find a match, we create an attribution annotation and use the subject

[2] As with modality, the dictionary is based on Saurí and Pustejovsky [2012] plus WordNet synonyms.

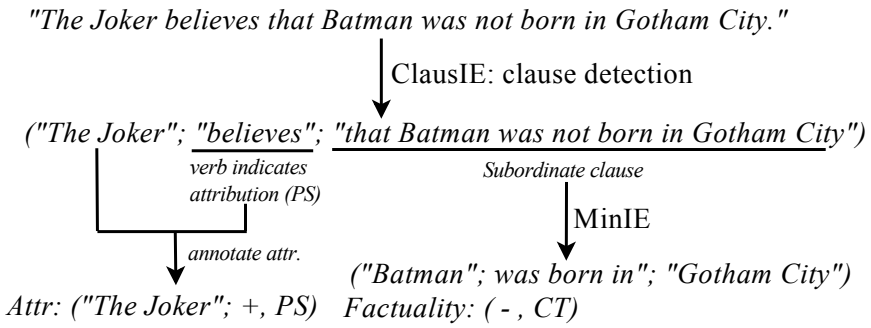


Fig. 3.2 Example of attribution annotation via subordinate clauses

of the extraction as the supplier of information. Each entry in the attribution dictionary is annotated with a modality. For example, relations such as *know*, *say*, or *write* express certainty, whereas relations such as *believe* or *guess* express possibility. If the relation is modified by a negation word, we mark the attribution with negative polarity (e.g., *never said that*). After the attribution has been established, we run ClausIE on the main clause and add the attribution to each extracted triple. Figure 3.2 illustrates such an example.

"according to" adverbial patterns

We search for adverbials that start with *"according to"* and take whatever follows as the supplier with factuality (+, CT). The remaining part of the clause is processed as before. Consider the sentence *"Donald M. Wallace have produced a Lead Detection Kit according to the FDA"*. ClausIE identifies the clause information as SVO(A), where: Subject: *"Donald Wellace"*; Verb: *"have produced"*; Object: *"a Lead Detection Kit"*; Adverbial (optional): *"according to the FDA"*. Then, we feed this information to MinIE, which spots the *"according to"* adverbial and produces the final triple *("Donald Wellace"; "have produced"; "Lead Detection Kit")* with its semantic annotation about the attribution: FDA (example illustrated on Figure 3.3).

3.5.4 Quantities

A *quantity* is a phrase that expresses an amount (or the absence) of something. It either modifies a noun phrase (e.g., *"9 cats"*) or is an independent complement (e.g., *"I have 3"*). Quantities include cardinals (*"9"*), determiners (*"all"*), even whole phrases (*"almost 10"*). If we detect a quantity, we replace it by a placeholder *Q* and add an annotation with the original quantity. The goal of this step is to unify extractions that only differ in quantities. For example, the phrases *"9 cats"*, *"all cats"* and *"almost about 100 cats"* are all rewritten to *"Q cats"*, where only the quantity annotation differs.

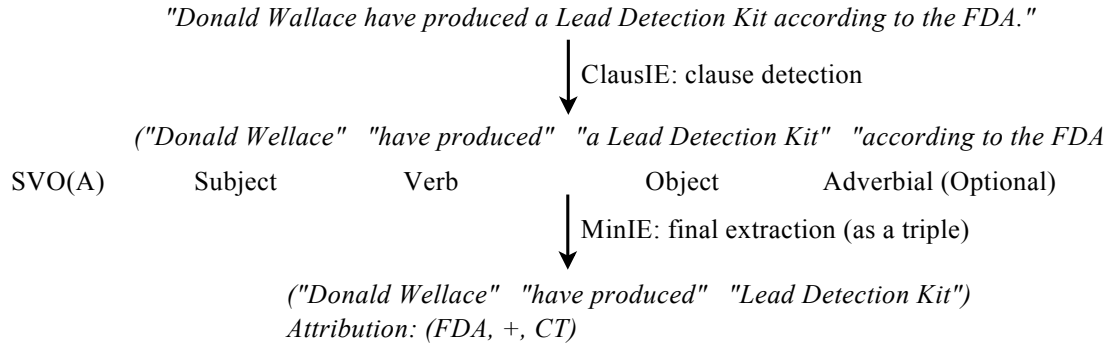


Fig. 3.3 Example of attribution annotation via the *"according to"* pattern

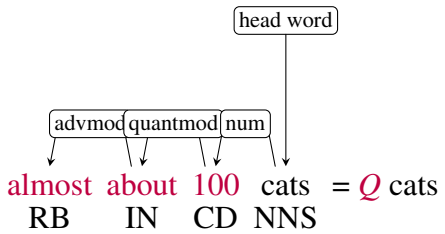


Fig. 3.4 Constituent with a quantity

We detect quantities by first looking for numbers (NER types such as NUMBER or PERCENT) or words expressing quantities (such as *all, some, many*). We then extend such words via relevant typed dependencies, such as quantity modifiers (*quantmod*) and adverbial modifiers (*advmod*). An example can be seen on Figure 3.4.

3.6 Minimization

3.6.1 Overview

After adding semantic annotations, MinIE minimizes extractions by dropping additional words.[3] Since such minimization is risky, MinIE employs various minimization modes with different levels of aggressiveness, which effectively control the minimality-precision trade-off. More precisely, MinIE has four different minimization modes (listed from the least aggressive to the most aggressive level of minimization):

1. **MinIE-C(omplete):** prunes triples which have a whole clause as an object.

[3] MinIE, however, keeps the dropped words from the minimization procedures as annotations. Thus, one can reconstruct the original extraction from these annotations.

Input sentence: *"The big celebration on the campus lasted for 2 days."*				
	Output extractions			
MinIE mode	Subject	Relation	Object	Annotations
MinIE-C	*("The big celebration on the campus";*	*"lasted for"*	*"Q_1 days")*	(+, CT); $Q_1 = 2$
		⇓		
MinIE-S	*("big celebration on campus";*	*"lasted for"*	*"Q_1 days")*	(+, CT); $Q_1 = 2$
		⇓		
MinIE-D	*("celebration on campus";*	*"lasted for"*	*"Q_1 days")*	(+, CT); $Q_1 = 2$
		⇓		
MinIE-A	*("celebration";*	*"lasted for"*	*"days")*	(+, CT)

Table 3.5 An example of MinIE's different modes of minimization. Each minimization mode includes the minimizations of the less aggressive mode(s). The words colored in brown indicate the words which would be dropped in MinIE's next level of aggressiveness.

2. **MinIE-S(afe):** drops words which are considered safe to be dropped. Removing such words (usually) does not damage the semantics of the triple (e.g. determiners modifying nouns).

3. **MinIE-D(ictionary):** drops words which are considered more risky to be dropped (e.g. adjectives modifying nouns). The decision of whether such risky words should be dropped is informed by a dictionary of multi-word expressions.

4. **MinIE-A(ggressive):** drops most of the modifiers of the head words of the constituents in a triple (e.g. adjectives, quantities, adverbs, even whole phrases).

See Table 3.5 for an example sentence and the extractions produced by MinIE's different modes of minimization.

MinIE represents each constituent of an annotated extraction by its words, its dependency structure, its POS tags, and its named entities (detected by a named-entity recognizer). In general, each mode defines a set of *stable subconstituents*, which will always be fully retained, and subsequently searches for candidate words to drop outside of the stable subconstituents. Whenever a word is dropped from a constituent, we add the dropped word as an annotation to the original, unmodified constituent.

In all of MinIE's modes, noun sequences (which include the head) and named entities (from NER) are considered stable subconstituents. MinIE's minimization can be augmented with domain knowledge by providing information about additional stable subconstituents (e.g., collocations and other multi-word expressions).

	Pattern	Original example	Minimized example
Argument	$DT^+ \ [RB\|JJ\|VB]^* \ NN^+$	*"the Byzantine Empire"*	*"Byzantine Empire"*
	$[DT\|RB\|JJ\|VB]^* \ \text{PRP\$} \ [DT\|RB\|JJ\|VB]^* \ NN^+$	*"its officials"*	*"officials"*
	$Phrase \xrightarrow{adj\_mod/adv\_mod} PERSON$	*"particularly Samelsson"*	*"Samelsson"*
	$.^* \ DT^+ \ [RB\|JJ]^* \ NER^+ \ .^*$	*"another $ 6.000"*	*"$ 6.000"*
Relation	$RB^+ \ VB^+$	*"even guessed"*	*"guessed"*
	$(\hat{}VB^+ \ RB^+ \ VB^+) \wedge (head(relation) \neq RB)$	*"has suddenly found"*	*"has found"*
	$(\hat{}VB^+ \ RB^+) \wedge (head(relation) \neq RB)$	*"took just"*	*"took"*

Table 3.6 MinIE-S minimization rules for arguments and relations. The dropped words are written in brown. The minimization rules produced by implicit extractions are omitted in this table. If noun phrases are part of the relation, then the minimization rules for arguments apply to relations as well.

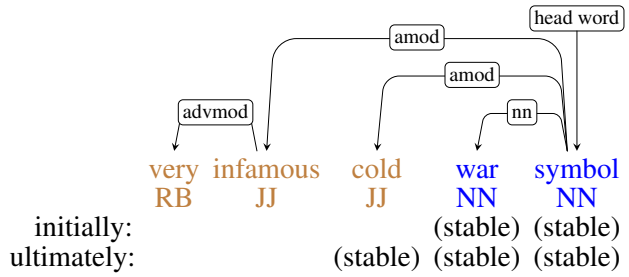
3.6.2 Complete Mode (MinIE-C)

MinIE's *complete mode* (MinIE-C) prunes all the extractions that contain subordinate clause as an object, but does not otherwise modify the annotated extractions. The rationale is that extractions containing subordinate clauses are almost always overly specific. MinIE-C serves as a baseline. One exception of pruning such extractions, however, is the case when MinIE detects an attribution expressed with a subordinate clause (refer to Section 3.5.3 for details on attribution detection from subordinate clauses).

3.6.3 Safe Mode (MinIE-S)

MinIE's *safe mode* (MinIE-S) drops only words which we consider universally safe to drop. We first drop all constituents that are covered by the implicit extractions discussed in Section 3.4.2 (e.g., *"Mr."* before persons). We then drop all determiners, possessive pronouns, adverbs modifying the verb in the relation, as well as adjectives and adverbs modifying words tagged as PERSON by the NER. An exception to these rules is given by named entities, which we consider as stable subconstituents (e.g., we do not drop *"Mr."* in *("Joe"; "cleans with"; "Mr. Muscle")*). Table 3.6 lists the minimization rules along with illustrative examples for MinIE-S.

Note that this procedure cannot be considered safe when used on input extractions. We consider it safe, however, when applied to annotated extractions. In particular, all determiners, pronouns, and adverbs indicating negation, modality, or quantities are already processed and captured in annotations. The safe mode thus only performs simple rewrites such as *"the great city"* to *"great city"*, *"his car"* to *"car"*, *"had also"* to *"had"*, and *"the eloquent president Mr. Barack Obama"* to *"Barack Obama"*.



PSS include: cold war symbol, cold symbol, **cold war**, infamous war symbol, infamous symbol, ...

Fig. 3.5 Illustration of PSS generation in MinIE-D. Initially stable words are marked blue. Entries in dictionary $\mathscr{D}$ are printed in bold face.

Possible errors of the safe minimization mostly come from incorrect prior tagging (e.g. POS tags or NER tags). For instance, from the input sentence *"Personal Ensign won the race"*, MinIE-S extracts the triple *("Ensign"; "won"; "race")*. *"Personal Ensign"* is a name of a particular racing horse, but the prior POS and NER taggers tagged this phrase as follows: *"Personal"*—noun with no NER tag, *"Ensign"*— noun with NER tag "person". Because MinIE-S uses implicit extractions as signal for minimization (Section 3.4.2), the word *"Personal"* is dropped because it is not considered as part of the name of the entity, but merely a descriptive detail (rule 11 in Table 3.2). As a result, we get a triple with changed semantics, because the phrase *"Personal Ensign"* (which is a name of a particular racing horse) has a different meaning than the phrase *"Ensign"* (which can mean either a certain military rank or a flag on a ship).

3.6.4 Dictionary Mode (MinIE-D)

MinIE's dictionary mode (MinIE-D) uses a *multi-word expression dictionary $\mathscr{D}$ of stable constituents*. We first discuss how the dictionary is being used and subsequently how we construct it. An example is given in Figure 3.5.

MinIE-D first performs all the minimizations of the safe mode (MinIE-S), and then searches for maximal noun phrases of the form $P \equiv [adverb|adjective|verb]^+ [noun^+|ner]$. For each instance of P, MinIE-D drops a certain subset of P's words when possible. For example, a suitable minimization for the phrase *"very infamous cold war symbol"* (i.e., the Berlin wall) is *"cold war symbol"*, i.e., we consider *"cold"* as essential to the meaning of the constituent and *"very infamous"* as overly specific. The decision of what is considered essential and what overly specific is informed by dictionary $\mathscr{D}$. Similarly as in [Angeli et al. 2015], we use a dictionary of nonsubsective adjectives [Nayak et al. 2014] which are never allowed to be dropped (e.g. *"artificial hand"* is not a *"hand"*). Note that in order to minimize

Pattern	Examples
$[RB\|JJ]^+\ NN^+$	*"very big city"*
$[RB\|JJ]^+\ NER^+$	*"simply American"*
$(\ VB^+\ [NN^+\|NER^+]) \wedge (g.children(\text{VB}) = \emptyset)$	*"increased GDP"*

Table 3.7 Patterns which determine whether a phrase is elligible for generating PSS; g refers to the dependency-parse tree from the original sentence.

mistakes, we consider for dropping only words in instances of pattern P. In particular, we do not touch subconstituents that contain prepositions because these are notoriously difficult to handle (e.g., we do not want to minimize *"Bill of Rights"* to *"Bill"*).

Our goal is to retain phrases occurring in $\mathscr{D}$, even if they occur in different order or with additional modifiers. We proceed as follows for each instance I of P. We first mark all nouns modifying the root (or the named entity) as *stable*. Afterwards, we create a set of *potentially stable subconstituents* (PSS). Each PSS is queried against dictionary $\mathscr{D}$. If it occurs in $\mathscr{D}$, all of its words are marked as stable. Once all PSS have been processed, we drop all words from I that are not marked stable. In our example, if $\{$*"cold war"*$\} \in \mathscr{D}$, we obtain *"cold war symbol"*.

To generate the set of PSS, we first check if the triple contains an instance I of P (patterns are listed on Table 3.7). If so, we enumerate all syntactically valid subconstituents of I. For example, *"infamous symbol"* or *"cold infamous war"* are syntactically valid, whereas *"very symbol"* or *"very cold war"* are not. Conceptually[4], we enumerate all subsequences of I and check whether (1) at least one noun (or named entity) is retained, and (2) whenever an adverb or adjective is not retained, neither are its modifiers. For each such subsequence, we generate all permutations of adverbial and adjective modifiers originating from the same dependency node, and each result as a PSS. This step ensures that the order of modifiers in I does influence whether or not a word is marked as *stable*. In our example, the set of PSS for *"very infamous cold war symbol"* contains 22 entries:

- combinations from *stable constituents* (1 combination)
  - *"war symbol"*
- combinations from one dependency path (9 combinations)
  - *"*[*very*] *infamous war symbol", "*[*very*] *infamous war", "*[*very*] *infamous symbol"* (6 combinations)

[4]We generate both instances of P as well as the set of PSS directly from the dependency parse structure of the constituent.

  - *"cold war", "cold symbol", "cold war symbol"* (3 combinations). Here, *"cold war"* is found in the dictionary $\mathscr{D}$. Therefore, *"cold"* is marked as *stable* and is not dropped.

- Combinations from several dependency paths (12 combinations)
  - *"[very] infamous cold war", "[very] infamous cold symbol", "[very] infamous cold war symbol"*
  - *"cold [very] infamous symbol", "cold [very] infamous war", "cold [very] infamous war symbol"*

The construction of dictionary $\mathscr{D}$ is inspired by the lexical constraint of Fader et al. [2011]: our assumption is that everything sufficiently frequent in a large corpus is not overly specific. To obtain $\mathscr{D}$, we process the entire corpus using the safe mode and include all frequent (e.g., frequency ≥ 10) subjects, relations and objects into $\mathscr{D}$. Constructing such dictionary could be adapted for a downstream task: applications can extend the dictionary using suitable collocations, either from domain-dependent dictionaries or by using methods to automatically extract collocations from a corpus [Gries 2013; 2015].

3.6.5 Aggressive Mode (MinIE-A)

All previous modes aimed to be more conservative. MinIE-A proceeds in the opposite direction: all words for which we are not sure if they need to be retained are dropped (including the minimization rules for MinIE-S and the minimization rules for MinIE-D assuming an empty dictionary $\mathscr{D}$). For every word in a constituent of an annotated extraction, we drop all adverbial, adjective, possessive, and temporal modifiers (along with their modifiers). We also drop prepositional attachments (e.g., the phrase *"man with apples"* becomes *"man"*), quantities modifying nouns, auxiliary modifiers to the main verb (e.g., *"have escalated"* becomes *"escalated"*), and all compound nouns that have a different named-entity type than their head word (e.g., the noun phrase *"European Union official"* becomes *"official"*). In most cases, after applying these steps, only a single word, named entity, or a sequence of nouns remains for subject and argument constituents. Table 3.8 lists minimization rules of MinIE-A along with examples.

3.7 MinIE-SpaTe: Extension of MinIE

We demonstrated how MinIE minimizes OpenIE extractions into more compact triples by removing unnecessary words (e.g. determiners) without damaging the semantic content of

Pattern	Original example	Minimized example
$Phrase_1 \xrightarrow{adv\_mod} Phrase_2$	*"only vessels"*	*"vessels"*
$Phrase_1 \xrightarrow{adj\_mod} Phrase_2$	*"large machinery"*	*"machinery"*
$Phrase_1 \xrightarrow{predet} Phrase_2$	*"such disposition"*	*"disposition"*
$Phrase_1 \xrightarrow{tmod} Phrase_2$	*"attack last night"*	*"attack"*
$Phrase_1 \xrightarrow{npadvmod} Phrase_2$	*"DePino himself"*	*"DePino"*
$Phrase_1 \xrightarrow{prep} Phrase_2$	*"speculation among gambling analysts"*	*"speculation"*
$Word \xrightarrow{aux} Phrase$	*"car to fix"*	*"car fix"*
Quantity Phrase	*"20,000 fans"*	*"fans"*
NER NP	*"English speakers"*	*"speakers"*
VB_1^+ *TO* VB_2^+	*"reluctance to continue to address needs"*	*"reluctance to address needs"*

Table 3.8 MinIE-A minimization rules for arguments and relations. The dropped words are written in brown. The minimization rules produced by implicit extractions are omitted in this table. The minimization rules for all the other modes of MinIE also apply.

the triple and by providing semantic annotations (for factuality, attribution and quantities). The semantic annotations move auxiliary information from the triple (thereby simplifying it) to annotations. MinIE was designed with such annotations in mind and is flexible to extending their scope. In this section, we discuss an extension of MinIE, which, in addition to the currently available semantic annotations, also includes semantic annotations for space and time. We refer to this extension of MinIE as MinIE-SpaTe.

Space and time is type of information which is used frequently in discourse, because it provides context about where and when events occurred. In fact, large portion of the sentences in large corpora—e.g., Wikipedia or the New York Times corpus [Sandhaus 2008]—contain some sort of temporal or spatial reference. In particular, we conducted a preliminary study to measure the coverage of spatio-temporal information in large corpora. According to our preliminary study, we found that roughly 56% of all the sentences in Wikipedia (and 44% of all the sentences in the New York Times corpus) contain some sort of temporal or spatial information. Detecting time and space in text is an important task, because such semantic information is useful for improving performance on other downstream tasks, including information retrieval [Andogah et al. 2012; Campos et al. 2014], question answering [Jia et al. 2018b] and named entity-disambiguation [Agarwal et al. 2018]. Moreover, current knowledge bases often contain temporal and/or spatial annotation for their facts (e.g., YAGO), thus structuring the triples into SPOTL (Subject Predicate Object Time Location) format [Hoffart et al. 2013]. Finally, attaching temporal information to facts is important for precise temporal slot filling [Wang et al. 2019b; Wang and Jiang 2020], where each fact gets temporal validity (e.g. *("Barack Obama"; "be president of"; "United States") (from=2009, to=2017)*). Since

spatio-temporal information is important, we modified MinIE's output as well by adding additional semantic annotations for space and time, thereby producing OpenIE SPOTL facts.

3.7.1 General Overview

Generally, MinIE-SpaTe makes use of syntactic information provided in the dependency parse as well as information provided by SUTime [Chang and Manning 2012] and the Stanford NER system [Finkel et al. 2005], which is used by default in MinIE.

We subsequently refer to a triple with any spatial or temporal annotation as a *spatial/temporal triple*. MinIE-SpaTe differentiates between three types of such spatial or temporal annotations: (i) annotations on entire triples, (ii) annotations on arguments, and (iii) spatial or temporal references. In what follows, we briefly discuss these types for temporal annotations. Note that similar distinctions apply to spatial annotations as well.

3.7.2 Annotation Format: Time and Space

Temporal annotations on the entire triple

Temporal annotations on triples provide temporal context for the entire triple. For example, from the input sentence *"Bill Gates founded Microsoft in 1975."*, MinIE–SpaTe extracts the triple *("Bill Gates"; "founded"; "Microsoft")* with temporal annotation *("in", "1975")*. Here, *"in"* is a *lexicalized temporal predicate* and *"1975"* is the *core temporal expression*. The core temporal expression is the phrase that carries the main semantic content of the temporal information (in our example, that is the phrase *"1975"*). MinIE-SpaTe complements the core temporal expression with *temporal modifiers* when found. Such phrases indicate modifiers to the core temporal expression, which can be *temporal pre-modifiers* or *temporal post-modifiers*. For example, a clause containing (or being modified by) the phrase *"... at precisely 11:59 PM"* the temporal annotation is *("at", "11:59 PM", premod: "precisely")*, where *"at"* is the lexicalized temporal predicate, *"11:59 PM"* is the core temporal expression and *"precisely"* is the temporal pre-modifier, because it occurs *before* the core temporal expression. On the other hand, the post-modifiers are phrases which modify the core temporal expression and occur *after* the core temporal expression. For example, the phrase *"at the time of writing"* yields the temporal annotation *("at", "time", premod="the", postmod="of writing")*, where *"at"* is the temporal predicate, *"time"* is the core temporal expression (tagged as "past reference"), *"the"* is the temporal premodifier and *"of writing"* is the temporal postmodifier. MinIE-SpaTe uses the TIMEX3 format [Saurı et al. 2006] for representing the temporal information about the triple. TIMEX3 is flexible data format for

representing disambiguated temporal information, which can be adjusted for representing temporal data in different domains (e.g., clinical data [Viani et al. 2019] and legal data [Navas-Loro et al. 2019]) and different languages (e.g., English [Chang and Manning 2012], Korean [Im et al. 2009] and Romanian [Forăscu and Tufiş 2012]).

Temporal annotations on arguments

Sometimes the triple arguments contain temporal information that refers to a phrase, but not to the whole triple. For such cases, MinIE-SpaTe provides temporal annotations for arguments when such temporal information is present within the provenance sentence. For example, from the sentence *"Isabella II opened the 17th-century Parque del Retiro."*, MinIE-SpaTe extracts *("Isabella II"; "opened"; "Parque del Retiro")* with a temporal annotation *("17th-century", "Parque del Retiro")* for the object argument. Generally, the temporal annotation contains information on its target (e.g., object), the temporal expression (*"17th-Century"*) and the head word (*"Retiro"*) being modified by the temporal expression.

Figure 3.6 shows an example which illustrates the difference between a temporal annotation on the entire OpenIE triple and a temporal annotation on an argument. From the sentence: *"Isabella II opened the 17th-century Parque del Retiro in 1868."*, MinIE extracts the following triple: *("Isabella II"; "opened the **17th-century** Parque del Retiro **in**"; "**1868**")*. This triple contains two pieces of temporal information: *"1868"* and *"17th-century"*. MinIE-SpaTe rewrites this as:

> *("Isabella II"; "opened"; "Parque del Retiro")*
> *T: (in, 1868); Object → T: 17th-century*

Distinguishing such type of temporal information is important, because the temporal information should be attached to the adequate information, thus avoiding ambiguity. In this example, we should not attach *"17th century"* as a temporal annotation to the whole triple (because the park was not opened in 17th century by Isabella II), but just to the object (the park *itself* dates from 17th century).

Temporal references

Finally, some triples contain temporal references as subject or object; MinIE-SpaTe annotates such references. For example, from the input sentence *"2003 was a hot year."*, MinIE-SpaTe extracts the triple: *("2003"; "was"; "hot year")*, where the subject (*2003*) is annotated with a temporal reference.

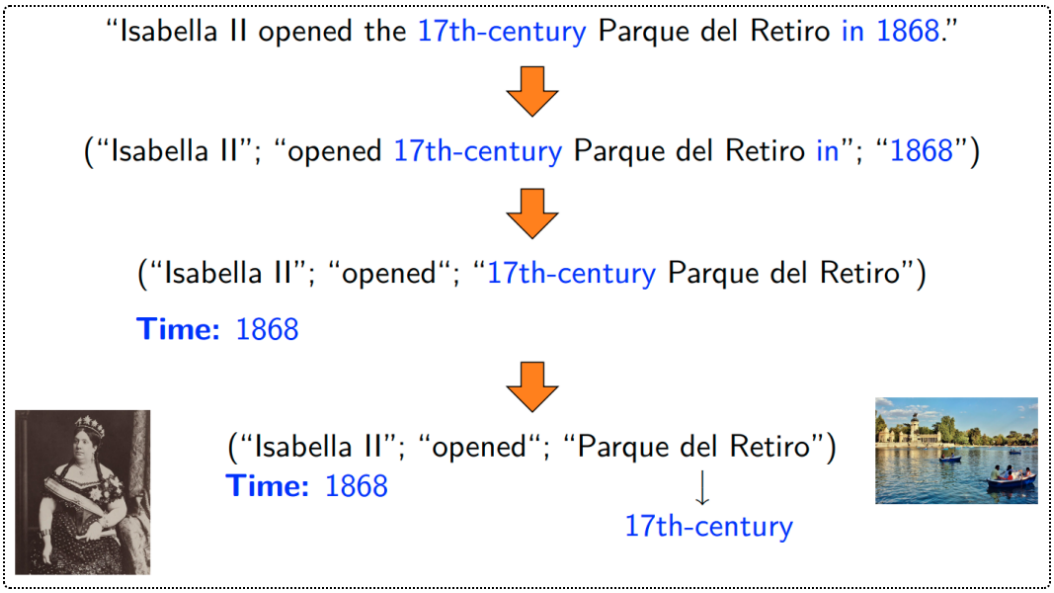


Fig. 3.6 Temporal annotations on an OpenIE triple. The temporal annotation *"1868"* refers to the whole triple and *"17th-century"* refers to the object only.

Spatial annotations

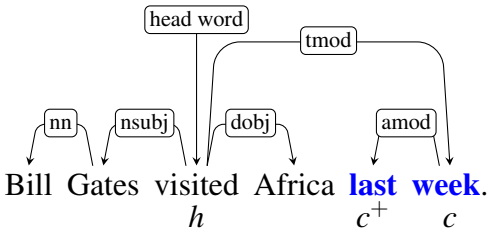
The spatial annotations of MinIE-SpaTe follow the same structure as the temporal annotations. As with the temporal annotations, we have three types of spatial annotations: (i) annotations on entire triples, (ii) annotations on arguments, and (iii) spatial references.

3.7.3 Methodology

Temporal annotations on the entire triple

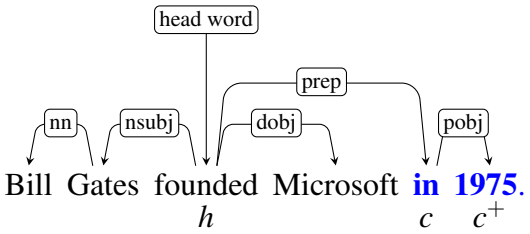
To detect temporal annotations for the entire triple, we use: (i) the dependency parse tree of the input sentence; (ii) the *n*-ary extraction provided by ClausIE; (iii) the triple generated by MinIE; and (iv) a list of temporal expressions from the input sentence, generated in a pre-processing step by the temporal tagger SUTime [Chang and Manning 2012]. As main signals for detecting such temporal annotations, we use certain typed dependencies which modify the head word of the relation. We distinct three groups of typed dependencies for detecting potential temporal annotations on the entire triple: (i) *tmod* or *advmod*; (ii) *prep*; and (iii) *xcomp*.

***tmod* or *advmod*.** We start with the head word of the triple's relation (h) and its descendants in the dependency parse tree (g). If h has children in g with typed dependency *tmod* or *advmod*, then we check if any child node c is contained in the list of temporal expressions. If such child node c exists, we create a temporal annotation for the entire triple with c and



n-ary extraction: ("Bill Gates"; "visited"; "Africa" "last week")
Final extraction: ("Bill Gates"; "visited"; "Africa") **T: (last week)**

Fig. 3.7 Example of temporal annotation on triple with *tmod*



n-ary extraction: ("Bill Gates"; "founded"; "Microsoft **in 1975**")
Final extraction: ("Bill Gates"; "founded"; "Microsoft") **T: (pred=in, t=1975)**

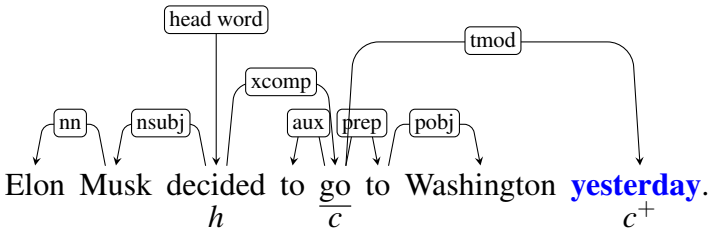
Fig. 3.8 Example of temporal annotation on triple with *prep*

its descendant nodes in the dependency parse tree[5] (c^+). If c and c^+ are a single constituent in the n-ary tuple (such that $n > 3$) extracted by ClausIE, then we remove that constituent by annotating it as a temporal annotation to the entire triple and restructure the n-ary tuple as an OpenIE triple. This OpenIE triple is then passed to MinIE for further processing as explained in Sections 3.5 and 3.5. Figure 3.7 shows an example of such temporal annotation.

prep. If the head word of the relation (h) modifies a child node c with the typed dependency *prep*, then we check if the child node of c is a temporal expression. If so, we temporally annotate the triple with c^+, and c becomes the *lexicalized temporal predicate*. Figure 3.8 shows such example.

xcomp. When the head word of the relation (h) modifies a child node c with typed dependency *xcomp*, we check for temporal annotations the same way as with the typed dependencies *tmod* or *advmod* and *prep*. We treat the direct descendant node of c, however, as though it is the relation head word itself; then we check if the words of c^+ obey the same rules as for *tmod, advmod* or *prep*. Such example is shown on Figure 3.9.

[5]We exclude descendants containing the typed dependencies *rcmod, punct, appos, dep, cc, conj* and *vmod*



n-ary extraction: ("Elon Musk"; "decided"; "to go to Washington **yesterday**")
Final extraction: ("Elon Musk"; "decided to go to"; "Washington") **T: (yesterday)**

Fig. 3.9 Example of temporal annotation on triple with *xcomp*

Temporal Annotations on Arguments

To obtain such annotations, we search for a noun or an adjective in a phrase (word *w*), and check if it is modified by one the following dependencies: *amod*, *acmop*, *advmod*, *nn*, *num*, *number*, *tmod*. If so, we consider the children of *w*. If a child of *w* is a temporal word, we include it in the temporal annotation and drop it from the triple[6]. On the example shown on Figure 3.6, the head word of the object (*"Retiro")* is modified by a temporal word (*"17th-century"*) via the typed dependency *amod*. Subsequently, the temporal word is dropped from the triple and is assigned to the object as a temporal annotation.

Spatial Annotations

For spatial annotations on the entire triple, we follow a similar approach as for the temporal annotations. The main differences are (i) instead of obtaining a list of annotated temporal expressions, we get a list of locations from the Stanford NER system [Finkel et al. 2005]; and (ii) we use the similar rules as for detecting temporal annotations for the type dependency *prep*, but not for the typed dependencies *xcomp*, *tmod*, and *advmod*. The methodology for detecting spatial annotations for arguments and detecting spatial references follows the same methodology as for the temporal annotations.

3.7.4 Confidence Score

In order to estimate whether or not a triple is correctly extracted, we followed ReVerb [Fader et al. 2011] and Wanner [2017] and trained a logistic regression classifier. We used the labeled datasets provided by the experimental study shown on Section 3.8 as training data

[6]We ignore the children derived from the following dependencies: *rcmod*, *punct*, *appos*, *cc*, *conj*

Feature	Value
Length (sentence / extraction)	int / int
Clause type	SVA, SVO, ...
Dropped all optional adverbials	bool
Dropped preposition	bool
Relation appears as a substring in the sentence	bool
Comparison of POS Tags of conjunction words in subject	bool
Contains possessive relation	bool
Contains gerund	bool
Contains infinitive verb in subject / relation	bool / bool
Same order of words in both the extraction and the sentence	bool
Extraction contains typed dependency *dep*	bool
Processed conjunction subject / relation / object	bool / bool / bool
Object appears before the subject in sentence	bool
Extraction occurs in MinIE-D / MinIE-A	bool / bool
Is the relation frequent?[7]	bool
Extracts quantity / time / space	bool / bool / bool

Table 3.9 Features for the confidence score of MinIE-SpaTe

(modified for the spatio-temporal extractions). Features were constructed based on an in-depth error analysis. The most important features were selected using chi-square relevance tests and include features such as the clause type, whether a coordinated conjunction has been processed, and whether or not MinIE minimized the triple. See Table 3.9 for a complete list of features.

3.7.5 Filters

In particular, MinIE retains the NER types provided during preprocessing in its extractions. We aggregated the most frequent relations per argument type and found that many of the triples were of the form *(person; "be"; organization)*, *(location; "be"; organization)*, and so on. These extractions almost always stemmed from an incorrect dependency parses obtained from sentences containing certain conjunctions. Therefore, MinIE-SpaTe filters all triples with lemmatized relation *"be"* and different NER types for subject and object from the output extraction of MinIE-SpaTe (e.g. *("Bill Gates"; "be"; "Microsoft")* is clearly an incorrectly extracted triple).

3.7.6 Precision of Spatio-Temporal Annotations

We performed an experimental study to assess the precision of the spatio-temporal annotations. We used the NYT-10k dataset and ran MinIE-SpaTe on these sentences. Next, we created four

[7] minimum support: 100 K

sets of extractions: triples containing temporal annotations on (1a) the triple level and (1b) the argument level, and triples containing spatial annotations on (2a) the triple level and (2b) the argument level. From each subset, we selected 200 random triples and labeled whether the corresponding MinIE extraction (which does not provide spatio-temporal annotations) was correctly extracted. We construct our final evaluations sets by including 100 random *correctly extracted* triples. In each subset, a human labeler then assessed whether the spatial and/or temporal annotations provided by MinIE-SpaTe were correct as well.

For the triple-level temporal annotations, 91/100 extractions were labeled as correctly annotated; for the argument-level temporal annotations, 80/100 were correctly annotated. Similar precision was measured for the spatial annotations: 91/100 on the triple-level and 82/100 on the argument level. We performed an error analysis and found that common errors stem from either incorrect spatio-temporal tags (e.g., the argument *"Summer Olympics"* gets reduced to *"Olympics"* with temporal annotation *"Summer"*) or errors in the dependency parse tree of the provenance sentence.

3.8 Experimental Study

The goal of our experimental study is to investigate the differences in the various modes of MinIE w.r.t. precision, recall, and extraction length as well as to compare it with popular prior methods. For precision and recall, we followed standard practice and used a random sample of 200 sentences from two datasets each and we ran OpenIE systems on them. Then, human labelers manually assessed whether each extraction was correctly extracted according to the annotation guides (Appendix A). The resulting numbers are used for measuring precision and recall. To measure compactness of the extractions, we report statistics of extraction length (i.e. the number of words per extraction) for each OpenIE system. With these experimental methods, we report precision and recall as well as compactness of MinIE w.r.t. prior OpenIE systems.

We found that MinIE-C, MinIE-S and MinIE-D have high precision and recall. The high precision and recall of MinIE-C—the baseline MinIE system—is mostly attributed to: 1) ClausIE, which is MinIE's underlying OpenIE system; 2) MinIE's implicit extractions; 3) the removal of complex extractions which have whole clauses as arguments, which often introduce incorrectly extracted triples (details about MinIE-C are discussed in Section 3.6.2). As for the minimization process, we found that as the minimization becomes more aggressive, the precision scores suffer, though the extractions become more compact. In particular, the minimization strategies of MinIE-S and MinIE-D significantly improve compactness of the extractions without significant loss of precision points. Surprisingly, even though MinIE-A

suffers significant loss of precision at the expense of compactness w.r.t. the less aggressive modes of MinIE, it still has comparable precision w.r.t. prior OpenIE systems.

3.8.1 Experimental Setup

Datasets

For our experiments, we used three different datasets, which consist of randomly sampled sentences:

- *NYT-10k:* 10,000 randomly sampled sentences from the New York Times Corpus [Sandhaus 2008]
- *NYT-200:* a random sample of 200 sentences from the same corpus
- *Wiki-200:* a random sample of 200 sentences from Wikipedia

The datasets NYT-200 and Wiki-200 are well established datasets for manual evaluation of OpenIE systems and were used in the evaluation of ClausIE and NestIE[8].

Methods

We used ClausIE, OLLIE, and Stanford OpenIE as baseline systems. We adapted the publicly available version of ClausIE to Stanford CoreNLP 3.8.0 [Manning et al. 2014] and implemented MinIE on top. For MinIE-D, we built dictionary $\mathscr{D}$ from the entire New York Times and Wikipedia corpus, respectively.

Labeling

We ran each OpenIE system[9] on each sentence of NYT-200 and Wiki-200. We used these two datasets because of two main reasons. First, these two datasets are well-established for manual evaluation of prior OpenIE systems [Mausam et al. 2012; Del Corro and Gemulla 2013; Bhutani et al. 2016]. Second, labeling OpenIE extractions is very time-consuming and non-trivial. Therefore, due to practical reasons, we cannot use the larger dataset NYT-10k. Overall, there were more than 9,400 distinct extractions from NYT-200 and Wiki-200 produced by all OpenIE systems in our study, which were manually labeled.

[8] We did not use the OpenIE benchmark of Stanovsky and Dagan [2016] because it treats an extraction as correctly extracted if the heads of each constituent match the ones of a gold extraction. This is not suitable for us because it does not account for minimization (which does not change grammatical heads).

[9] The baseline OpenIE systems (ClausIE, OLLIE and Stanford OIE) + MinIE in all its modes.

Then, two labelers independently provided two labels per extraction: one label for the extraction correctness of the triple (without attribution) and one label for the extraction correctness of the attribution alone. Each triple is labeled as *correctly extracted* if it is entailed by its corresponding clause; here factuality annotations are taken into account but attribution errors are ignored. For example, all triples except #3 of Table 3.1 are considered correctly extracted. An attribution is incorrectly extracted if there is an attribution in the sentence which is neither present in the triple nor in the attribution annotation. In Table 3.1, the attribution is considered to be incorrectly extracted for the extractions #2, #3, #5, and #6. Attribution is labeled only when the fact triple is labeled as correctly extracted. For more details about the labeling guidelines, see Appendix A.

Each extraction was labeled by two independent labelers. We treat an extraction as *correctly extracted* if both labelers labeled it as *correctly extracted*. The inter-annotator agreement was moderate (NYT-200: Cohen's $\kappa = 0.53$, 78% of labels agree; Wiki-200: $\kappa = 0.5$, 79% of labels agree). The reason for the moderate inter-annotator agreement is because in some cases it is hard to assess whether the triple is correctly extracted or not, therefore leading to different conclusions from the labelers. Consider the following input sentence and its OpenIE extraction:

> *"Harry Eagle won the Waterford International Biomedical Award and Eli Lilly Award in Bacteriology."*
>
> *("H. Eagle"; "won Waterford International Biomedical Award in"; "Bacteriology")*

In this particular example, one can label this extraction as *correctly extracted* or as *incorrectly extracted*, and this depends on the underlying assumptions of the labeler. In particular, if we assume that the full name of the second award is *"Eli Lilly Award in Bacteriology"*, then we cannot know for sure if the Waterford International Biomedical Award is in fact about bacteriology at all, thus leading us to a conclusion that this is an incorrectly extracted triple, because we need more information to know for sure. On the other hand, if we assume that both awards are about bacteriology—which is possible judging by the sentence alone—, then the OpenIE triple is indeed correctly extracted. In our example, one labeler labeled the triple as *incorrectly extracted* and the other labeler as *correctly extracted*. For our study, we consider the overall label of this example as *incorrectly extracted*, because only one of the labelers considered the triple as *correctly extracted*.

Measures

For each system, we measured the total number of extractions, the total number of correctly extracted triples (*recall*), the fraction of correctly extracted triples out of all extractions

	OLLIE	ClausIE	Stanford	MinIE-C	MinIE-S	MinIE-D	MinIE-A
# non-redundant extr.	20,557	36,173	16,350	**37,465**	37,093	36,921	36,474
# with redundant extr.	24,316	58,420	43,360	47,637	45,492	45,318	42,842
$\mu \pm \sigma$	9.9 ± 5.8	10.9 ± 7.0	6.6 ± 3.0	8.3 ± 4.9	7.2 ± 4.2	7.0 ± 4.1	$\mathbf{4.7 \pm 1.9}$
with attributions	6.8%	-	-	10.8%	10.8%	10.7%	10.8%
with negative polarity	-	-	-	3.8%	3.7%	3.7%	3.8%
with possibility	-	-	-	10.1%	9.9%	10.0%	9.7%
with quantity	-	-	-	17.6%	17.8%	17.8%	1.9%

Table 3.10 Results on the unlabeled NYT-10k dataset (μ=avg. extraction length, σ=standard deviation)

(*factual precision*), and the fraction of correctly extracted triples that have correctly extracted attributions (*attribution precision*). We also determined the mean word count per triple (μ) and its standard deviation (σ) as a proxy for minimality. Finally, as some systems produced a large number of redundant extractions, we also report the number of non-redundant extractions. For simplicity, we consider a triple t_1 redundant if it appears as subsequence in some other triple t_2 produced by the same extractor from the same sentence (e.g., extraction #5 in Table 3.1 is redundant given extraction #6).

3.8.2 Extraction Statistics

In our first experiment, we used the larger but unlabeled NYT-10k dataset. The goal of this experiment was to investigate the total number of redundant and non-redundant extractions produced by each OpenIE system and how frequently semantic annotations were produced (Table 3.10). For MinIE's semantic annotations, we only show the fraction of negative polarity and possibility annotations for triples (i.e., we exclude the factuality annotations of the attribution).

In terms of number of extractions, MinIE (all modes) and Stanford OIE are roughly on par. On the other hand, OLLIE has significantly fewer and ClausIE significantly more extractions compared to MinIE. The reason why ClausIE has more extractions than MinIE is that different (partly redundant) extractions from ClausIE may lead to the same minimized extractions in MinIE. This is also the reason why the number of extractions drops in the more aggressive modes of MinIE. We also determined the number of non-redundant extractions produced by each system and found that most systems produced only a moderate number of redundant extractions. A notable exception is Stanford OIE, which produced many extraction variants by dropping different subsets of words.

We observed that all modes of MinIE achieved significantly shorter extractions than ClausIE (MinIE's underlying OpenIE system), and that the average extraction length indeed

	OLLIE	ClausIE	Stanford	MinIE-C	MinIE-S	MinIE-D	MinIE-A
			NYT				
# non-redundant (correct/total)	246/414	505/821	178/342	**581/785**	574/781	569/777	439/753
# w/ redundant (correct/total)	302/497	**792/1300**	530/1052	727/970	690/924	681/916	505/860
factual prec.	(0.61)	(0.61)	(0.5)	**(0.75)**	**(0.75)**	(0.74)	(0.59)
attr. prec.	(0.9)	-	-	**(0.94)**	(0.93)	(0.93)	(0.93)
			Wiki				
# non-redundant (correct/total)	229/479	424/704	217/398	**500/666**	489/661	486/669	401/658
# w/ redundant (correct/total)	284/565	628/1002	**651/1519**	635/851	602/816	593/816	474/783
factual prec.	(0.50)	(0.63)	(0.43)	**(0.75)**	(0.74)	(0.73)	(0.61)
attr. prec.	**(0.97)**	-	-	**(0.97)**	(0.96)	(0.96)	**(0.97)**

Table 3.11 Results on the labeled NYT-200 and Wiki-200 datasets

dropped as we used more aggressive modes. Out of all minimization modes of MinIE, only MinIE-A produced shorter extractions than Stanford OIE. The main reason for the short extraction length of Stanford OIE is its aggressive creation of short redundant extractions (at the cost of precision; see section 3.8.3). We also found that to further minimize the extractions of both MinIE-S and MinIE-D, it is often necessary to minimize subjects and objects with prepositional modifiers (which MinIE-S and MinIE-D currently avoid).

Only OLLIE and MinIE make use of semantic annotations. The fraction of extracted attribution annotations was significantly smaller for OLLIE than for MinIE, mainly because OLLIE's attribution detection is limited only to the *ccomp* dependency relation. Our results also indicate that MinIE frequently provides semantic annotations (with the notable exception of negative polarity). We found, however, that when the input sentences are from different domains, the coverage of semantic annotations of the OpenIE extractions could also be different. We ran (a variant of) MinIE-S on the entire English Wikipedia (Chapter 4) and found that sometimes the coverage of a certain semantic annotation for those triples (Table 4.3) is different than the coverage of semantic annotations for the triples extracted from NYT-10k (Table 3.10). For example, there is significantly higher coverage of attributions detected in the triples extracted from NYT-10k compared to the triples extracted from Wikipedia. The reason for such difference could be because NYT is newswire corpus, where large portion of the sentences are about reporting who said what, while Wikipedia contains encyclopedic knowledge, which rarely contains such sentences.

3.8.3 Precision

In our second experiment, we compared the precision and recall of the various OpenIE systems on the smaller NYT-200 and Wiki-200 datasets. Our results are summarized in Table 3.11.

We found that Stanford OIE had the lowest factual precision and recall for non-redundant extractions throughout; it produced many incorrect and many redundant extractions (e.g., Stanford OIE produced 400 extractions from five sentences on NYT-200). For MinIE, the factual precision dropped as expected when we use more aggressive modes. Interestingly, the drop in precision between MinIE-C and MinIE-D was quite low, even though the extractions of MinIE-D get shorter. The aggressive minimization of MinIE-A led to a more severe drop in precision. Surprisingly to us, even MinIE's aggressive mode achieved precision comparable to ClausIE and higher than Stanford OIE. We found that MinIE-C, MinIE-S, and MinIE-D had higher precision than ClausIE. Reasons include that MinIE produces additional high-precision implicit extractions and breaks up very long and thus error-prone extractions. We also tried enriching the dictionary of MinIE-D with WordNet and Wiktionary collocations. We found that the precision of MinIE-D was almost the same.

As for attribution precision, most of the sentences in our samples did not contain attributions; these numbers thus have low accuracy. OLLIE and MinIE achieved similar results, even though MinIE additionally annotated attributions with factuality information.

3.8.4 Discussion

For all modes, errors in dependency parsing transfer over to errors in MinIE, which we believe was the main source of incorrectly extracted triples in MinIE-C and MinIE-S. One particular difficulty for the dependency parser is processing the conjunctions Chen and Manning [2014]. Subsequently, Saha and Mausam [2018] also observed this issue and proposed techniques for correcting such errors in OpenIE that originate from errors in dependency parses of conjunctions. In particular, they proposed methods for breaking down a conjunctive sentence into several simpler sentences that are without conjunctions, which are then fed to the OpenIE system. This strategy decreases the probability of producing errors by the dependency parser, because the sentences are both simpler and without conjunctions. As a consequence, this leads to decreasing the probability of errors produced by the OpenIE extractor. We believe that implementing such techniques on MinIE as a preprocessing step might result in significant improvement of the extractions.

For MinIE-D, a frequent source of error is that MinIE-D sometimes drops adjectives which in fact form multi-word expressions with the noun they are modifying (e.g., "*assistant*

director"). This happens when the multi-word expression is not present in the dictionary. For improving the performance of MinIE-D due to such issues, the use of better multi-word expression dictionaries will help address this problem. This could be achieved by enriching the dictionaries either with methods for detecting collocations from large corpora [Espinosa-Anke et al. 2016] or adding multi-word expressions from other already existing resources.

Another source of error stems from the underlying NER system used by MinIE. For example, the first word of the entity *"Personal Ensign"* (which is a name of a horse) was not recognized as a named entity, while the second was recognized as PERSON, thus leading to the incorrect extraction: *("Personal"; "is"; "Ensign")*.

3.9 Use of MinIE for Downstream Tasks

We believe that the use of minimized extractions with semantic annotations are a promising direction for OpenIE. The techniques presented in this chapter can be seen as a step towards this goal. One important direction could be adding more types of semantic annotation. For this reason, MinIE is designed to be flexible towards adding new semantic annotations. For example, we showed in Section 3.7 such extension of MinIE (namely, MinIE-SpaTe), where we added spatial and temporal annotations to the extractions produced by MinIE. MinIE-SpaTe was used to produce the largest OpenIE corpus to date which was constructed from the entire English Wikipedia (details in Chapter 4). Moreover, subsequent work [Lauscher et al. 2019] proposed yet another extension of MinIE (called MinScIE), which is specifically tuned for extracting information from scientific literature. The authors enriched MinScIE with additional semantic annotations specialized for paper citations (in particular, semantic annotations for citation marker, citation polarity and citation function).

The compactness of the extractions provided by MinIE is helpful for KB-related downstream tasks. For example, Lin et al. [2020] proposed an end-to-end system for KB population (called KBPearl). To leverage the information found in natural language text for populating a KB, KBPearl uses an underlying OpenIE system in the early stages of its pipeline. The authors tested KBPearl with several OpenIE systems and, through an empirical experimental study, showed that the compactness of the OpenIE extractions produced by MinIE makes KBPearl perform best among several OpenIE systems for KB population. The study shows the potential of using OpenIE for KB-related downstream tasks.

MinIE was also used in subsequent work for improving another downstream task: fact salience. Ponza et al. [2018] proposed SalIE, which is a system for detecting salient OpenIE facts from a corpus. The authors showed that with the use of OpenIE facts from MinIE, their

system SalIE competes with state of the art text summarization systems, outperforming them on benchmark datasets w.r.t. the ROUGE metric.

3.10 Conclusions and Future Work

In this chapter, we discussed methodologies for minimizing facts in open information extraction. One direction for effective minimization is through structuring some aspects of the semantics carried in the triple by moving its words to semantic annotations (e.g. factuality, space, time, etc.). Other direction for minimization is to remove words in the triple that are considered to be overly-specific. In principle, the minimized triple should still carry the same information as before minimization. Such minimization methods produce more compact extractions, which can be subsequently used for other downstream tasks, such as KB population [Lin et al. 2020] and fact salience [Ponza et al. 2018]. In this chapter, we proposed the OpenIE system MinIE, which performs such minimizations. We examined different minimization strategies, which differ in their level of aggressiveness, thus resulting in several minimization modes—MinIE-C(omplete), MinIE-S(afe), MinIE-D(icitionary) and MinIE-A(ggressive)—as well as an extension of MinIE that performs semantic annotations for space and time (MinIE-SpaTe).

With our experimental study (Section 3.8), we found that MinIE in its complete and safe mode outperforms other prior methods in terms of precision and recall. As expected, we found that as MinIE progresses from less aggressive to more aggressive modes, while the extractions indeed get shorter, the precision points get lower. What we found surprising, however, was the fact that even when using MinIE-D—which includes riskier minimization strategies like dropping adjectives that modify nouns—, the precision score did not suffer significantly. One reason for such observation is the method for constructing dictionary of multi-word expressions. Because we used a dictionary of frequent relations and arguments which were extracted by the use of MinIE's safe mode on large corpora, it became less likely for MinIE-D to drop words which are important for the meaning of the argument and relation. One way to fix the problem of the slight drop of precision points is to investigate better ways of constructing dictionaries of multi-word expressions, which can be fed into MinIE-D. Other surprising observation was that even in its most aggressive mode, MinIE is still competitive with other prior methods. In this case, however, the drop in precision is significant w.r.t. to the previous level of aggressiveness (MinIE-D). We believe that a promising direction for future research work is finding better trade-off between the shortness of extractions of MinIE-A and the precision and recall points of MinIE-S.

Since the publication of MinIE, the research community focused on training neural models for learning OpenIE extractors. The advantage of such neural models is that they do not rely on prior complex linguistic processing (e.g., dependency parsing). Consequently, the errors produced by such linguistic processing would not propagate throughout the OpenIE pipeline as is the case with MinIE and other OpenIE systems that rely on prior NLP computations, such as dependency parsing or part-of-speech tagging. Instead, such systems use spans as features and treat the task as either sequence tagging problem [Stanovsky et al. 2018; Roy et al. 2019; Zhan and Zhao 2020], or as sequence generation problem [Cui et al. 2018; Sun et al. 2018b;a; Kolluru et al. 2020].

One common problem for designing such neural OpenIE systems is the lack of training data. Stanovsky et al. [2018] use the benchmark OpenIE data they previously published [Stanovsky and Dagan 2016]. This dataset was originally constructed for another task (QA-SRL), which was then automatically transformed into OpenIE dataset. Therefore, the dataset is biased towards QA-SRL scenarios and it also contains considerable amount of noise due to the automatic transformation from QA-SRL to OpenIE [Bhardwaj et al. 2019]. Other line of work generates training data by simply running a prior system on large corpus (e.g., Wikipedia) and selecting only the triples with high confidence score (e.g. > 0.9) as positive training data [Cui et al. 2018; Zhan and Zhao 2020]. Similarly, other work uses the output triples of several OpenIE systems (instead of just one), which are subsequently filtered in order to create high-quality training data [Roy et al. 2019; Kolluru et al. 2020]. While such bootstrapped datasets provide larger quantities of data, they still suffer from some drawbacks. One drawback is that many of the prior OpenIE systems do not produce all the possible extractions that can be made, which means that the training data is limited only to extractions produced by prior systems. Moreover, they still contain considerable amount of noise, because the automatic extractors and filters are not perfect. One way to tackle such issues is with manual evaluation of the extractors and filters or by manually generating training data. For these reasons, another direction is to use crowdsourced dataset [Sun et al. 2018b;a]. In principle, such crowdsourced datasets have higher quality than automatically bootstrapped data, though, due to the complexity of the task, they are hard to scale.

Overall, neural models for OpenIE are a promising direction of research. As discussed previously, the major problem for training such models is the lack of training data that is both high-quality and high-quantity. We believe that investigating better methods for generating training data that is both high-quality and high-quantity is promising research direction. As for the compactness of OpenIE extractions, we believe that one interesting future direction of research is to investigate methods for training a model that effectively minimizes OpenIE extractions. Ideally, such minimization model can be used for minimizing

extractions on any already existing OpenIE system. The minimization model would tune the precision-compactness trade-off which we observed between MinIE-S and MinIE-A. Again, training data would be a major bottleneck for such research. One possible way to overcome the problem of lack of training data is to manually annotate a smaller seed of extractions that would contain information about which words are (and which words are not) safe to drop without damaging the semantics of the extractions. Then, through the use of weakly-supervised strategies, MinIE could be used to generate more high-confidence data for minimization.

In the following chapter, we present a large OpenIE corpus (called OPIEC) which was produced by running MinIE-SpaTe on the entire English Wikipedia. In Chapter 5 we study the alignments of OPIEC with KBs constructed from the same resource (Wikipedia), namely DBpedia and YAGO. MinIE-SpaTe is suitable OpenIE system for performing such comparisons, because (i) it provides compact extractions; and (ii) it can provide insights of spatio-temporal extractions compared with YAGO which also contains spatio-temporal KB triples.

Chapter 4

OPIEC: An Open Information Extraction Corpus

OpenIE extractions are most efficiently used in downstream tasks when they are extracted from large corpora. For example, OpenIE extractions from large corpus can be used for populating a knowledge base (KB) [Lin et al. 2020], because the information contained in OpenIE extractions often complements the information in already existing structured KBs. Moreover, when large amounts of text data is structured with OpenIE triples, we can understand the underlying information within the text easier through other downstream tasks, such as fact retrieval [Löser et al. 2011], offering explanation of entity-ranking results in information retrieval through OpenIE extractions [Kadry and Dietz 2017], summarizing text through selecting OpenIE salient facts [Ponza et al. 2018], or comparing the OpenIE corpus against other similar resources (e.g. semantically unambiguous KBs; for more details, refer to Chapter 5).

In this chapter, we describe OPIEC—the largest OpenIE corpus to date—which was extracted from the entire English Wikipedia. OPIEC retains the original (golden) links found in Wikipedia articles and was extracted with MinIE–SpaTe (described in Chapter 3.7), thus providing the necessary compactness (discussed in Chapter 3) for studying the alignments between OpenIE triples and canonicalized KBs constructed from the same resources (discussed in Chapter 5).

4.1 Introduction

The extractions of OpenIE systems from large corpora are a valuable resource for downstream tasks [Etzioni et al. 2008; Mausam 2016], such as automated knowledge base con-

struction [Riedel et al. 2013; Wu et al. 2018; Vashishth et al. 2018; Shi and Weninger 2018], knowledge base population [Lin et al. 2020], open question answering [Fader et al. 2013; 2014; Khot et al. 2017; Yan et al. 2018; Saha Roy and Anand 2020], event schema induction [Balasubramanian et al. 2013], generating inference rules [Jain and Mausam 2016], or for improving OpenIE systems themselves [Mausam et al. 2012; Yahya et al. 2014]. Besides the large corpora produced by OpenIE systems (also known as *Open Knowledge Bases* [Galárraga et al. 2014] or *Open Knowledge Graphs* [Gupta et al. 2019] in different literature), a number of derived resources have been produced from OpenIE extractions, including entailment rules [Jain and Mausam 2016], question paraphrases [Fader et al. 2013], Rel-grams [Balasubramanian et al. 2012], and OpenIE-based embeddings [Stanovsky et al. 2015].

The role of OpenIE corpora in downstream tasks is in that they structure the information of the text data into more machine-readable format (e.g. triples). For example, there are knowledge bases which are constructed from semi-structured data (e.g., DBpedia extracts information from Wikipedia infoboxes), which do not extract information from natural language text. By structuring large amounts of textual data into triples, OpenIE makes the information found in text much more accessible, which is subsequently used for tasks such as populating KBs [Lin et al. 2020] or link prediction between entities in the KB [Gupta et al. 2019].

In this chapter, we discuss our OpenIE corpus called *OPIEC*[1] [Gashteovski et al. 2019]. The OPIEC corpus was extracted from the full text of the entire English Wikipedia, using the Stanford CoreNLP pipeline [Manning et al. 2014] and the OpenIE system MinIE-SpaTe (discussed in Chapter 3.7). With more than 341M OpenIE triples, OPIEC is the largest publicly available OpenIE corpus to date and, for each of its extractions, it contains valuable metadata information which is not available in existing resources (see Table 4.1 for an overview, and section 4.2 for a detailed discussion on related OpenIE corpora). In particular, for each triple, OPIEC provides detailed provenance information (e.g. Wikipedia article ID where the triple was extracted from), syntactic annotations (such as POS tags, lemmas, dependency parses), semantic annotations (such as polarity, modality, attribution, space, time), entity annotations (NER types and, when available, Wikipedia links), as well as confidence scores. Figure 4.1 shows an example of an OpenIE triple from OPIEC along with its annotations.

OPIEC complements available OpenIE resources [Fader et al. 2011; Lin et al. 2012; Nakashole et al. 2012; Moro and Navigli 2012; 2013; Delli Bovi et al. 2015b;a]. For example, WiSeNet [Moro and Navigli 2013] was also constructed from the English Wikipedia and it

[1]The OPIEC corpus is available at https://www.uni-mannheim.de/dws/research/resources/opiec/

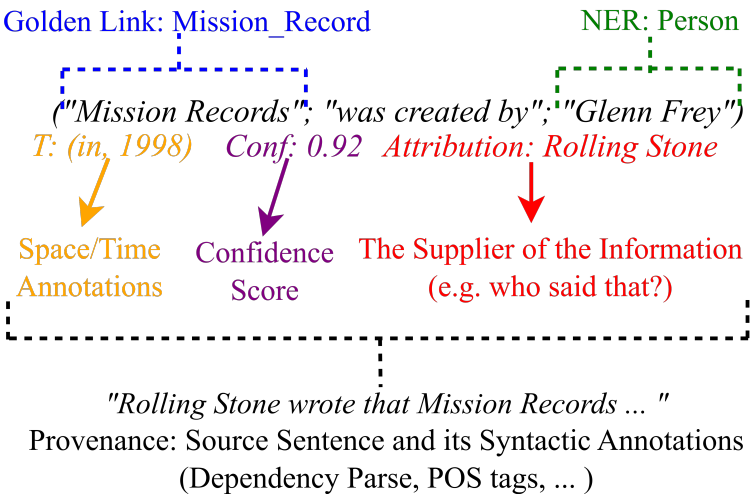


Fig. 4.1 Example of an OpenIE triple from OPIEC along with its annotations

also retained the original links found in Wikipedia articles. OPIEC, however, was constructed from a more recent version of Wikipedia, which means that the information found in OPIEC can be used to add more recent information to WiSeNet (e.g. in WiSeNet there is no triple indicating that Emmanuel Macron is president of France). Moreover, OPIEC was constructed with the OpenIE system MinIE-SpaTe, which has relatively high recall (discussed in section 3.8), thus increasing the potential of adding new information to WiSeNet as well. Finally, OPIEC contains many syntactic and semantic annotations, which means that even if WiSeNet contains some fact that is also present in OPIEC, the semantic annotations of OPIEC can provide additional semantic context to the WiSeNet fact (e.g. entity types for the arguments).

To analyze OPIEC's content and its potential usefulness for downstream applications, we performed a detailed data profiling study of the OPIEC corpus. We observed that a substantial fraction of the OpenIE extractions was either not self-contained (e.g., because no anaphora resolution was performed) or overly specific (e.g., because arguments were complex phrases). Since these extractions are more difficult to work with, we created the *OPIEC-Clean* subcorpus (104M triples), in which we only retained triples that express relations between named entities (e.g. *Alan Turing*) or concepts (e.g. *dolphin*). In particular, OPIEC-Clean contains triples in which arguments are either named entities (as recognized by an NER system), match a Wikipedia page title (e.g., concepts such as *political party* or *movie*), or link directly to a Wikipedia page. Although OPIEC-Clean is substantially smaller than the full OPIEC corpus, it is nevertheless four times larger than the largest prior OpenIE corpus (KB-Unify).

As discussed in section 2.1, OpenIE extractions are consisted of surface patterns, which means that their semantic content is not disambiguated. Contrary to OpenIE, traditional

	# triples (millions)	# unique arguments (millions)	# unique relations (millions)	disamb. args (aut./gold)	confi-dence	prove-nance	syntactic annotat.	semantic annotat.
ReVerb	14.7	2.2	0.7	- / -	✓	✓	✓	-
ReVerb-Linked	3.0	0.8	0.5	✓ / -	-	-	-	-
PATTY (Wiki)	15.8	0.9	1.6	✓ / -	-	-	-	-
WiseNet 2.0	2.3	1.4	0.2	- / ✓	-	-	-	-
DefIE	20.3	2.5	0.3	✓ / -	-	-	-	-
KB-Unify	25.5	2.1	2.3	✓ / -	-	-	-	-
OPIEC	341.0	104.9	63.9	- / -	✓	✓	✓	✓
OPIEC-Clean	104.0	11.1	22.8	- / -	✓	✓	✓	✓
OPIEC-Linked	5.8	2.1	0.9	- / ✓	✓	✓	✓	✓

Table 4.1 Available OpenIE corpora and their properties. All numbers are in millions. Syntactic annotations include POS tags, lemmas, and dependency parses. Semantic annotations include attribution, polarity, modality, space, and time.

KBs contain semantically disamiguated relations and arguments. In Chapter 5, we analyze alignments between OPIEC on the one hand and well-established KBs on the other (namely, DBpedia and YAGO, which were also constructed from Wikipedia). Such analysis is possible with OPIEC because 1) it retains the golden links found in the Wikipedia articles; 2) provides the necessary compactness of extractions discussed in Chapter 3; and 3) provides wide range of semantic annotations, which helps to understand the surface patterns better. For more elaborate discussion on the alignments between OpenIE triples of OPIEC and KBs constructed from the same resource (DBpedia and YAGO), see Chapter 5.

Along with the OPIEC corpus as well as the OPIEC-Clean and OPIEC-Linked subcorpora, we release the codebase used to construct the corpus as well as a number of derived resources, most notably a corpus of open relations between arguments of various entity types along with their frequencies. We believe that the OPIEC corpus is a valuable resource for future research on automated knowledge base construction.

4.2 Related Corpora

In recent years, many structured information resources were created from semi-structured or unstructured data. In this chapter, we focus on large-scale OpenIE corpora, which do not make use of a predefined set of arguments and/or relations. OpenIE corpora complement more targeted resources such as knowledge bases (e.g., DBpedia, YAGO) or NELL [Mitchell et al. 2018], as well as smaller, manually crafted corpora such as the one of Stanovsky and

Dagan [2016]; see also more elaborate discussion in Section 5.2. An overview of the OPIEC corpus, its subcorpora, and related OpenIE corpora is given in Table 4.1.

The largest prior corpus is KB-Unify, which consists of 25.5M triples having roughly 2.1M distinct arguments and 2.3M distinct relations. Both OPIEC (341M triples, 105M distinct arguments and 64M distinct relations) and OPIEC-Clean (104M triples, 11M arguments and 23M relations) are significantly larger than KB-Unify, both in terms of number of triples as well as in terms of distinct arguments and relations. One of the reasons for this size difference is that the MinIE extractor, which we used to create the OPIEC corpus, produces more extractions than the extractors used to create prior resources. The OPIEC corpus—but not OPIEC-Clean and OPIEC-Linked—also contains *all* extractions produced by MinIE unfiltered, whereas most prior corpora use filtering rules aiming to provide higher-quality extractions (e.g., frequency constraints or thresholds of extraction confidence scores).

Most of the available corpora use automated methods to disambiguate entities (e.g., w.r.t. a reference knowledge base). On the one hand, such links are very useful because the ambiguity of the OpenIE triples is restricted to the open relations only, while the OpenIE arguments become disambiguated. On the other hand, the use of automated entity linkers may introduce errors and—perhaps more importantly—restricts the corpus only to arguments that can be confidently linked. We did not perform automatic disambiguation in OPIEC, although we retained Wikipedia links when present (similar to WiseNet). Since these links are provided by humans, we consider them as *golden disambiguation links*. The OPIEC-Linked subcorpus contains almost 6M triples from OPIEC in which both arguments are disambiguated via such golden links.

A key difference between OPIEC and prior resources is in the amount of metadata provided for each triple. First, only ReVerb and OPIEC provide confidence scores for the extractions. The confidence score measures how likely it is that the triple has been extracted correctly (but not whether it is actually true). For example, given the sentence *"Bill Gates is a founder of Microsoft."*, the extraction *("Bill Gates"; "is founder of"; "Microsoft")* is correct, whereas the extraction *("Bill Gates"; "is"; "Microsoft")* is not. Since OpenIE extractors are bound to make extraction errors, the extractor confidence is an important signal for downstream applications [Dong et al. 2014]. Similarly, OPIEC provides provenance information (i.e., information about where the triple was extracted from), which is not provided in many prior resources. In particular, OPIEC provides the Wikipedia page ID from where the triple was extracted from as well as the sentence number within the Wikipedia page (e.g. *sentence number = 2* indicates that the triple was extracted from the second sentence of the article). In addition, OPIEC provides many other annotations which are not provided in most of the resources, such as named entity type of the arguments (when present), syntactic

information about the provenance sentence (e.g., dependency parse, POS tags, etc.) and semantic annotation of the triple itself (e.g., factuality and attribution). An example of an OPIEC triple along with its annotations is shown on Figure 4.1 and a full list of all the meta-data is shown in Table 4.2.

Besides the compact extractions provided by MinIE-SpaTe, one of the reasons why we chose MinIE-SpaTe to construct OPIEC is that it provides syntactic and semantics annotations for its extractions. Syntactic annotations include part-of-speech tags, lemmas, and dependency parses. Semantic annotations include attribution (source of information according to sentence), polarity (positive or negative), modality (certainty or possibility), space and time. The use of semantic annotations simplifies the resulting triples significantly and provides valuable contextual information. More elaborate discussion is presented in Section 4.3.

4.3 Corpus Construction

OPIEC was constructed from all the articles of the English Wikipedia dump of June 21, 2017. Figure 4.2 gives an overview of the pipeline that we used. First, in the preprocessing step, we take a Wikipedia dump, which is then processed to extract only the textual part of every Wikipedia article. We also keep the original links found in each Wikipedia article as meta-data. Next, we process the text with NLP pipeline, providing syntactic and semantic annotations (POS tags, dependency parse, NER types, etc.). The output of the NLP pipeline is then fed to our OpenIE system—MinIE-SpaTe—, which uses the syntactic and semantic annotations for producing the extractions. In order to prevent breaking-up words of entities in the arguments, we then postprocess the resulting triples, by rearranging the words in the arguments and relations according to the meta-data about the span of the links. At this point, we already have produced the OPIEC corpus. Finally, we apply filters for producing the subcorpora OPIEC-Clean and OPIEC-Linked.

The pipeline is (apart from the preprocessing step) not specific to Wikipedia and thus can be used with other natural language text datasets as well (e.g. newswire corpora). We used Apache Spark [Zaharia et al. 2016] to distribute the corpus construction across a computing cluster, so that large datasets can be handled more effectively. We released the entire codebase along with the actual OPIEC corpora.

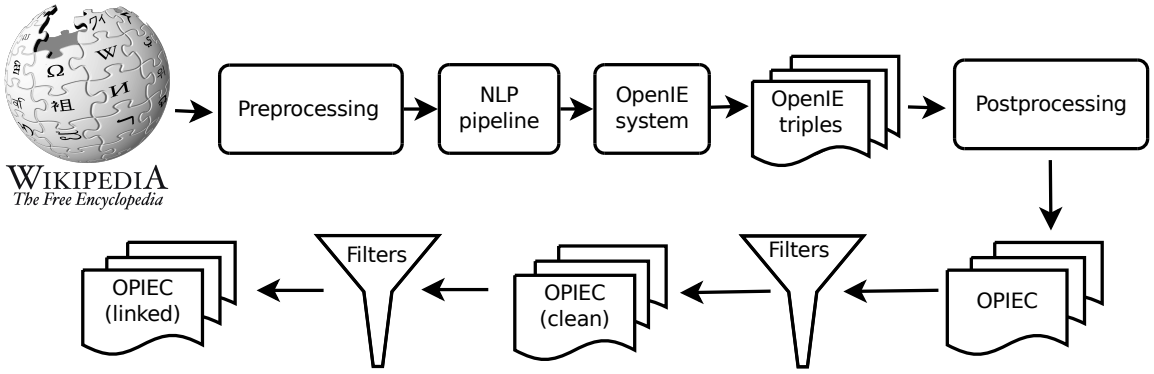


Fig. 4.2 Corpus construction pipeline

4.3.1 Preprocessing

We used a modified version of WikiExtractor[2] to extract the plain text from Wikipedia pages. In particular, we modified WikiExtractor such that it retains internal Wikipedia links from within the article's text to other Wikipedia articles. These links are annotated by humans, which makes them *golden disambiguation links*. The links are provided as additional metadata in the form of (*span*, *target page*) annotations. If needed, custom entity linkers can be inserted into the pipeline by providing link annotations in such format.

Wikipedia generally does not link the first phrase of an article to the article page itself. For example, the page on New Hampshire starts with *"New Hampshire is a state in ..."*, where the starting phrase *"New Hampshire"* is not linked. The first few sentences of a Wikipedia article are very important, because they contain definitional knowledge about the entity (or the concept) for which the article is about[3]. To avoid losing this important information, we link the first phrase of the first sentence of the Wikipedia article that *exactly* matches the Wikipedia page name (if any) to that same Wikipedia page.

Keeping the links from Wikipedia page articles, however, introduces inevitable bias in the data. One type of bias stems from Wikipedia's policy on how to place links within the articles, which encourages its contributors to avoid overlinking the articles. In Wikipedia's manual of style, an *overlinked article* is defined as an article that "contains an excessive number of links, making it difficult to identify links likely to aid the reader's understanding significantly". [4] They also discourage repetition of links within the articles, meaning that a link within a Wikipedia article should ideally appear only once. Such linking strategy results in links appearing mostly at the beginning of the articles, where the definitional knowledge

[2] https://github.com/attardi/wikiextractor
[3] Delli Bovi et al. [2015b] refer to such type of sentences as "textual definition about an entity or a concept"
[4] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking#What_generally_should_not_be_linked

is found. This suggests that the information which is found at the later parts of an article usually does not get links, leading to OpenIE extractions which contain links mostly for the starting parts of the articles. Such issues could be addressed with subsequent use of effective entity linking tools, which we leave for future work.

4.3.2 NLP Pipeline

We ran an NLP pipeline on the preprocessed Wikipedia articles by using CoreNLP [Manning et al. 2014], version 3.8.0. We performed tokenization, sentence splitting, part-of-speech tagging [Toutanova et al. 2003], lemmatization, named entity recognition (NER) [Finkel et al. 2005], temporal tagging [Chang and Manning 2012], and dependency parsing [Chen and Manning 2014]. The resulting corpus of the NLP pipeline is the entire English Wikipedia with linguistic annotations (POS tags, dependency parses, NER types, etc.), which we also released as an additional corpus (WikiNLP).

4.3.3 OpenIE System

Next, we feed the NLP annotations to OPIEC's underlying OpenIE system. The OPIEC pipeline uses MinIE-SpaTe as an OpenIE system, because it provides compact and semantically annotated OpenIE extractions. We use MinIE's safe mode as an underlying system for MinIE-SpaTe, because we want to avoid losing precision on the extractions at the expense of compactness (for more elaborate discussion on the precision-compactness trade-off, refer to Section 3.8).

4.3.4 Postprocessing

In a final postprocessing step, we rearranged triples such that links within the triples are not split across its constituents. For example, the triple *("Peter Brooke"; "was member of"; "Parliament")* produced by MinIE splits up the linked phrase "member of Parliament", which results in an incorrect link for the object (since it does not link to *Parliament*, but to *member of parliament*). We thus rewrite such triples to *("Peter Brooke"; "was"; "member of Parliament")*.

4.3.5 Provided Metadata

All metadata collected in the pipeline are retained for each triple, including provenance information (provenance sentence, Wikipedia article ID of the provenance sentence, position of the provenance sentence within the Wikipedia article), syntactic annotations (dependency

Field	Description
Article ID	Wikipedia article ID
Sentence	Sentence from which the triple was extracted, including annotations: 1) *Sentence number* within the Wikipedia page; 2) *Span* of the sentence within the Wikipedia page; 2) *Dependency parse*; 4) *Token* information. For each token, OPIEC provides POS tag, NER type, span, the original word found in the sentence, lemma, position of the token within the sentence, and the WikiLink object (contains offset begin/end index of the link within the article, the original phrase of the link, and the link itself).
Polarity	The polarity of the triple (either *positive* or *negative*)
Negative words	Words indicating negative polarity
Modality	The modality of the triple (either *possibility* or *certainty*)
CT/PS words	Words indicating the detected modality
Attribution	Attribution of the triple (if found) including attribution phrase, predicate, factuality, space and time
Quantities	Quantities in the triple (if found)
Subj. / rel. / obj.	Lists of tokens with linguistic annotations for subject, releation, and object of the triple
Dropped words	To minimize the triple and make it more compact, MinIE sometimes drops words considered to be semantically redundant words (e.g., determiners). All dropped words are stored here.
Time	Temporal annotations: information about TIMEX3 type, TIMEX3 xml, disambiguated temporal expression, original core words of the temporal expression, pre-modifiers/post-modifiers of the core words and temporal predicate
Space	Spatial annotations, containing information about the original spatial words, the pre/post-modifiers and the spatial predicate
Time/Space for phrases	Information about the temporal annotation on phrases. This annotation contains: 1) *modified word*: head word of the constituent being modified, and 2) *temporal/spatial* words modifying the phrase
Confidence score	The confidence score of the triple.
Canonical links	Canonical links for all links within the triple (follows redirections)
Extraction type	Either one of the clause types listed in ClausIE (SVO, SVA, ...), or one of MinIE's implicit extractions proposed in Chapter 3.4.2 (Hearst patterns, noun phrases modifying persons, etc.)

Table 4.2 Meta-data fields for each triple in OPIEC

parse tree, POS tags, lemmas ...), semantic annotations (NER types, temporal/spatial annotations, factuality, ...) and confidence scores. A full description of the provided metadata fields can be found in Table 4.2.

4.3.6 Filtering

We constructed the OPIEC-Clean and OPIEC-Linked subcorpus by filtering the OPIEC corpus. In general, OPIEC-Clean only retains triples between entities (e.g. *Richard Feynman*) or concepts (e.g. *chair*), whereas OPIEC-Linked only retains triples in which both arguments are linked. The filtering rules are described in more details in the following section.

4.4 Statistics

Basic statistics such as corpus sizes, frequency of various semantic annotations, and information about the length of the extracted triples of OPIEC and its subcorpora are shown in Table 4.3. We first discuss the properties of the OPIEC corpus, then we describe how we constructed the OPIEC-Clean and OPIEC-Linked subcorpora, and finally we provide more in-depth statistics.

4.4.1 The OPIEC Corpus

The OPIEC corpus contains all extractions produced by MinIE-SpaTe. We analyzed these extractions and found that a substantial part of the triples are more difficult to handle by downstream applications. We briefly summarize the most prevalent cases of such triples; all these triples are filtered out in OPIEC-Clean.

First of all, a large part of the triples are under-specific in that additional context information from the extraction source is required in order to obtain a coherent piece of information. By far the main reason for under-specificity is lack of coreference information. In particular, 22% of the arguments in OPIEC are personal pronouns, such as in the triple *(“He”; “founded”; “Microsoft”)*. Such triples are under-specific because provenance information is needed to resolve what *“He”* refers to. Similarly, about 1% of the triples have determiners as arguments (e.g. *(“This”; “lead to”; “controversy”)*), and 0.2% Wh-pronouns (e.g. *(“what”; “are known as”; “altered states of consciousness”)*). Coreference resolution in itself is a difficult problem, but the large fraction of such triples shows that coreference resolution is important to further boost the recall of OpenIE systems.

Another problem for OpenIE systems are entity mentions—most notably for works of art—that constitute clauses. For example, the musical *“Zip Goes a Million”* may be

	OPIEC	OPIEC-Clean	OPIEC-Linked
Total triples (millions)	341.0	104.0	5.8
Triples with semantic annotations	166.3 (49%)	51.46 (49%)	3.37 (58%)
negative polarity	5.3 (2%)	1.33 (1%)	0.01 (0%)
possibility modality	13.9 (4%)	3.27 (3%)	0.04 (1%)
quantities	59.4 (17%)	15.91 (15%)	0.45 (8%)
attribution	6.4 (2%)	1.44 (1%)	0.01 (0%)
time	65.3 (19%)	19.66 (19%)	0.58 (10%)
space	61.5 (18%)	22.11 (21%)	2.64 (45%)
space OR time	111.3 (33%)	37.22 (36%)	3.01 (52%)
space AND time	15.4 (5%)	4.54 (4%)	0.20 (4%)
Triple length in tokens ($\mu \pm \sigma$)	7.66 ± 4.25	6.06 ± 2.82	6.45 ± 2.65
subject ($\mu \pm \sigma$)	2.12 ± 2.12	1.48 ± 0.79	1.92 ± 0.94
relation ($\mu \pm \sigma$)	3.01 ± 2.47	3.10 ± 2.56	2.77 ± 2.14
object ($\mu \pm \sigma$)	2.52 ± 2.69	1.48 ± 0.79	1.76 ± 0.94
Confidence score ($\mu \pm \sigma$)	0.53 ± 0.23	0.59 ± 0.23	0.61 ± 0.26

Table 4.3 Statistics for different OPIEC corpora. All frequencies are in millions. We count triples with annotations (not annotations directly). Percentages refer to the respective subcorpus.

interpreted as a clause, leading to the incorrect extraction *("Zip"; "Goes"; "a Million")*. A preliminary study showed that almost 30% of all the OPIEC triples containing the same recognized named entity in both subject and object were of such a type. These cases constitute around 1% of OPIEC.

Finally, a substantial fraction of the triples in OPIEC has complicated expressions in its arguments. Consider for example the sentence *"John Smith learned a great deal of details about the U.S. Constitution"*. From this input sentence, MinIE-SpaTe (with MinIE's safe mode) extracts the triple *("John Smith"; "learned"; "great deal of details about U.S. Constitution")*, which has a complicated object and is thus difficult to handle. One possible way to achieve more acceptable level of compactness is to use a mode of MinIE that performs more aggressive modes of minimization. For instance, if we use the aggressive mode of MinIE, then we would get the triple *("John Smith"; "learned"; "deal")*, which we would consider to be incorrectly extracted, because the phrase *"U.S. Constitution"* carries the core semantic information in the argument. The complicated extraction, however, could be further minimized to more compact form, which we would consider to be correctly extracted. For instance, a minimized variant such as *("John Smith"; "learned about"; "U.S. Constitution")* looses some information, but it expresses the main intent in a simpler way. Currently, MinIE

does not support versions that would handle such complex cases without suffering significant loss of precision.

The above difficulties make the OPIEC corpus a helpful resource for research on improving or reasoning with complex OpenIE extractions rather than for downstream tasks.

4.4.2 The OPIEC-Clean Corpus

The OPIEC-Clean corpus is obtained from OPIEC by simply removing underspecified and complex triples. In particular, we consider a triple *clean* if the following conditions are met:

(i) each argument is either linked, an entity recognized by the NER tagger, or matches a Wikipedia page title

(ii) links or recognized named entities are not split up across constituents

(iii) the triple has a non-empty object.

Conditions (i) and (ii) rule out the complex cases mentioned in the previous section. Note that we ignore quantities (but no other modifiers) when checking condition (i). For example, the triple *("Q_1 electric locomotives"; "were ordered from"; "Alsthom")* with Q_1 = *"Three"* is considered clean; here *"electric locomotives"* holds a link to *TCDD E4000* and *"Alsthom"* holds a link to *Alsthom*.

MinIE is a clause-based OpenIE system and can produce extractions for so-called SV clauses: these extractions consist of only a subject and a relation, but no object. 3.5% of the triples in OPIEC are of such type. An example is the triple *("Civil War"; "have escalated"; "")*. Although such extractions may contain useful information, we exclude them via condition (iii) to make the OPIEC-Clean corpus uniform.

Roughly 30% of the triples (104M) in OPIEC are clean according to the above constraints. Table 4.3 shows that clean triples are generally shorter on average and tend to have a higher confidence score than the full set of triples in OPIEC. The OPIEC-Clean corpus is easier to work with than the full OPIEC corpus; it is targeted towards both downstream applications and research in automated knowledge base construction.

4.4.3 The OPIEC-Linked Corpus

The OPIEC-Linked corpus contains only those triples from OPIEC-Clean in which both arguments are disambiguated with Wikipedia links. Although the corpus is much smaller than OPIEC-Clean (5.8M triples, i.e., roughly 5.5% of OPIEC-Clean), it is the largest corpus to date with golden disambiguation links for the arguments.

Fig. 4.3 Example of OPIEC-Linked triple

Such corpus is useful for downstream tasks (e.g. Open Link Prediction [Broscheit et al. 2020]), because the entities on both sides are disambiguated. For example, the OpenIE triple *("Michael Jordan"; "grew up in"; "Wilmington")* could be highly ambiguous, because in Wikipedia there are 14 people named *Michael Jordan* and 23 places named *Wilmington*. OPIEC-Linked removes such ambiguities by providing golden entity links for the arguments, thus narrowing down the meaning of an argument to a single disambiguated entity (example shown on Figure 4.3). We use OPIEC-Linked mainly to compare OpenIE extractions with the information present in the DBpedia and YAGO knowledge bases. For more elaborate discussion, see Chapter 5.

A common problem with OpenIE systems is that they often extract triples, which are considered to be overly-specific.

4.4.4 Semantic Annotations

About 49% of all triples in OPIEC contain some sort of semantic annotation (cf. Table 4.3). For OPIEC-Clean, the proportion of semantically annotated triples remains roughly the same. On the other hand, in OPIEC-Linked, the fraction of semantically annotated triples increases to 58%. Most of the semantic annotations refer to quantities, space or time; these annotations provide important context for the extractions.

There is a significantly smaller amount of negative polarity and possibility modality annotations. Similarly, the OPIEC triples rarely contain attribution information. One reason for the lack of such annotations may be in the nature of the Wikipedia articles, which aim to contain encyclopedic, factual statements and are thus more rarely negated or hedged. This possible explanation is more evident when compared with the experimental study in Chapter 3, where we observed that these semantic annotations produced by MinIE (negative or possibility factuality and attribution) have significantly higher coverage compared to the coverage observed in OPIEC. Those experiments, however, were performed on the New York Times corpus, which is a corpus of newswire data. These observations suggest that the coverage of semantic annotations might differ across different domains.

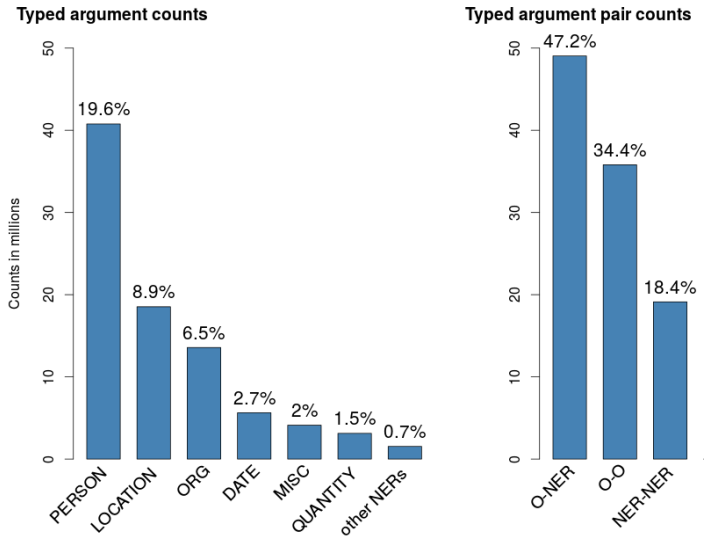


Fig. 4.4 Distribution of NER types for arguments and argument pairs in OPIEC-Clean. Here "O" refers to arguments that are not recognized as a named entity.

The distribution of semantic annotations is similar for OPIEC and OPIEC-Clean, though it significantly differs for OPIEC-Linked. In particular, we observed a drop in quantity annotations in OPIEC-Linked because most of the linked phrases do not contain quantities. The fraction of spatial triples in OPIEC-Linked is much higher than the rest of the corpora. The reason for such coverage is because Wikipedia contains many pages for locations, which contain linked text about other locations (e.g. a city being a capital of a country) thus resulting in many triples with spatial reference (for example, the OpenIE triple *("Berlin"; "is capital of"; "Germany")*).

4.4.5 NER Types and Frequent Relations

For OPIEC-Clean, Figure 4.4 shows the fraction of arguments and argument pairs that are recognized as named entities by the NER tagger, along with the NER type distribution of the arguments.

Out of the around 208M arguments, roughly 42% are recognized named entities. The most frequent NER type is *person*, followed by *location* and *organization*. The remaining NER types are not that frequent (less than 3% each). On the other hand, 58% of the arguments are not typed. These are mostly concepts (more precisely, strings that match Wikipedia pages not referring to an entity) and are thus not recognized by the NER system. The top-10 most frequent arguments which are not typed are the words *film, population, city, village, father, song, time, town, album* and *company*, with frequencies varying between 427k and 616k (Table 4.4).

argument	frequency
"film"	613,955
"population"	564,389
"city"	512,443
"village"	501,051
"father"	468,813
"song"	453,388
"time"	448,951
"town"	431,641
"album"	428,775
"company"	427,011

Table 4.4 Top-10 most frequent arguments which are not typed. Frequency in OPIEC-Clean.

Figure 4.4 also reports the fraction of triples in which none, one, or both arguments are recognized as a named entity. We found that 18% of the triples (19M) in OPIEC-Clean have two typed arguments, and around 66% of the triples (68M) have at least one typed argument. Thus the majority of the triples involves named entities. 34% of the triples do not have recognized named entity arguments.

Table 4.5 shows the most frequent open relations between arguments recognized as NERs (which in turn are the top-9 most frequent argument type pairs). We will analyze some of the open relations in more detail in Chapter 5.2. For now, note that the most frequent relation between persons is *"have"*, which is highly polysemous. Other relations, such as "*marry*" and *"be son of"*, are much less ambiguous. We provide all open relations between recognized argument types as well as their frequencies with the OPIEC-Clean corpus.

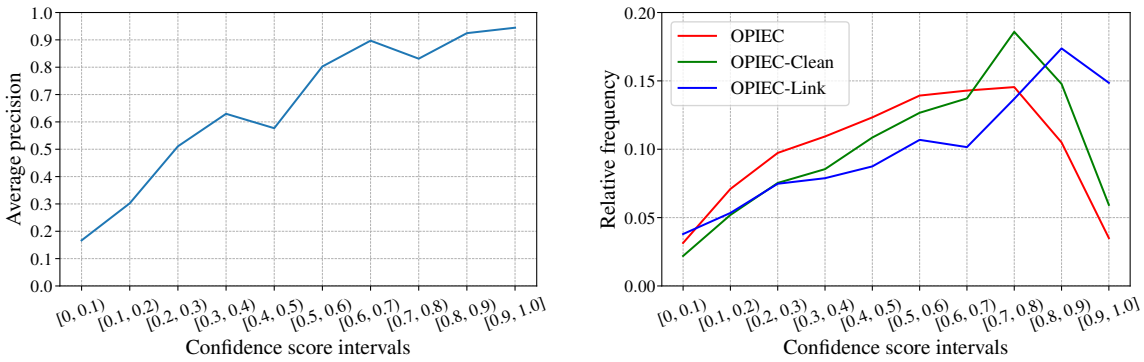
4.4.6 Precision and Confidence Score

Each triple in the OPIEC corpora is annotated with a confidence score indicating if the triple was correctly extracted (for details of the confidence score, see Section 3.7.4). To evaluate the accuracy of the confidence score, we took an independent random sample of triples (500 in total) from OPIEC and manually evaluated the correctness of the triples following the procedure explained in Appendix A. We found that 355 from the 500 triples (71%) were correctly extracted. Next, we bucketized the triples by confidence score into ten equi-width intervals and calculated the precision within each interval; see Figure 4.5a. We found that the confidence score is highly correlated to precision (Pearson correlation of $r = 0.95$) and thus provides useful information.

The distribution in confidence scores across the various corpora is shown in Figure 4.5b. We found that OPIEC-Clean and, in particular, OPIEC-Link contain a larger fraction of

	PERSON-PERSON		LOCATION-LOCATION		PERSON-LOCATION	
total freq.		*2,890,326*		*2,887,577*		*2,455,670*
	"have"	(130,019)	*"be in"*	(2,126,562)	*"be bear in"*	(203,091)
	"marry"	(49,405)	*"have"*	(40,298)	*"die in"*	(37,952)
	"be son of"	(40,265)	*"be village in administrative district of"*	(9,130)	*"return to"*	(36,702)
	"be daughter of"	(37,089)	*"be north of"*	(3,816)	*"move to"*	(36,072)
	"be bear to"	(29,043)	*"be suburb of"*	(3,291)	*"be in"*	(25,847)
	"be know as"	(25,607)	*"be west of"*	(3,238)	*"live in"*	(22,399)
	"defeat"	(22,151)	*"be part of"*	(3,188)	*"grow up in"*	(17,571)
	"be marry to“	(21,694)	*"be municipality in”*	(3,175)	*"have“*	(13713)
	"meet”	(20,491)	*"defeat“*	(3,137)	*"leave”*	(12,573)
	"be replace by“	(17,949)	*"include”*	(2,983)	*"represent“*	(10,025)
	PERSON-ORG.		ORG.-LOCATION		PERSON-DATE	
total freq.		*2,320,462*		*1,660,278*		*1,331,183*
	”be member of“	(46,446)	*”be in“*	(832,292)	*”die in“*	(61,706)
	”have“	(43,643)	*”be of“*	(68,837)	*”die on“*	(58,443)
	”join“	(36,544)	*”be at“*	(66,616)	*”be bear on“*	(37,597)
	”attend“	(35,315)	*”be from“*	(16,931)	*”be bear in“*	(27,962)
	”be president of“	(33,039)	*”be for“*	(10,476)	*”be bear“*	(17,564)
	”graduate from“	(23,642)	*”have“*	(9,621)	*”die“*	(13,459)
	”be educate at“	(16,738)	*”be base in“*	(8,677)	*”have death in“*	(13,176)
	”be chairman of“	(15,954)	*”be on“*	(7,094)	*”have“*	(8,485)
	”be founder of“	(14,115)	*”be near“*	(6,021)	*”be found in“*	(6,455)
	”found“	(13,877)	*”be headquarter in“*	(3,200)	*”be release in“*	(6,304)
	ORG.-ORG.		PERSON-MISC		ORG.-DATE	
total freq.		*697,253*		*610,298*		*457,254*
	”have“	(46,739)	*”be“*	(104,727)	*”be found in“*	(25,493)
	”be know as“	(11,357)	*”have“*	(22,887)	*”be establish in“*	(17,347)
	”be member of“	(6,429)	*”win“*	(12,435)	*”be form in“*	(6,540)
	”be part of“	(5,463)	*”speak“*	(2,937)	*”open in“*	(6,491)
	”defeat“	(5,302)	*”play in“*	(2,100)	*”be build in“*	(4,206)
	”acquire“	(5,044)	*”receive“*	(2,084)	*”be create in“*	(3,372)
	”become“	(4,844)	*”join“*	(2,014)	*”open on“*	(2,935)
	”beat“	(4,403)	*”compete in“*	(1,864)	*”be know“*	(2,793)
	”be subsidiary of“	(4,155)	*”be award“*	(1,852)	*”close in“*	(2,629)
	”be own by“	(4,038)	*”be induct into“*	(1,789)	*”be start in“*	(2,168)

Table 4.5 Most frequent open relations between NERs (as recognized by the NER tagger) in OPIEC-Clean



(a) Precision of confidence score

(b) Distribution of confidence scores

Fig. 4.5 Precision and distribution of confidence scores

high-confidence triples than the raw OPIEC corpus: these corpora are cleaner in that more potentially erroneous extractions are filtered out. Although triples with lower confidence score tend to be more inaccurate, they may still provide suitable information. We thus included these triples into our corpora; the confidence scores allow downstream applications to handle these triples as appropriate.

4.5 Use of OPIEC for Downstream Tasks

Such large OpenIE corpora are useful for many downstream tasks, especially if they contain disambiguated links. In particular, OPIEC-Linked was used for constructing the entity-aspect linking dataset EAL-D [Nanni et al. 2019]. Entity aspect links provide more fine-grained semantic context about a given entity. Nanni et al. [2018] present an illustrative example of an entity aspect link: consider the sentence *"I'm watching the debate between **Clinton** and **Sanders**"*. The entity mentions *"Clinton"* and *"Sanders"* refer to the entities *Hillary Clinton* and *Bernie Sanders* respectively. Linking these mentions with the corresponding entities, however, is too coarse-grained to fully capture the semantics of the mentions within the context of the sentence. A more precise linking would be the Wikipedia links *Hillary_Clinton#2016_presidential_campaign* and *Bernie_Sanders#2016_presidential_campaign* respectively, thus providing more fine-grained semantic context (aspect) of the entities. Such links are found in Wikipedia articles, thus making them available in OPIEC as well. By leveraging the information in the OpenIE triples from OPIEC-Linked, Nanni et al. [2019] released EAL-D, which is the largest entity-aspect linking dataset.

Moreover, a subset of OPIEC was leveraged for the newly proposed task of *Open Link Prediction* (OLP) [Broscheit et al. 2020]. The OLP task aims at answering queries in the form

of *(“subject”; “relation”; “?”)* or *(“?”; “relation; ”object“)*, where *”subject“* and *”object“* are surface patterns of entity mentions. Contrary to traditional link prediction tasks [Nickel et al. 2011; Bordes et al. 2013; Trouillon et al. 2016; Ruffinelli et al. 2020], in OLP the task is to predict the *mention* of an entity (which is a surface textual pattern) in the OpenIE triple, not the entity's underlying disambiguation link. In order to provide the necessary resources for OLP, Broscheit et al. [2020] used a subset of OPIEC. Using the disambiguation links in the OpenIE arguments for OLP data is crucial, because they can provide a dictionary of mentions, which can serve for testing the correctness of the predicted mention. To this end, Broscheit et al. [2020] published an OLP-tailored dataset, which contains 30M distinct OpenIE triples (with 2.5M entity mentions and 1M open relations). Approximately 1.25M of these unique OpenIE triples contain disambiguation links on both the subject and the object (i.e. they are a subset of OPIEC-Linked).

4.6 Conclusions

We created OPIEC, a large open information extraction corpus extracted from Wikipedia. OPIEC consists of hundreds of millions of triples, along with rich metadata such as provenance information, syntactic annotations, semantic annotations, and confidence scores. In this chapter, we reported on a data profiling study of the OPIEC corpus as well as its subcorpora OPIEC-Clean and OPIEC-Linked. OPIEC-Clean reduces the noise of OPIEC, by selecting only the triples that have entities or concepts as arguments, thus avoiding noisy and overly-specific extractions. The disambiguated arguments of OPIEC-Linked reduce the inherent ambiguities of OpenIE extractions by providing unambiguous Wikipedia links to the arguments of the triples. The disambiguated arguments of OPIEC-Linked are important for other downstream tasks, such as aligning OpenIE triples with knowledge bases (KBs). For example, the OpenIE triple *(”Barack Obama“; ”was born in“; ”U.S.“)* is considered to be aligned with the KB triple (Barack Obama; dbp:birthPlace; U.S.). Such alignments are used for many downstream tasks, including KB population [Lin et al. 2020], link prediction [Gupta et al. 2019] and automated KB construction [Dong et al. 2014].

OPIEC, however, suffers from several disadvantages. One of them is the significant amount of underspecific triples (e.g. triples with personal pronouns as arguments). Though such triples can be correctly extracted, they are nevertheless not informative. For example, in the OpenIE triple *(”He“;”co-founded“; ”Microsoft“)* it is not clear to what entity the argument *”he“* is referring to. On the other hand, OPIEC also contains many triples which are considered to be overly-specific. For example, the OpenIE triple *(”John Smith“; ”learned“; ”great deal of detail about U.S. Constitution“)* can be rewritten into more compact form *(”John*

Smith"; "learned about"; "U.S. Constitution") without losing the main semantic content. Other studies suggest that efficient postprocessing of both underspecific and overly-specific triples is needed for making such corpora more useful for downstream tasks. For example, Zhan and Zhao [2020] point out that applying coreference resolution on such underspecific triples is useful for generating training data for training neural models for OpenIE. On the other hand, the experimental study by [Lin et al. 2020] suggests that using more compact OpenIE triples is preferable for using OpenIE for KB population. Therefore, we believe that more work is needed in this direction for improving OpenIE systems.

Another drawback of the triples in OPIEC stems from the confidence score. Even though we found that the correctness of the extractions highly correlates with the confidence score (Section 4.4.6), still, we sometimes find perfectly extracted triples that get relatively low confidence score. For example, OPIEC contains the OpenIE triple *("Monique Leyrac"; "is actress from"; "Quebec")*, which is correctly extracted. While one would expect that such correctly extracted and compact triple should get much higher confidence score, still, this triple gets a confidence score of 0.34. In a preliminary study, we found that one of the main reasons for such misleading scores is because they are extracted from conjunctions of sentences. In particular, MinIE-SpaTe tends to assign lower confidence scores of such extractions, because processing conjunctions for dependency parsing is difficult and is more prone to errors in the dependency parse tree [Chen and Manning 2014]. As a consequence, these errors are reflected in the training data for the confidence score (Section 3.7.4), and the classifier learns to reduce the confidence score once we have conjunctions in the sentence. Such correctly extracted triples with relatively low confidence score are somewhat misleading, because selecting only high-confidence extractions of a large OpenIE corpus—as used by many downstream tasks—could lead to losing significant amount of correctly extracted information, thus hurting recall.

Finally, due to the guidelines of Wikipedia about placing links within the articles by Wikipedia contributors, we found that the triples in OPIEC that contain links usually come from the introduction parts of the articles. With such limitations, the information written in the middle (or bottom) part of an article is usually not linked, thus introducing bias in the data (details are discussed in Section 4.3.1). This problem could be avoided by the use of efficient entity linking systems, which would increase the number of triples for OPIEC-Linked, thus reducing further the ambiguity of OPIEC in general.

As for subsequent work, OPIEC still remains the largest publicly-available OpenIE corpus to date. The OPIEC framework is designed in such manner that can be applied to any large-scale text corpora, which has the potential of producing much larger OpenIE corpora. Moreover, we believe that future work that would address the above mentioned issues would

produce OpenIE corpus of higher quality and utility. We hope that the OPIEC corpus, its subcorpora, derived statistics, as well as the codebase used to create the corpus are a valuable resource for automated KB construction and downstream applications.

In the next chapter, we analyze to what extent OPIEC-Linked overlaps with the DBpedia and YAGO knowledge bases, which are constructed from the same resource as OPIEC (Wikipedia). Our study indicates that most open facts do not have counterparts in the KB such that OpenIE corpora contain complementary information. For the information that overlaps, open relation are often more specific, more generic, or simply correlated to KB relations (instead of semantically equivalent). In the next chapter, we also manually investigate the distant supervision assumption (DSA) within the context of OpenIE, which in prior work is used as an underlying assumption for other downstream tasks or for constructing OpenIE systems themselves. Again, for this study, it is crucial that the arguments of the OpenIE triples are disambiguated, and we used OPIEC-Linked for this reason. In particular, we evaluated the semantics of the open relations w.r.t. the KB relations and found that OpenIE relations are semantically more specific. In the same spirit, we manually evaluated if *any* OpenIE triple can be expressed with a *single* KB fact. We found that while a single KB fact is often capable of expressing one OpenIE triple, though with losing more specific information contained in the OpenIE triple. We found that such issues can be significantly improved by the use of KB formulas. A more elaborate discussion for these issues is presented in Chapter 5.

Chapter 5

On Aligning OpenIE Extractions with Knowledge Bases: A Case Study

OpenIE extractions are consisted of surface patterns, which means that their semantic content is not disambiguated. Contrary to OpenIE, traditional KBs contain semantically disambiguated relations and arguments. The structure of both an OpenIE extraction and a KB fact is (usually) in the form of triple *(subj, relation, object)*. This enables for OpenIE extractions to be aligned with KB facts for a number of downstream tasks, such as automated KB construction [Dong et al. 2014], KB population [Lin et al. 2020] and KB extension [Dutta et al. 2015]. The semantics of such alignments, however, was not studied more closely in previous work. Usually, the quality of the alignments' semantics is measured w.r.t. a specific downstream task (e.g. KB population [Lin et al. 2020] or link prediction [Gupta et al. 2019]). In this chapter, we study such alignments directly (without any specific downstream task in mind) and manually analyze their semantics.

5.1 Introduction

OpenIE triples contain surface relations, which often makes their semantics ambiguous [Gashteovski et al. 2019]. This poses difficulties for OpenIE output to be used in downstream applications [Broscheit et al. 2017]. By contrast, KB relations have precise semantics and are more machine-readable [Banko and Etzioni 2008; Bizer et al. 2009]. To bridge this gap between OpenIE and KBs, many methods were proposed for aligning OpenIE triples with reference KBs. In such work, the goal is to associate an OpenIE triple with an *existing* KB fact (assuming they have the same disambiguated arguments), such that both triples have equivalent semantics. For example, the OpenIE triple *(Jeff Bezos; "be CEO of";*

Amazon.com)[1] has equivalent semantics with the KB fact (Jeff Bezos; dbo:ceo; Amazon.com). These alignment methods are primarily used for bootstrapping OpenIE systems [Mausam et al. 2012; Lockard et al. 2019], because training data for learning extraction rules (or for training neural models) are scarce. A simple approach for bootstrapping is to compare the argument pairs against a KB. Conceptually, if the argument pair of the OpenIE triple also exists in the reference KB, it is assumed that the OpenIE extraction contains information equivalent with the KB fact having the same argument pair. This implies that the OpenIE triple is correctly extracted and therefore it is used as a positive example for the training data [Weld et al. 2009]. Other line of work maps any OpenIE triple to a predefined KB schema [Soderland et al. 2013; Zhang et al. 2019]. For example, the OpenIE triple *(Emmanuel Macron; "be president of"; France)* could be mapped to (Emmanuel Macron; dbo:president; France) even if this fact is not necessarily present in the reference KB. Such methods are used for downstream tasks such as automatic KB construction [Dong et al. 2014] or KB extension [Dutta et al. 2015], because they are able to harness knowledge from natural language text to a target KB.

Such methods for aligning OpenIE triples with KBs are typically measured w.r.t. downstream task, such as KB population [Lin et al. 2020], link prediction [Gupta et al. 2019] or slot filling [Angeli et al. 2015]. In this chapter, we explore how the OpenIE triples from the OPIEC corpus are related to KBs constructed from the same resource (Wikipedia) w.r.t. their information content, without any specific downstream task in mind. Both OPIEC and the KBs (namely, DBpedia and YAGO) are resources which are automatically generated from the same domain—OPIEC from the textual data and the KBs from the semi-structured data of Wikipedia (infoboxes)—which makes the OPIEC comparable with the DBpedia and YAGO. To gain more insights into the semantics of such alignments, we analyze them more thoroughly and manually.

First, in section 5.2 we compare the content of OPIEC with the content of DBpedia [Bizer et al. 2009] and YAGO [Hoffart et al. 2013]. Such analysis is helpful for understanding to what extent does the information content of the OpenIE triples complements the KBs, which is important for tasks such as KB extension [Dutta et al. 2015]. Since such analysis is difficult to perform due to the openness and ambiguity of OpenIE extractions, we followed standard practice and used a simple form of distant supervision. In particular, we analyze the *OPIEC-Linked* subcorpus (5.8M triples), which contains only those triples in which both arguments are linked to Wikipedia articles, i.e., where we have golden labels for disambiguation (for details, refer to Chapter 4). We found that most of the facts between entities present in OPIEC-Linked cannot be found in DBpedia and/or YAGO, which shows the potential of

[1] the arguments *Jeff Bezos* and *Amazon.com* are disambiguated

harnessing knowledge from natural language text for tasks such as populating KBs. We also observed that OpenIE triples often differ in the level of specificity compared to knowledge base facts: in some cases the OpenIE triples are more specific than the KB fact, in others the OpenIE triples are more generic. Finally, we found that frequent open relations are generally highly polysemous. This suggests that direct mapping of open relations to KB relations should be avoided, because more context is needed for disambiguating the semantics of the open relations. Note that this study is not manual and the findings do not explain the underlying semantics of the alignments in details.

Second, we study the semantics of the alignments between OPIEC triples and DBpedia facts that have the same argument pair (section 5.3). Contrary to the study discussed in section 5.2, in section 5.3 we perform manual semantic analysis of an OpenIE triple w.r.t. the aligned KB fact[2]. Consequently, this study reveals more detailed findings of the semantics of such alignments. Consider the OpenIE triple *t: (Jeff Bezos; "is CEO of"; Amazon.com)* and the two possible KB alignments f_1 : (Jeff Bezos; dbo:ceo; Amazon.com) and f_2 : (Jeff Bezos; dbo:employer; Amazon.com). The KB fact f_1 has equivalent semantics with the OpenIE triple t, which is the distant supervision assumption within the context of OpenIE [Weld et al. 2009]. On the other hand, even though t and f_2 have the same argument pair, t is semantically more specific than f_2, because it provides additional information about Jeff Bezos being employed as a CEO. Therefore, f_2 expresses *some* information in t, but not all information. In our study, we always consider the best possible alignment (in the previous example, f_1 is considered to be the best alignment) and we investigate its semantics. Note that our goal is *not* to compare different alignment strategies. Rather, *we consider the best possible alignment* and the goal is to *investigate the limits* of such alignments. Investigating the limits of such alignments is important for understanding the limits of the distant-supervision assumption within the context of OpenIE in general, which is widely used for bootstrapping OpenIE systems [Weld et al. 2009; Yahya et al. 2014; Zhu et al. 2019]. Moreover, with the recent trend of neural OpenIE systems [Stanovsky et al. 2018; Cui et al. 2018; Zhan and Zhao 2020; Kolluru et al. 2020], generating large amounts of training data is important for training neural models, because labeled OpenIE data are scarce. Generating training data for neural models can be achieved through distant-supervision strategies as well [Cui et al. 2018], which also makes this study insightful for understanding the limits for generating training data with distant supervision. We found that these alignments are usually semantically related, but

[2]we use only DBpedia as a reference KB in this study, because: 1) DBpedia has much wider range of relations compared to YAGO; 2) the KB facts (including types) are completely extracted from Wikipedia (YAGO is partially derived from WordNet), which makes it more comparable with OPIEC which is also completely extracted from Wikipedia.

quite often the open relation is more specific, thus carrying more information than the KB fact.

Third, we study the expressibility of any OPIEC triple w.r.t. DBpedia by studying whether a given OpenIE triple can be mapped to a KB fact (Section 5.4). In this case, there might not be a known relation in DBpedia between the arguments of the OPIEC triple. We evaluate whether *any* OPIEC triple can be expressed with a *single* DBpedia fact. Consider the OpenIE triple *(Emmanuel Macron; "be president of"; France)*. DBpedia does not contain this fact, nevertheless, it can be fully expressed with (Emmanuel Macron; dbo:president; France) and partially expressed with (Emmanuel Macron; dbo:nationality; France). Such assumption is used for other downstream tasks, such as slot filling [Angeli et al. 2015]. We found that most of the OPIEC triples can be expressed with DBpedia facts, but many of them only partially. Moreover, large fraction of the partially expressible triples can be fully expressed with the use of KB formulas. For example, the OpenIE triple *(John F. Kennedy; "be grandchild of"; P. J. Kennedy)* can be partially expressed with the KB fact (John F. Kennedy; dbo:relative; P. J. Kennedy) and fully expressed with the KB formula:

$$\exists x : \text{(John F. Kennedy; dbo:parent; x)} \wedge \text{(x; dbo:parent; P. J. Kennedy)}.$$

The rest of the chapter is organized as follows: in Section 5.2 we compare the content between OPIEC and two reference knowledge bases (DBpedia and YAGO). Next, in Section 5.3 we perform a manual semantic analysis which compares the semantics of OPIEC triples and DBpedia facts having the same argument pairs. In Section 5.4 we study the expressibility of OPIEC triples with DBpedia. Finally, in Section 5.6 we discuss the main findings and conclusions of this chapter.

5.2 Analysis: Content Comparison of Alignments

In this section, we compare the information which is present in the OpenIE triples in OPIEC with the information present in the DBpedia [Auer et al. 2007] and YAGO [Hoffart et al. 2013] knowledge bases. Since all resources extract information from Wikipedia—OPIEC from the text and DBpedia as well as YAGO from the semi-structured parts of Wikipedia—, we wanted to understand whether and to what extent they are complementary. Such analysis is helpful for understanding the potential of harnesing knowledge from natural language text resources for tasks such as KB population or KB extension.

Generally, the disambiguation of OpenIE triples w.r.t. a given knowledge base is in itself a difficult problem. We avoid this problem here by (1) restricting ourselves to the

dbo:location		dbo:associatedMusicalArtist		dbo:spouse	
"be in"	(43,842)	*"be"*	(6,273)	*"be wife of"*	(1,965)
"have"	(3,175)	*"have"*	(3,600)	*"be"*	(1,308)
"be"	(1,901)	*"be member of"*	(740)	*"marry"*	(702)
"be at"	(1,109)	*"be guitarist of"*	(703)	*"be widow of"*	(479)
"be of"	(706)	*"be drummer of"*	(458)	*"have"*	(298)
"be historic home located at"	(491)	*"be feature"*	(416)	*"be husband of"*	(284)
"be national historic district located at"	(465)	*"be frontman of"*	(394)	*"be marry to"*	(281)
"be lake in"	(296)	*"be lead singer of"*	(254)	*"be consort of"*	(195)
"serve village of"	(291)	*"be singer of"*	(234)	*"be second wife of"*	(156)
"be base in"	(262)	*"be bassist of"*	(229)	*"be first wife of"*	(146)
"be from"	(177)	*"be vocalist of"*	(164)	*"be daughter of"*	(119)
"be historic home located near"	(176)	*"be former member of"*	(156)	*"have marry"*	(83)
"be near"	(165)	*"form"*	(151)	*"be queen consort of"*	(68)
"be located in"	(146)	*"have work with"*	(137)	*"be former wife of"*	(64)
"be headquarter in"	(126)	*"be in"*	(116)	*"have marriage to"*	(64)

Table 5.1 The most frequent open relations aligned to the DBpedia relations dbo:location, dbo:associatedMusicalArtist, and dbo:spouse in OPEIC-Linked

OPIEC-Linked corpus (for which we have golden entity links) and (2) focusing on statistics that do not require a full disambiguation of the open relations but are nevertheless insightful.

5.2.1 Alignment with Knowledge Bases

To align the OpenIE tripes from OPIEC-Linked to YAGO or DBpedia, we make use of the distant supervision assumption [Mintz et al. 2009]. For each open triple (s, r_{open}, o) from OPIEC-Linked, we search the KB for any triple of form (s, r_{KB}, o) or (o, r_{KB}, s). Here s and o refer to disambiguated entities, whereas r_{open} refers to an open relation and r_{KB} to a KB relation. If such an OpenIE triple exists, we say that the triple (s, r_{open}, o) has a *KB hit*. Note that an OpenIE triple might have more than one KB hit. For instance, the OpenIE triple *(Jeff Bezos; "is founder of"; Amazon.com)* might have two KB hits—e.g., t_1 = (Amazon.com; dbo:foundedBy; Jeff Bezos) and t_2 = (Amazon.com; dbo:ceo; Jeff Bezos)—, whereas the OpenIE triple *is aligned with* t_1 and with t_2. As shown in this example, r_{open} is mapped to two KB relations, i.e. r_{open} is a mention of two KB relations: dbo:foundedBy and dbo:ceo. Thus, if r_{open} is a mention of a KB relation r_{KB}, then r_{open} does not necessarily epxress the same information as r_{KB} (more detailed discussion is provided in Section 5.3). We can thus think of the number of KB hits as an optimistic measure of the number of OpenIE triples that

are represented in the KB (with caveats, see below): the KB contains some relation between the corresponding entities, although not necessarily the one being mentioned.

We observed that 29.7% of the OpenIE triples in OPIEC-Linked have a KB hit in either DBpedia or YAGO. More specifically, 25.5% of the triples have a KB hit in DBpedia, 20.8% in YAGO, and 16.6% in both DBpedia and YAGO. Most of these triples have exactly one hit in the corresponding KB. Consequently, 70.3% of the linked triples do not have a KB hit (we analyze these triples in Section 5.2.3). This observation shows that most of the OpenIE triples contain knowledge which is not present in the KBs that were constructed from the same resource (Wikipedia). Part of these triples contain noise and need post-processing steps to make them useful for KBs. For the OpenIE triples that are not noisy, we found that most of them are relevant for such KBs—for DBpedia in particular—and they could be used for extending them (see Section 5.4 for details).

Table 5.1 shows the most frequent open relations mapped to the the DBpedia relations dbo:location, dbo:associatedMusicalArtist, and dbo:spouse. The frequencies correspond to the number of OpenIE triples that (1) have the specified open relation (e.g., *"be wife of"*) and (2) have a KB hit with the specified KB relation (e.g., dbo:spouse). There is clearly no 1:1 correspondence between open relations and KB relations. On the one hand, open relations can be highly ambiguous (e.g., the open relation *"be"* has hits to the KB relations dbo:location and dbo:associatedMusicalArtits). On the other hand, open relations can also be more specific than KB relations (e.g., the open relation *"be guitarist of"* is more specific than the KB relation dbo:associatedMusicalArtist) or semantically different than the KB relations they align to (e.g., the open relation *"be widow of"* and the KB relation dbo:spouse).

To gain more insight into the type of triples contained in OPIEC-Clean, we selected the top-100 most frequent open relations for further analysis. These relations constitute roughly 38% of the OPIEC-Clean corpus, which shows that the relation frequencies are highly skewed. We then used OPIEC-Linked as a proxy for the number of DBpedia hits of these relations. The results are summarized in Table 5.2 as well as in Appendix B. The open relation *"have"*, for example, is mapped to 330 distinct DBpedia relations, the most frequent ones being dbo:author, dbo:director and dbo:writer. Generally, the fraction of KB hits (from OPIEC-Linked) is quite low, averaging at 16.8% for the top-100 relations. This indicates that there is a substantial amount of information present in OpenIE triples that is not present in reference KBs which are constructed from the same resource as the OpenIE corpus. Moreover, about 42 distinct KB relations align on average with each open relation, which again indicates that open relations should not be directly mapped to KB relations.

By far the most frequent open relations in OPIEC-Clean are *"be"* and *"have"*, which constitute 21.1% and 6.1% of all the triples, respectively. These open relations are also the

Open relation	Frequency in OPIEC-Clean	Frequency in OPIEC-Link	# KB hits	# distinct KB rel.s	Top-3 mapped DBpedia rel. and hit frequency	
"be"	21,911,174	1,475,332	173,107 (11.7%)	410	dbo:type dbo:occupation dbo:isPartOf	72,077 12,508 8,012
"have"	6,369,086	216,332	137,865 (63.7%)	330	dbo:author dbo:director dbo:writer	14,056 10,416 9,765
"be in"	3,219,301	1,150,667	804,378 (69.9%)	225	dbo:country dbo:isPartOf dbo:state	287,557 222,175 64,675
"include"	487,899	14,746	1,573 (10.7%)	128	dbo:type dbo:associatedBand dbo:associatedMusicalArtist	380 83 83
"be bear in"	289,947	7,138	1,477 (20.7%)	30	dbo:birthPlace dbo:isPartOf dbo:deathPlace	1,147 73 62
"win"	236,169	8,819	910 (10.3%)	54	dbo:award dbo:race dbo:team	299 210 50
"be know as"	215,809	7,993	675 (8.4%)	123	dbo:location dbo:associatedBand dbo:associatedMusicalArtist	46 42 42
"become"	213,807	5,123	393 (7.7%)	90	dbo:successor dbo:associatedBand dbo:associatedMusicalArtist	63 33 33
"have be"	191,140	1,855	101 (5.4%)	32	dbo:type dbo:position dbo:leader	12 9 7
"play"	163,643	4,842	835 (17.2%)	54	dbo:portrayer dbo:author dbo:instrument	367 101 76
"be know"	157,751	351	51 (14.5%)	15	dbo:occupation dbo:family dbo:country	20 12 3
"die in"	146,681	638	127 (19.9%)	20	dbo:deathPlace dbo:battle dbo:commander	71 13 9
"join"	134,159	2,656	903 (34.0%)	65	dbo:team dbo:associatedBand dbo:associatedMusicalArtist	301 105 105

Table 5.2 The most frequent open relations in OPIEC-Clean, along with DBpedia mapping information from OPIEC-Link (continued in Tab. 6, Appendix B)

most ambiguous ones in that they are mapped to 410 and 330 different DBpedia relations, respectively. Here the open relations are far more "generic" than the KB relations that they are mapped to. This is illustrated in the examples shown on Table 5.3. Note that in these cases, *"have"* refers to the possessive (e.g., *"Odbbins' Wine"*).

("Claudia Hiersche"; "be"; "actress")	$\xrightarrow{DBpedia}$	(Claudia Hiersche; dbo:occupation; Actress)
("Warren Harding"; "be"; "Republican")	$\xrightarrow{DBpedia}$	(Warren G. Harding; dbo:party; Rep. Party (U. S.))
("Cole Porter"; "have"; "Can-Can")	$\xrightarrow{DBpedia}$	(Can-Can_(musical); dbo:musicBy; Cole_Porter)
	$\xrightarrow{DBpedia}$	(Can-Can (musical); dbo:lyrics; Cole Porter)
("Oddbins"; "have"; "Wine")	$\xrightarrow{DBpedia}$	(Oddbins; dbo:product; Wine)

Table 5.3 Example of alignments of OpenIE triples with the open relations *"be"* and *"have"*

5.2.2 Spatio-Temporal Facts

We also investigated to what extent the space and time annotations in OpenIE triples relate to the corresponding space and time annotations in YAGO. In particular, YAGO provides:

- *YAGO date facts*, which have entities as subjects and dates as objects:
  e.g., (Keith Joseph, wasBornOnDate, 1918-01-17).
- *YAGO meta-facts*, which are spatial or temporal information about other YAGO facts:
  e.g., (Steven Lennon, playsFor, Sandnes Ulf) has meta-fact (occursUntil, 2014).

Note that date facts roughly correspond to temporal reference annotations in OPIEC, whereas meta-facts correspond to spatial or temporal triple annotations (for more details, refer to Section 3.7).

To compare OPIEC with YAGO date facts, we selected all triples with (i) a disambiguated subject and (ii) an object that is annotated as date from OPIEC. There are 645,525 such triples. As before, we align these triples to YAGO using an optimistic notion of a KB hit. In particular, a *KB date hit* for an OpenIE date fact $(s, r_{\text{open}}, d_{\text{open}})$ is any KB fact of the form $(s, r_{\text{KB}}, d_{\text{KB}})$, i.e., we require that there is temporal information but ignore whether or not it matches. We use this optimistic notion of KB date hit to avoid disambiguating the open relation or date. Even with this optimistic notion, we observed that only 36,262 (5.6%) of the OpenIE date facts have a KB date hit in the YAGO date facts.

We also compared the spatial and temporal annotations of OPIEC-Linked with the YAGO meta-facts. We found that roughly 13,203 OPIEC-Linked triples have a KB hit with a YAGO

triple that also has an associated a meta-fact. Out of these linked OpenIE triples, 2,613 are temporal and 2,629 are spatial.

To provide further insight, we analyzed the spatial-temporal annotations of OPIEC more closely. We identified two major reasons why spatio-temporal information of OPIEC triples is not found in YAGO:

(i) the information is missing from the KB

(ii) the information is available in the KB, but only indirectly

For an example of missing information, consider the OPIEC-Linked triple *(Iain Duncan Smith, "is leader of", Conservative Party)* with temporal annotations *(pred="from", 2001)* and *(pred="to", 2003)*. YAGO contains the KB hit (Iain Duncan Smith; isAffiliatedTo; Conservative Party (UK)). Note that the YAGO relation is less specific than the open relation, and that no temporal information is present. As another example, consider the OpenIE triple *(Neue Nationalgalerie; "be built by"; Ludwig Mises van der Rohe)* with temporal annotation *(pred="in", 1968)*. Again, the YAGO hit (Neue Nationalgalerie; linksTo; Ludwig Mies van der Rohe) is less specific than the OpenIE triple and it lacks temporal information. On the other hand, YAGO does contain the triple (Neue Nationalgalerie; hasLongitude; 13.37) with temporal meta-fact 1968-01-01. Here the temporal information is present in the KB, but only indirectly and for a different relation.

Generally, the low number of KB hits indicates that a wealth of additional spatial and/or temporal information is present in OpenIE data, and that the spatial/temporal annotations provided in OPIEC are potentially very valuable for automated KB completion tasks, such as precise temporal slot filling [Wang et al. 2019b; Wang and Jiang 2020].

5.2.3 Non-Aligned OpenIE Triples

We found that more than half of the triples in OPIEC-Linked that do not have a KB hit refer to one of the top-100 most frequent relations in OPIEC-Clean. Since OPIEC-Clean is much larger than OPIEC-Linked, this indicates that it contains many facts that are not present in DBpedia. Naturally, not all of these facts are correctly extracted, though, and disambiguation is indeed a major challenge [Galárraga et al. 2014]. In particular, we took a random sample of 100 non-aligned triples from OPIEC-Linked and manually labeled each triple as *correctly extracted* or *incorrectly extracted*. 60% of the triples were considered to be correctly extracted. In another sample of 100 high-confidence triples (whereas the confidence score was > 0.5), 80% were correctly extracted. This shows the potential and the challenges of harnessing the knowledge contained within the OpenIE triples (for more detailed study on this claim, refer to Section 5.4).

5.3 Analysis of OPIEC Triples and DBpedia Facts with Same Arguments

In this section, we study the semantics of alignments between OPIEC triples and DBpedia facts with the same arguments. Such alignments are inspired by the Distant Supervision Assumption (DSA), which is originally used for traditional information extraction tasks [Mintz et al. 2009]. The DSA states that if there is a KB fact and a sentence mentioning the entity pair of the KB fact, then that sentence expresses the information contained in the KB fact. Similarly, the DSA within the OpenIE context states that if there is an OpenIE triple for which there is a KB fact having the same arguments (i.e. the OpenIE triple has a KB hit), then the OpenIE triple expresses the information of the KB fact. Note that the notion of KB-hit is similar. In particular, an OpenIE triple is considered to have a KB-hit if there exists an entry in the knowledge base, such that the OpenIE triple and the KB fact have the same argument pair. The KB-hit, however, does not make assumptions as to *how* the OpenIE triple and the KB fact are semantically related. The DSA, on the other hand, goes further and makes assumptions about semantic relatedness (namely, that the OpenIE triple and the KB fact have equivalent semantics).

The DSA is the key assumption used for bootstrapping an OpenIE extractor [Wu and Weld 2010; Pal and Mausam 2016; Gotti and Langlais 2019]. Some methods for constructing OpenIE systems bootstrap a training set via the DSA, which is subsequently used either for learning OpenIE extraction rules [Wu and Weld 2010] or learning a neural model for extracting OpenIE triples [Cui et al. 2018]. Lockard et al. [2019] use the DSA in similar manner, though instead of a reference KB, they exploit the DOM nodes from web pages. In the first attempt to bootstrap an OpenIE extractor, Wu and Weld [2010] used Wikipedia infoboxes (via DBpedia) as a source for distant supervision: if there is a sentence in Wikipedia containing an entity pair and a corresponding DBpedia entry having the same entity pairs, then they store the syntactic patterns between the two entities (e.g. the shortest path in the dependency parse tree between the two target entities). Then, these syntactic patterns are used for learning OpenIE extraction rules. The underlying assumption is that the KB relation and the instance of the syntactic pattern (i.e. the *open* relation) express the same information. Other OpenIE systems exploit DSA in similar spirit, including OLLIE [Mausam et al. 2012], ReNoun, [Yahya et al. 2014], NestIE [Bhutani et al. 2016], BONIE [Saha et al. 2017] and IMoJIE [Kolluru et al. 2020].

Though the DSA is mainly used for bootstrapping OpenIE extractors or generating training data for training neural models for OpenIE, its limits and validity were not thoroughly studied. Because such analysis involves comparing semantics of information in a manner

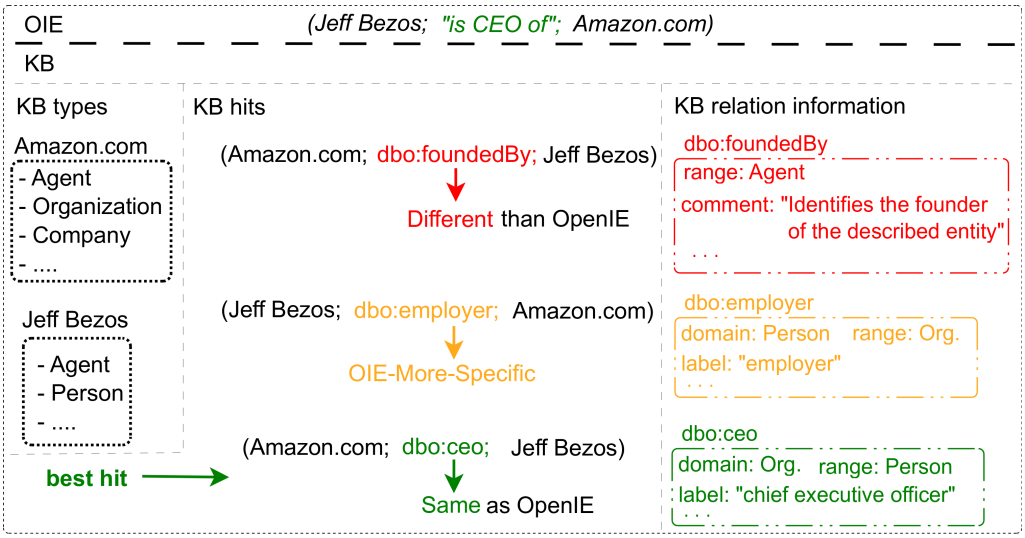


Fig. 5.1 Hit categories indicate semantic relatedness b/w OpenIE triple and its KB hits

that a human would interpret, the study needs to be done manually [Gashteovski et al. 2020]. In this section, we provide manual study of OPIEC triples and DBpedia facts with same arguments.

5.3.1 KB Hits

The concept of KB hit is closely related with the concept of DSA for OpenIE. As discussed in Section 5.2, a single KB hit indicates an OpenIE triple for which a KB fact exists. Note that an OpenIE triple may have one or more KB hits. For instance, in Figure 5.1, the OpenIE triple *(Jeff Bezos; "is CEO of"; Amazon.com)* has three KB hits. A KB hit, however, says nothing about *how* the OpenIE triple and the KB fact are semantically related. The DSA goes a step further and indicates semantic relatedness: if there is an OpenIE triple with a KB hit, then the OpenIE triple expresses the information of the KB fact. We study the semantic relatedness between an OpenIE triple and its KB hit using four *hit categories: Same, OIE-More-Specific, KB-More-Specific* and *Different*.

Same: OpenIE triple and KB fact are semantically equivalent, i.e. they express the same information. On Figure 5.1, the OpenIE triple *(Jeff Bezos; "is CEO of"; Amazon.com)* expresses the same information as the KB fact (Amazon.com; dbo:ceo; Jeff Bezos).

OIE-More-Specific: OpenIE triple is semantically more specific than the KB fact, i.e. it expresses the KB fact along with additional information not present in the KB fact. On Figure 5.1, the OpenIE triple is more specific than the KB hit (Jeff Bezos; dbo:employer; Amazon.com), because the OpenIE triple implies the KB fact and additionally expresses that Jeff Bezos is a CEO.

KB-More-Specific: KB fact is semantically more specific than the OpenIE triple, i.e. it expresses the OpenIE triple along with additional information not present in the OpenIE triple. Consider the OpenIE triple *(Angela Merkel; "is politician from"; Germany)* and the KB hit (Angela Merkel; dbo:chancellor; Germany). The KB fact is more specific, because it implies the OpenIE triple and additionally expresses that Angela Merkel is a chancellor. Contrary to *OIE-More-Specific*, KB relations in such cases cannot be inferred from the OpenIE triple.

Different: OpenIE triple is semantically different than the KB fact, i.e. it expresses conceptually different information than the KB fact. Such KB hits cannot be compared in terms of more-general or more-specific relatedness. On Figure 5.1, the KB hit (Amazon.com; dbo:foundedBy; Jeff Bezos) expresses different information w.r.t. the OpenIE triple, because *CEO* and *founder* are two different concepts which cannot be compared in terms of specificity.

In case there are several KB hits for one OpenIE triple, each KB hit is assigned a separate category (see example on Figure 5.1). We assign only one label —*best hit*— describing the best possible semantic relatedness of the OpenIE triple w.r.t. all KB hits (e.g. on Figure 5.1, the best hit is *Same*). In particular, we prioritize the KB hits that allow for the KB triple to be inferred by the OpenIE triple as precisely as possible. Thus, for the best hit, the order of preference of the hit categories is *Same* (OpenIE triple expresses the same information as the KB fact), *OIE-More-Specific* (KB fact can be inferred by the OpenIE triple, though the OpenIE triple is more specific), *KB-More-Specific* (KB fact cannot be inferred by the OpenIE triple, but the inverse is true) and finally *Different*. In our study, we consider the best hits only, because we are interested in the best possible alignment of OpenIE extractions with KB facts.

5.3.2 Study Design

The goal of the study is to *investigate the limits* of the distant supervision assumption between OPIEC triples and DBpedia facts having the same arguments. We study this by investigating what can be achieved if: (1) the OpenIE arguments are correctly disambiguated; (2) OpenIE triples are correctly extracted; (3) OpenIE relations are semantically disambiguated. To this end, we constructed a suitable corpus. We used a subset of OPIEC-Linked, which is the largest publicly-available OpenIE corpus to date, having 6M OpenIE triples with arguments which contain golden disambiguated links. Since we investigate only OpenIE triples which are correctly extracted, we filtered out triples from OPIEC-Linked that we found to be noisy, which left us with approximately 3M triples.[3] Because we need KB triples

[3]In the remainder of the chapter, we refer to this dataset as OPIEC for simplicity

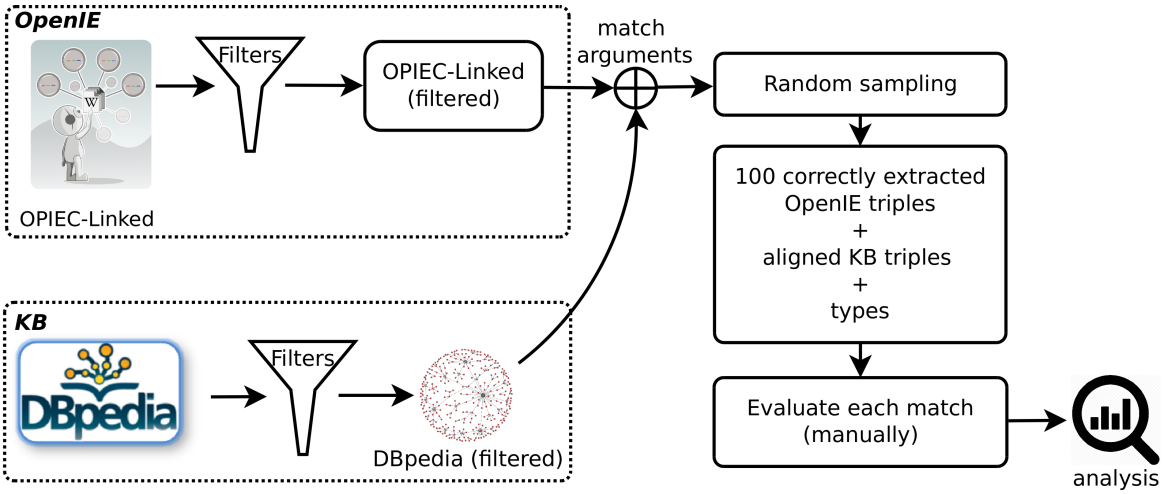


Fig. 5.2 Analysis of OPIEC triples and DBpedia facts with same arguments: study design

with disambiguated arguments on both sides (subject and object), for this particular study we filtered out triples containing literals, abstracts, dates, etc.[4] We discuss the methodologies for construction and further details about the OpenIE data and the KB data in Appendix C.1 and Appendix C.2 respectively.

Next, we constructed a random sample of 100 correctly extracted OpenIE triples from OPIEC which also have KB hits in DBpedia. We show to a human annotator the OpenIE triple and the relevant KB hit information: 1) KB hits: every possible KB hit; 2) KB types: to assure the labeler that the types of the OpenIE triple's arguments match the domain/range constraints of the KB relation counterpart; 3) KB relation information: domain, range, description, etc., to help the labeler fully understand the exact semantics of the KB relation. Each KB hit of the OpenIE triple was labeled with one of the four possible hit categories. For each OpenIE triple, we keep the label of the best hit.

Finally, we split the OPIEC data into two subsets: *All relations* and *Is-a relation*. Both of these subsets are studied separately. The reason for such split is because we have a substantial amount of triples having the *Is-a relation* form *(subject; "be"; object)*. Such triples express types — e.g. *(Berlin; "be"; City)* — and we treat them differently, because we want to investigate how the type information extracted from OpenIE compares with the current KB information. The subset *All relations* are all OPIEC triples except the triples having *Is-a relation*. Both of the sub-studies follow the procedure explained in the previous two paragraphs (also, illustrated on Figure 5.2).

[4] In this chapter we refer to the filtered DBpedia dataset as DBpedia for simplicity

5.3.3 Experimental Results and Discussion

All-relations

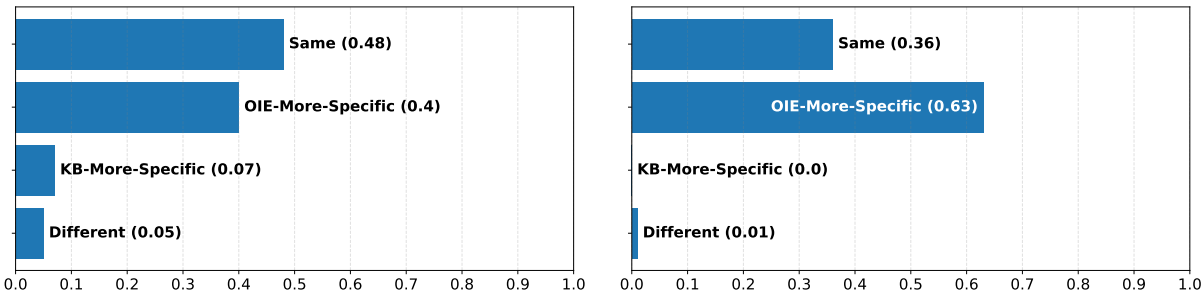
We observed that in 88% of the cases, the OpenIE triple from OPIEC is able to semantically express its best hit KB fact from DBpedia (Figure 5.3a). However, in almost half of these cases (40% of all the triples) the OpenIE triple is more specific, meaning that it expresses the information contained in the KB fact along with additional information. Consider the OpenIE triple: *(All We Grow; "be debut album of"; S. Carey)* and its KB hit (All We Grow; dbo:artist; S. Carey). In this example, the OpenIE triple can express the information of the KB hit fact, though it also contains additional information about the album (namely, that the album is a *debut* album). In 12% of the cases, however, the OpenIE triple is not able to express its best hit. These are the cases when either the KB triple is more specific — which means that the KB triple cannot be inferred by the OpenIE triple — or the semantics of the OpenIE triple is entirely different than the semantics of the KB fact. More precisely, in 7% of the cases the OpenIE triple is more generic than its KB hit. For example, we have the OpenIE triple *(Rhacophorus annamensis; "be species of"; Frog)* and the KB hit (Rhacophorus annamensis; dbo:order; Frog). Judging from the OpenIE triple only, it is not enough to infer the relation between the two entities (could be order, genus, family, kingdom, etc.). Finally, 5% of the triples have entirely different semantics than their KB hit (e.g. *(Saab Automobile; "test V8 in"; Saab 99)* v.s. (Saab 99; dbo:manufacturer; Saab automobile)).

Is-a relation

We observed that the OpenIE triples with *Is-a* relation are more specific than the DBpedia types in roughly 2/3 of the cases (Figure 5.3b). In only 1/3 of the OpenIE triples, the KB contains an equivalent type. There are almost no cases where either the OpenIE type was more generic than the type found in the KB, nor when they are different. This suggests that the OpenIE triples having *Is-a relation* can provide more fine-grained types for the KB. For example, the OpenIE triple *(Tony Blair; "be"; Prime minister)* is more fine-grained than the DBpedia type (Tony Blair; type; OfficeHolder). From the type "Prime minister" one can infer the type "OfficeHolder", but not the other way around.

5.3.4 Qualitative Study

We found multiple reasons for cases where the OpenIE triple from OPIEC is more specific than its KB hit in DBpedia. Sometimes, the details in the relation refer to more fine grained types for the argument(s). Consider the OpenIE triple *(Strul; "is Swedish film directed by";*



(a) *All relations* (b) *Is-a relation*

Fig. 5.3 Semantic relatedness between OpenIE triples from OPIEC and their DBpedia hits

Jonas Frick). The KB hit of this triple is (Strul; dbo:director; Jonas Frick) and the type available for Strul in DBpedia is "film". If there was a type "Swedish film" in DBpedia, then this alignment would have been equivalent. The above mentioned example, however, can be fully expressed with a KB formula. Consider the candidate formula:

$$\text{(Strul; dbo:director; Jonas Frick)} \wedge \text{(Strul; dbo:origin; Sweden)},$$

which perfectly captures the information shown in the OpenIE triple. For the cases where the OpenIE triple represents different information than the KB hit, we found that usually the information on both sides is somehow semantically related. For example, we might have the OpenIE triple *(subject; "be CEO of"; object)* and its KB hit (subject; dbo:founder; object). In this example, *CEO* and *founder* are related concepts, though they are semantically different.

5.4 Expressibility of OPIEC triples with DBpedia

The study in Section 5.3 was about investigating the limits of aligning OpenIE extractions from OPIEC for which KB facts exist in DBpedia. Such cases, however, comprise only a small fraction of the data (more precisely, 25% of the data[5]). In this section, we study all cases: the limits of aligning *any* OPIEC triple with a *single* DBpedia fact. Our goal is to answer the question of whether any OPIEC triple contains information which is relevant for DBpedia and, if so, *how* can it be expressed with the KB. We measure relevance by quantifying the information found in the OpenIE triples that can be expressed with KB language and we study how can such information be expressed (i.e., to what extent the

[5]High confidence subset of OPIEC-Linked (the same dataset used in the study presented in Section 5.3). Details about the data are discussed in Appendix C, Section C.1.

OpenIE triples can be expressed with a single KB fact, multiple KB facts or with the use of KB formulas). Finally, we measure how much of the OpenIE information that is relevant for the KB is actually present in the KB. We found that most of the information found in the OPIEC triples is either not fully present or not present at all in DBpedia, suggesting that further post-processing of such OpenIE triples have the potential of extending reference KBs.

5.4.1 One Triple Assumption

Many methods use large-scale outputs of OpenIE systems for downstream tasks by trying to express *one* OpenIE triple with *one* KB fact. This includes mapping open relations to a KB relation in order to improve a slot filling task [Soderland et al. 2013; 2015a; Angeli et al. 2015] or to unify open relations into a single KB schema [Bovi et al. 2015], canonicalize open relations by clustering them in relational synsets which are then mapped to a KB relation [Galárraga et al. 2014], mapping open relations to lexical KBs such as WordNet [Grycner and Weikum 2014], and mapping OpenIE triples to KB facts [Soderland et al. 2010; Zhang et al. 2019; Putri et al. 2019] which are used for downstream tasks such as KB population [Soderland et al. 2013; Dutta et al. 2013; 2015] and slot filling [Yu et al. 2017]. Such methods implicitly make the *One Triple Assumption (OTA): "Any OpenIE triple can be expressed with one KB fact"*. For example, the OpenIE triple *(Emmanuel Macron; "be president of"; France)* can be expressed with the KB relation dbo:president: (Emmanuel Macron; dbo:president; France). Note that such mapping is possible even if this particular instance does not exist in the KB (e.g. in DBpedia, there is no KB fact stating that Emmanuel Macron is the president of France).

Sometimes, an OpenIE triple cannot be expressed by a single KB fact, though it can be expressed by multiple KB facts or a first-order logic KB formula. Consider the OpenIE triple *(John F. Kennedy; "be grandchild of"; P. J. Kennedy)*. This triple can be represented with the following KB formula:

$$\exists x : \text{(John F. Kennedy; dbo:parent; x)} \wedge \text{(x; dbo:parent; P. J. Kennedy)},$$

because there is no DBpedia relation expressing "grandchild" relationship between two entities. In this section, we also study to what extent multiple KB triples or the use of KB formulas can help improve the expressibility of the OpenIE triples with KB language when one KB triple is not enough. Such cases are important to study, because they have the potential of improving the usefulness of OpenIE data for KB-related tasks. For example, one might use multi-hop reasoning of OpenIE triples in order to infer new relations between entities in a KB. In similar spirit, Das et al. [2016] use multi-hop reasoning between two

entities in a KB to infer new relations for automated KB construction, while Fu et al. [2019] do multi-hop reasoning over OpenIE data.

5.4.2 Expressibility Levels

In order to understand the semantic expressibility of an OpenIE triple w.r.t. KB facts, we differentiate three possible expressibility levels: *Fully-Expressible, Partly-Expressible* or *Not-Expressible*.

Fully-Expressible: The semantics of an OpenIE triple can be completely expressed with one KB fact. Consider the OpenIE triple *(Eric Schmidt; "be chairman of"; Google)*. This triple is *Fully-Expressible* because the semantic content of the triple can be fully expressed with the KB fact (Google; dbo:chairman; Eric Schmidt).

Partly-Expressible: The semantics of an OpenIE triple can be partly expressed with one KB fact, i.e. the OpenIE triple contains additional information which is not present in the KB fact. For example, the OpenIE triple *(Steffi Graf; "defeated"; Natasha Zvereva)* is *Partly-Expressible*, because there is no KB relation about one athlete defeating another. The OpenIE triple, however, can be partly expressed with the KB fact (Steffi Graf; dbo:opponent; Natasha Zvereva). Note that in such cases, the KB triple can be inferred from the OpenIE triple. In our example, from the OpenIE triple we know that one athlete defeated another, which implies that that the two athletes are opponents (this can be expressed with the KB relation dbo:opponent).

Not-Expressible: The semantics of an OpenIE triple cannot be expressed with one KB fact, i.e. it is neither *Fully-Expressible* nor *Partly-Expressible*. For example, the OpenIE triple *(IBM; "has Color Paint for"; IBM PCjr)* cannot be expressed with KB fact, because the KB is not capable of expressing such information in one fact.

We make use of the above-defined expressibility levels to understand the semantic expressibility of an OpenIE triple w.r.t. KB formulas as well. For example, the above mentioned OpenIE triple *(IBM; "has Color Paint for"; IBM PCjr)* is *Not-Expressible* w.r.t. a single KB fact, but it is *Fully-Expressible* w.r.t. KB formulas, because we can represent that particular OpenIE triple with the following KB formula (which is a conjunction of two KB facts):

(IBM; dbo:product; Color Paint) $\wedge$ (Color Paint; dbo:computingPlatform; IBM PCjr).

5.4.3 Study Design

The goal of the study is to investigate whether the information found in any OPIEC triple is relevant for DBpedia (findings are discussed in Section 5.4.3). Such study is important for

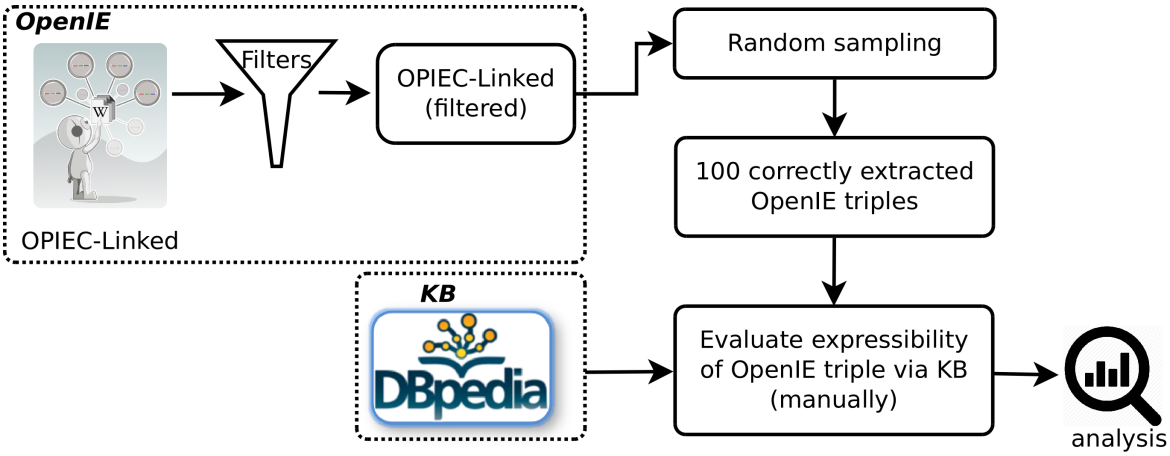


Fig. 5.4 Expressibility of OPIEC triples with DBpedia: study design

quantifying the relevance of OpenIE triples w.r.t. a KB: if the OpenIE triples are relevant for the KB, then the knowledge of the OpenIE triples can be harnessed further, which is helpful for improving downstream KB related tasks, such as slot filling [Yu et al. 2017]. Moreover, it is important to understand whether the information found in OpenIE triples—which is relevant for the KB—is present in the reference KB. While information of OpenIE triples that is also present in a reference KB is relevant for tasks such as knowledge fusion [Dong et al. 2014], OpenIE information which is not present in a reference KB is important for tasks such as KB extension [Dutta et al. 2015] or link prediction [Gupta et al. 2019]. In Section 5.4.5 we discuss how much of the OPIEC information is new w.r.t. the reference KB (DBpedia).

We study the relevance of OpenIE triples w.r.t. reference KB by measuring the amount of OpenIE information which can be expressed with KB language. We also study the different levels of expressibility. We constructed a random sample of 100 correctly extracted OpenIE triples from OPIEC and an expert labeler evaluated the expressibility level for each OpenIE triple w.r.t. a single DBpedia fact as well as w.r.t. a KB formula (Figure 5.4). First, we measure in how many cases an OPIEC triple can be expressed with *one* DBpedia fact fully or partially. Then, when the OPIEC triple is *Partly-Expressible* (or *Not-Expressible*), we investigate whether it can become *Fully-Expressible* (or *Partly-Expressible/Fully-Expressible*) with the use of a KB formula or multiple KB facts. Next, we measure how much of this OpenIE information (labeled for expressibility w.r.t. DBpedia) is present in DBpedia. We do this by manually checking if every OpenIE triple is present in DBpedia in some form (i.e., is it *Fully-Present, Partially-Present* or *Not-Present* in DBpedia; details are discussed in Section 5.4.5).

"be chairman of"		*"be natural son of"*		*"be municipality in"*		*"be released by"*	
dbo:keyPerson	(155)	dbo:relation	(4)	dbo:isPartOf	(2,270)	dbo:recordLabel	(660)
dbo:occupation	(96)	dbo:predecessor	(1)	dbo:district	(446)	dbo:distributor	(69)
dbo:party	(94)	dbo:parent	(1)	dbo:country	(100)	dbo:artist	(48)
dbo:foundedBy	(43)	/		dbo:federalState	(10)	dbo:publisher	(47)
dbo:chairman	(40)			dbo:province	(7)	dbo:developer	(33)
dbo:knownFor	(34)			dbo:region	(7)	dbo:product	(31)
...				...		...	

Table 5.4 Examples of hit relation counts for several open relations

Expressibility of OPIEC Triple with a Single DBpedia Fact

Each OpenIE triple is presented to a human annotator along with: 1) argument types from DBpedia of the OpenIE triple; 2) a list of candidate DBpedia relations; 3) relevant information about the candidate DBpedia relations (descriptions, domain/range types, ...); 4) all other relevant information from DBpedia. Then, the annotator was given the following question:

"Can the OpenIE triple be expressed with one KB fact?"

Given all the KB information available, the human annotator then assigned one of the three possible labels: *Fully-Expressible, Partly-Expressible* and *Not-Expressible*. Note that the assumption here is that we have a *perfect mapping* from OpenIE triple to KB Fact. Thus, the labeler assigns the best possible mapping as a final label. The goal is to study — given a perfect possible mapping — the expressibility of an OpenIE triple via KB fact.

The list of candidate KB relations was generated by two methods: *KB hit counts* (by aggregating *hit relations*) and *argument type constraints* (by using any DBpedia relation satisfying the type constraints of the OpenIE arguments).

Hit relation. When possible, we aligned every OPIEC triple to DBpedia via KB hit statistics. In a previous step, for every open relation, we counted the corresponding KB relation which we got from the KB hit. These counts were sorted in descending order. Table 5.4 shows several examples of hit relations. Consider the example for the open relation *"be chairman of"*. For each OpenIE triple having the open relation *"be chairman of"*, there were 155 KB hits in DBpedia with the KB relation dbo:keyPerson, 96 KB hits with the KB relation dbo:occupation and so on.

Any relation. For aligning the OpenIE triples to KB facts, it is important that we go beyond the KB hits statistics, because such statistically-based methods are useful only for open relations which appear frequently in the OpenIE corpus. To this end, we generate more candidates by selecting only the DBpedia relations that fit the domain/range constraints imposed by the argument types of the OPIEC triple. In case the DBpedia types themselves for

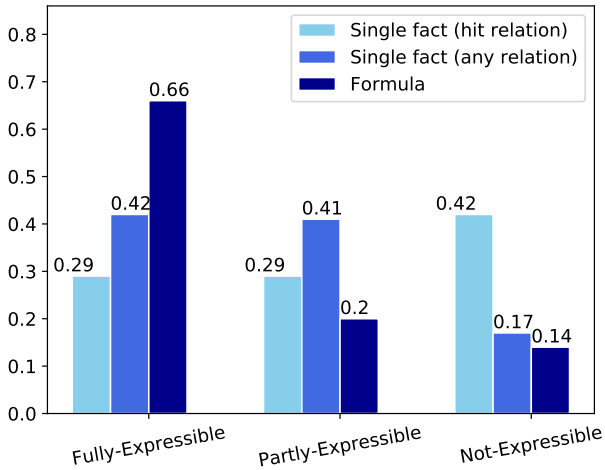


Fig. 5.5 Expressibility of OpenIE information with KB: Can an OPIEC triple be expressed in DBPedia?

the OpenIE arguments are wrong or missing, the labeler corrects them with the appropriate DBpedia type. With this strategy, we ensure that we show every possible fitting candidate to the labeler.

Expressibility of OPIEC Triple with KB Formula

Since for many cases OPIEC triples cannot be fully expressed with a single DBpedia fact, an expert labeler manually generated KB formulas (when possible) which switch the expressibility level from *Partly-Expressible* to *Fully-Expressible* or from *Not-Expressible* to *Fully/Partly-Expressible*. For example, the OpenIE triple *(Garrett Davis; "is Representative from"; Kentucky)* is *Partly-Expressible* with the DBpedia fact (Garret Davis; dbo:region; Kentucky), and it is fully expressible with the KB formula:

(Garrett Davis; dbo:profession; State representative) $\wedge$ (Garrett Davis; dbo:state; Kentucky).

5.4.4 Expressibility of OPIEC with DBpedia: Results and Discussion

Only 29% of the OPIEC triples can be fully expressed with a single DBpedia fact using one of the hit relation candidates (light-blue bars on Figure 5.5); another 29% of the cases can be only partly expressed and 42% of the OPIEC triples cannot be expressed at all. While this observation suggests that the KB hit counts do contain signals for KB expressibility, it is not enough to express all the OPIEC triples. The main reason is because KB hit counts work good only for the triples having open relations with high frequency. Higher frequency of an

#	OpenIE triple	KB formula
t_1	Temporal annotation *(Coral Fang; "was released by"; Sire Records)* *Time: (in, 2003)*	(Coral Fang; dbo:recordLabel; Sire Records) ∧ (Coral Fang; dbo:releaseDate; 2003)
t_2	Complex formula *(Garrett Davis; "was Rep. from"; Kentucky)*	(G. D.; dbo:profession; State representative) ∧ [(G. D.; dbo:region; K.) ∨ (G. D.; dbo:state; K.)]
t_3	Existential quantification *(Franz Liszt; "wrote piece for"; Piano solo)*	$\exists x$: (F. L.; dbo:write; x) ∧ (x; dbo:genre; P. solo)
t_4	Conjunctive formula *(Dick Ket;* *"was Dutch magic realist painter noted for";* *Still life)*	(Dick Ket; dbo:nationality; Netherlands) ∧ (Dick Ket; dbo:genre; Magic realism) ∧ (Dick Ket; dbo:occupation; Painter) ∧ (Dick Ket; dbo:knownFor; Still life)

Table 5.5 Selected examples of OPIEC triples expressed with KB formulas

open relation implies higher likelihood for a KB hit, thus higher likelihood for capturing the semantic content (fully or partially) of an OpenIE triple by one of the candidates. In reality, many of the open relations extracted from large corpora, however, are not frequent enough, which is the main reason why in 42% of the OpenIE triples the candidates generated by the KB hit counts cannot express the OpenIE triple neither fully nor partially.

When we extend the limits of the candidates by including any DBpedia relation which respects the constraints of the argument types (represented as blue bars in the middle on Figure 5.5), then we significantly reduce the amount of OPIEC triples which cannot be expressed with one DBpedia fact (from 42% down to 17%). More precisely, 42% of the triples can be fully expressed and 41% can be partly expressed with one DBpedia fact. This study shows that most of the OPIEC triples are relevant for DBpedia, because more than 80% of them can be expressed with a single DBpedia fact. We observed, however, that nearly half of these cases are only partly expressible, since the OPIEC triples contain additional details which are not found in DBpedia. The reason for this is because KB relations have very strict semantics, while open relations have the expressibility of natural language.

When we introduce KB formulas, the expressibility of the OPIEC triples is significantly improved. We observed that the number of OPIEC triples that can be fully expressed with DBpedia increased from 42% to 66%. This was mostly on the expense of the cases where an OPIEC triple is partly expressible w.r.t. DBpedia fact (it reduced these cases from 41% to 20%). Less significantly, the KB formulas allowed for some of the OPIEC triples which are not expressible via DBpedia to be expressible (percentage went down from 17% to 14%). To

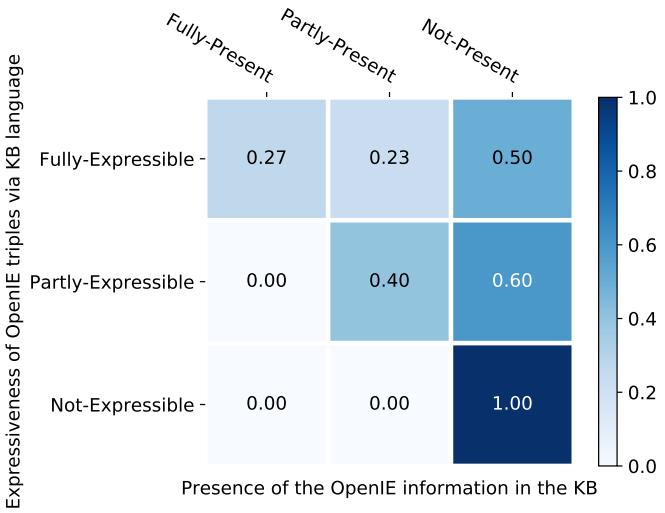


Fig. 5.6 Presence of OpenIE information with KB: does DBpedia contain the information from the OPIEC triple?

illustrate the expressibility of OpenIE triples with KB formulas more effectively, we show a set of selected examples of OPIEC triples expressed with KB formulas on Table 5.5.

Finally, 14% of the OPIEC triples cannot be expressed with DBpedia vocabulary neither with single DBpedia facts nor with KB formulas. Reasons include cases with tertiary relations or open relations which can be expressed via natural language, but are not available in DBpedia (e.g. the OpenIE triple *(Castlemaine XXXX; "sponsored"; Queensland Rugby League)* cannot be represented with DBpedia).

5.4.5 New Information for DBpedia from OPIEC

As previously discussed, much of the OpenIE information can be either fully expressed or partially expressed with KB language (Section 5.4.4). It is important to know, however, how much of such OpenIE information is new for the KB itself, because the new OpenIE information that is relevant for the KB can be subsequently used for extending the KB [Dutta et al. 2015]. Moreover, the OpenIE information that is not present in the KB and which is not relevant for the KB can be used for extending the KB schema.

To evaluate how much of the OPIEC information is new for DBpedia, we used the same OPIEC triples from the expressibility study. Based on the information content of the OpenIE triple and the information content of DBpedia, an expert labeled each OpenIE triple with one of the three possible options: 1) *Fully-Present* in the KB: there exists DBpedia fact or formula which fully expresses the OpenIE triple; 2) *Partly-Present* in the KB: there exists DBpedia fact or formula which partly expresses the OpenIE triple; 3) *Not-Present* in the KB:

there is no existing DBpedia fact or formula which can fully or partially express the OpenIE triple.

We found that in 59% of the OpenIE triples the content is not present in DBpedia at all, in 23% is partly present and only in 18% it is fully present. This observation suggests that the majority of OpenIE triples contain information which is either not present or not fully present in DBpedia. To investigate new *relevant*[6] information for DBpedia, we compared expressibility w.r.t. presence of the OpenIE information content in DBpedia (Figure 5.6). In general, we observed that most of the OpenIE information which is relevant for the KB is either not present or only partly present in DBpedia, showing the potential of such triples for improving downstream tasks such as KB population [Lin et al. 2020].

5.5 Transferability to other OpenIE Systems

In this section, we study whether and to what extent the results of the evaluations transfer to other OpenIE systems. We used three other popular OpenIE systems: Stanford OIE [Angeli et al. 2015], RnnOIE [Stanovsky et al. 2018] and OpenIE 5 [Saha and Mausam 2018]. We ran these OpenIE systems on the provenance sentences of the sampled triples used in our evaluations—explained in Section 5.3 and Section 5.4—and compared their outputs to their OPIEC counterparts (extracted with MinIE).

Consider the OPIEC triple *(Turf Buccaneers; "be album by"; Mac Dre)*. We use the provenance sentence from which this triple was extracted and we run the other OpenIE systems on the same sentence. Then, we select the triples that match the argument pair of the OPIEC triple; i.e. *(Turf Buccaneers, Mac Dre)* in the example. As before, we keep only the triples that are correctly extracted. Then, an annotator evaluates these OpenIE triples w.r.t. DBpedia for either hit category (Section 5.3) or expressibility level (Section 5.4). Finally, we compared these labels with the original labels of our evaluations for OPIEC.

In Section 5.5.3 we examine to what extent other OpenIE systems extract different entities (and entity pairs) than MinIE, given the same provenance sentences used in our study. In particular, we measure how many entities each OpenIE system extracts in general, how similar they are w.r.t. the entities extracted by MinIE, and to what extent the entities extracted by other OpenIE systems are also extracted by MinIE. Such study is important for evaluating whether other systems extract different entities, which would influence the findings of our study.

[6] An OpenIE triple is relevant for the KB if it is expressible in that KB

Label	Stanford OIE	OpenIE 5	RnnOIE	All
	OIE triples and KB facts with same args.			
Hit category	0.98	0.84	0.77	0.86
	Expressibility of OIE triples with DBpedia			
Single fact (hit relation)	0.92	0.93	0.85	0.90
Single fact (any KB relation)	0.88	0.86	0.77	0.84
KB formula	0.98	0.89	0.89	0.92

Table 5.6 Label equivalence ratio of the evaluations: labels from OPIEC triples (produced by MinIE) v.s. labels from triples produced by other OpenIE systems. *All* column considers all the labels for the triples produced by the other OpenIE systems combined.

5.5.1 Hit Categories

We compared the newly assigned labels for hit categories from the other OpenIE systems with the original labels of our evaluation presented in Section 5.3 (Table 5.6). Overall, we found that in 86% of the cases the labels were equivalent (i.e., the *label equivalence ratio*). In most cases for which there was a mismatch of the labels, the OPIEC triple has same semantics as the KB fact, while the triple by the other OpenIE system is more specific than the KB fact. Hence, when moving to other OpenIE systems, one should expect that they may produce more specific OpenIE triples.

We also observed the label equivalence ratio of the OPIEC triples w.r.t. the other OpenIE systems individually (Table 5.6). We found that RnnOIE has the lowest label equivalence ratio (77%). Again, the main reason for a mismatch of the labels is that RnnOIE extracts more specific triples than MinIE. This is because the goal of MinIE is to produce shorter extractions, while RnnOIE does not aim at reducing the length of the extractions, thus producing more specific triples. Consequently, in many cases MinIE extracts a triple having the same semantics as the KB fact, while RnnOIE extracts a more specific triple.

On the other hand, Stanford OIE produced triples that have almost the same labels as OPIEC (98% of the labels are equivalent). The reason for such observation is that Stanford OIE was constructed with the slot filling task in mind, which results in producing shorter extractions (same goal as MinIE). Therefore, the specificity levels with MinIE are similar. OpenIE 5 is in between: it produces more specific triples than Stanford OIE and less specific triples than RnnOIE.

DSA / OTA	*Entities*			*Entity pairs*		
	Count	Jaccard w.r.t. MinIE	Coverage by MinIE	Count	Jaccard w.r.t. MinIE	Coverage by MinIE
MinIE	272 / 235	1.0 / 1.0	1.0 / 1.0	221 / 169	1.0 / 1.0	1.0 / 1.0
Stanford	156 / 120	0.44 / 0.45	0.83 / 0.92	99 / 80	0.23 / 0.29	0.61 / 0.70
OpenIE 5	70 / 81	0.21 / 0.32	0.83 / 0.95	38 / 47	0.11 / 0.21	0.68 / 0.81
RnnOIE	49 / 69	0.15 / 0.27	0.84 / 0.94	27 / 41	0.07 / 0.17	0.63 / 0.73

Table 5.7 Extracted entities and entity pairs by MinIE and other OpenIE systems for both studies: DSA (Section 5.3) / OTA (Section 5.4).

5.5.2 Expressibility Levels

Following the same strategy as the labels for the hit categories, we compared the newly assigned labels for expressibility levels from other OpenIE systems with the original labels of our evaluation for expressibility of OpenIE triples with DBpedia (Section 5.4). We compared the labels for single fact (hit relation), single fact (any KB relation) and KB formula (Table 5.6).

Our findings for the expressibility levels are similar to the findings discussed in Section 5.5.1. Overall, we found that the label equivalence ratio of the OPIEC triples and the triples produced by other OpenIE systems is relatively high. Again, most mismatches of the labels are because other OpenIE systems tend to produce more specific triples. Consequently, in such cases, when MinIE extracts triple that is *Fully-Expressible*, the other OpenIE systems extract triple that is *Partly-Expressible* with the KB.

5.5.3 Extracted Entities

To compare the entities extracted by MinIE with the entities extracted by the other OpenIE systems, we used the same provenance sentences from OPIEC's triples used in our studies (Section 5.3 and Section 5.4). From them, we extracted OpenIE triples with MinIE and the other OpenIE systems. Again, we kept only the triples generated by all OpenIE systems that contain disambiguated arguments on both the subject and the object. We did not consider triples that contain more than one entity link per argument (e.g., some OpenIE systems generate whole clauses as an object, which may contain more than one entity). For such extractions, it is not clear to which entity the argument is referring to. Finally, for each entity and entity pair, we computed counts, Jaccard distance w.r.t. MinIE and coverage by MinIE (Table 5.7).

For both the DSA and the OTA sentences, we observed that MinIE extracts more arguments (and argument pairs) than the other OpenIE systems. This observation is consistent

with the findings presented in Chapter 3, where we reported high recall for MinIE. Moreover, Lin et al. [2020] reported that MinIE extracts entities that are easier to disambiguate to KBs compared to other OpenIE systems, which is another reason why the number of extracted entities is lower in other systems.

Because of the lower amount of entities extracted by the the other OpenIE systems, the Jaccard distance between the entities extracted by MinIE and other systems is relatively low. If we turn to coverage by MinIE, however, we observed that most entities extracted by the other OpenIE systems are also extracted by MinIE. This suggests that the extractions made by other OpenIE systems that are relevant for KBs were likely going to be extracted by MinIE as well. Based on these results, we conjecture that the findings of our studies—discussed in Section 5.3 and Section 5.4—largely transfer over to other OpenIE systems.

5.5.4 Discussion

Overall, we found that OpenIE triples produced by other OpenIE systems tend to have very similar hit categories (as well as expressibility levels) with the OPIEC triples. Due to the fact that MinIE—OPIEC's underlying OpenIE system—is designed to produce less specific extractions, we observed that if one uses other OpenIE systems, it should be expected the extractions to be more specific. This, in turn, results in 1) producing larger fraction of triples that are more specific than the KB triple with the same argument pair; 2) producing larger fraction of triples that are *Partially-Expressible*.

5.6 Discussion and Conclusions

In this chapter, we explored how OpenIE triples from the OPIEC corpus are related to facts of reference KBs (DBpedia and YAGO) w.r.t. information content. All resources are automatically generated from the same domain: OPIEC from the textual data, while DBpedia and YAGO from the semi-structured data of Wikipedia (infoboxes). To date, OPIEC is the largest OpenIE corpus generated from Wikipedia and DBpedia is the largest KB automatically constructed from Wikipedia.

First, we compared the content of OPIEC and reference KBs (DBpedia and YAGO), which are all constructed from the same resource (Wikipedia). Through the use of the concept of KB hits, we found that most of the OpenIE relations are ambiguous and should not be mapped directly to KB relations (i.e., an instance-level context of the OpenIE triple is needed). Moreover, most of the information found in OPIEC is not present in the reference KBs, which shows the potential of using OpenIE information for extending KBs [Dutta et al.

2015]. This part of the study, however, is fully automatic, and the findings do not explain the semantics of the alignments in details.

Second, we studied the semantic relatedness between OpenIE triples from OPIEC and DBpedia facts having the same arguments (i.e. the Distant Supervision Assumption). Such cases are important for many downstream applications as well as for constructing OpenIE systems themselves. In general, we found that, within the DSA settings, the OPIEC triples are indeed semantically related to the DBpedia facts, but quite often the OpenIE triples are more specific. This means that the OpenIE triple contains the information of the corresponding KB fact, as well as some additional information that the KB fact lacks. For example, the OpenIE triple *(Jeff Bezos; "is CEO of"; Amazon.com)* expresses the information content of the KB fact (Jeff Bezos; dbo:employer; Amazon.com), namely that Jeff Bezos is employed by Amazon.com. The OpenIE triple, however, also expresses an additional information: that Jeff Bezos is also a CEO of the company. Within the scope of this study, we also observed that OpenIE triples usually provide much more fine-grained types for KBs. For example, the OpenIE triple *(Tony Blair; "be"; Prime minister)* is more fine-grained than the DBpedia type (Tony Blair; type; OfficeHolder).

Third, we studied the expressibility of any OPIEC triple w.r.t. DBpedia: whether (and *how*) a given OPIEC triple can be expressed with a *single* DBpedia fact. For example, the OpenIE triple *(Emmanuel Macron; "be president of"; France)* can be fully expressed with the KB fact (Emmanuel Macron; dbo:president; France) even if this fact is not necessarily present in the reference KB. We call such assumption that each OpenIE triple can be expressed with a single KB fact *the One Triple Assumption (OTA)*. Such investigation is important, because OTA is the underlying assumption of many methods for aligning OpenIE triples with KB facts [Soderland et al. 2010; Nguyen et al. 2017; Zhang et al. 2019; Putri et al. 2019]. We found that, even though expressing an OPIEC triple with one KB fact is often possible, in roughly half of the cases this is only possible with some loss of information. For example, the OpenIE triple *(John F. Kennedy; "be grandchild of"; Patrick J. Kennedy)* can be partially expressed with the KB fact (John F. Kennedy; dbo:relative; Patrick J. Kennedy). Such KB expression loses some information contained in the OpenIE triple. While both the OpenIE and the KB triple suggest that John F. Kennedy and Patrick J. Kennedy are relatives, the OpenIE triple contains additional information: that John F. Kennedy is the grandchild of Patrick J. Kennedy. Such observations suggest that text is more fine-grained and thus is not limited to any KB schema. We found that the use of logical KB formulas can significantly improve the expressibility of OpenIE triples with KB vocabulary. In our previous example, the OpenIE triple can be fully expressed with the KB formula:

$$\exists x : \text{(John F. Kennedy; dbo:parent; x)} \wedge \text{(x; dbo:parent; P. J. Kennedy)}.$$

We also found that a substantial fraction of the OpenIE triples could not be expressed with the KB at all: neither with a single KB fact nor with a complex KB formula. Postprocessing of such cases could be used for updating the reference KB schema, which can be used subsequently for extending the KB itself.

Next, we observed that most of the OPIEC triples contain information which is not present in DBpedia. We found that even for the OpenIE triples which are relevant for the KB (i.e. are expressible with the KB) most of their information content is not present in DBpedia. For example, the OpenIE triple *(Emmanuel Macron; "be president of"; France)* is relevant for DBpedia, because it can be expressed with (Emmanuel Macron; dbo:president; France). Such KB fact, however, is not present in DBpedia. This shows the potential for the use of OpenIE extractions for KB-relevant downstream tasks such as automatic KB construction or KB population. One way to harvest such knowledge is to add OpenIE triples unmodified to a KB with universal schema [Riedel et al. 2013]. We did not study such scenarios and we leave such investigation for future work.

Finally, we studied to what extent our evaluations transfer to other OpenIE systems. To this end, we use triples produced by other popular OpenIE systems from the same provenance sentences of the OpenIE triples from our study in Sections 5.3 and 5.4 and comparing them with the original OPIEC triples. In particular, we selected the correctly extracted triples that have the same argument pairs as the OPIEC triples. Then, we labeled them for hit-categories (as in Section 5.3) and for expressibility w.r.t. DBpedia (as in Section 5.4). Next, we compared the labels of the new OpenIE triples with the labels of OPIEC. We found that most of the labels agree with the OPIEC labels, which suggests that the findings from our studies mostly transfer to other OpenIE systems as well. One should expect, however, to sometimes get more specific extractions by the other systems, since they were not designed with a goal to achieve compactness. This results in 1) producing larger fraction of triples that are more specific than the KB triple with the same argument pair; 2) producing larger fraction of triples that are *Partially-Expressible*. As for the entities extracted by MinIE and other OpenIE systems, we found that MinIE extracts most of the entities that are extracted by the other OpenIE systems as well as additional entities. The reason for this observation is the high recall of MinIE as well as the compact extractions made by MinIE that contribute to extracting more KB-centric entities.

A limitation of the studies presented in this chapter is that we focus on the most common form of OpenIE extractions: OpenIE triples. Some OpenIE systems extract more complex structures—e.g. nested extractions [Bhutani et al. 2016]—which are not covered by this chapter and require a separate study. As for the information content that is carried by the OpenIE triples, our evaluation suggests that most information found in the OpenIE triples is

not present in the reference KB. One way to harvest such knowledge is to add OpenIE triples to a KB with universal schema [Riedel et al. 2013]. We did not study such scenarios and leave such evaluations for future work.

Chapter 6

Conclusions and Future Work

Open Information Extraction (OpenIE) is the task of extracting arguments and their relations from natural language sentence in an unsupervised manner. A usual format of representing such information is in the form of *(“subject”; “relation”; “object”)*-triples. A common problem of such systems is that they often extract triples which are considered to be overly specific. This means that the triple contains words such that, if we remove them, the triple would not lose its meaning (e.g. removing the word *”the“* from the phrase *”the car“* does not damage the meaning of the phrase). We consider the OpenIE triple t_1 as more compact than the OpenIE triple t_2 if both t_1 and t_2 express the same information, whereas t_1 is shorter than t_2 (i.e., it contains less words than t_2).

In this thesis, we discussed methods for compact OpenIE (Chapter 3), as well as a large OpenIE corpus (OPIEC) generated from such methods (Chapter 4). OpenIE extractions are shallow in nature, which means that their representation lacks semantic rigor. On the other hand, knowledge bases provide precise meaning of their facts (which are also represented in the form of triples), which is why we analyzed the semantics of OPIEC w.r.t. reference knowledge bases (Chapter 5). OpenIE extractions are mainly used as intermediate format for representing textual information, which is used for deeper natural language understanding tasks, such as knowledge base completion, knowledge fusion or relation extraction. Achieving compactness of OpenIE extractions is important for improving the usefulness for such downstream tasks. For instance, Lin et al. [2020] showed empirically that higher compactness of OpenIE triples helps for reducing the ambiguities of the arguments for the KB completion task. Motivated from such reasons, this thesis discussed 1) methods for constructing such compact OpenIE systems; 2) large-scale corpora of compact OpenIE extractions; and 3) analysis of compact OpenIE extractions w.r.t. reference knowledge bases.

In particular, for achieving compactness in OpenIE extractions, we proposed the OpenIE system MinIE (discussed in Chapter 3). MinIE achieves compactness in two ways: through

the use of semantic annotations and through minimization (i.e., dropping words in the extractions which are considered to be overly specific). With the methods for semantic annotations, MinIE identifies certain semantic information—for factuality, attribution and quantities—within the extraction, which is then removed from the original extractions and is structured with semantic annotations. With its minimization strategy for compactness, MinIE identifies words in the extractions that are considered to be overly specific and removes them (mostly) without damaging the semantics of the extractions. Because the removal of words might damage the meaning of the extraction (e.g., a triple with argument *"data mining"* has a different meaning than a triple with argument *"mining"*), we proposed four different modes of minimization—MinIE-(C)omplete, MinIE-(S)afe, MinIE-(D)ictionary, MinIE-(A)ggressive—, which differ in the degree of aggressiveness of dropping words. Through experimental study, we found that, while the extractions get shorter as we move towards more aggressive modes of MinIE, the precision of MinIE drops as well. We also found, however, that MinIE effectively controls the compactness-precision trade-off. As a precaution for maximizing correctness, we chose MinIE-S for constructing the OpenIE corpus OPIEC (Chapter 4), though more aggressive modes of MinIE can override OPIEC's extractions if needed.

In principle, the methods proposed in MinIE can be applied to other OpenIE systems in order to make their extractions more compact. For the semantic annotations, MinIE uses simple rules that leverage the linguistic structure of the input sentence (e.g. dependency parse tree and POS tags) and a small set of domain-independent words that indicate a certain semantic annotation (e.g., the word *not* indicates negative polarity). As for the scope of semantic annotations, we showed that MinIE can be extended to include other semantic annotations, such as space and time. Moreover, the scope of the semantic annotations can be extended for extracting information from domain-specific text, such as the bio-medical or scientific domain [Lauscher et al. 2019]. In other work, OpenIE systems were tuned for extracting information from specific domains—such as the legal [Siragusa et al. 2018] or the biomedical domain [Wang et al. 2018]—, and we believe that extending the scope of the semantic annotations w.r.t. the application domain is a promising future direction of research.

The methods for minimization could also be integrated into other OpenIE systems in principle. In similar manner as with the semantic annotations, the minimization methods use a set of simple rules which exploit the linguistic structure of the sentence (e.g., dependency parsing tree and POS tags) as well as some semantic information (e.g. NER tags). We believe that a promising direction for future work is to go beyond such simple rules and to learn a neural model for minimization. The lack of training data, however, will be a large problem. We believe that possible solutions include educating crowd-workers to annotate

large amounts of such data or to use effectively methods for data-augmentation, whereas the labels from our study in Chapter 3 could be used as seed data.

OpenIE systems are useful when they are executed on large corpora, thus resulting in large OpenIE corpora. Such large OpenIE corpora are used for many downstream tasks, such as question answering, automated knowledge base construction, relation extraction and textual entailment. For these reasons, we published OPIEC—the largest OpenIE corpus to date (discussed in Chapter 4)—, which was constructed from the articles of the entire English Wikipedia. We used the OpenIE system MinIE-SpaTe (discussed in Chapter 3), because it provides more compact extractions, which are useful for downstream tasks such as KB population [Lin et al. 2020]. OpenIE extractions by themselves are shallow representations of text, which means that they tend to be ambiguous. To address such ambiguity, OPIEC keeps the links annotated by humans from within the text of the Wikipedia articles, thus making it the largest OpenIE corpus that contains golden disambiguation links for the arguments. Moreover, OPIEC contains large range of syntactic annotations (e.g., dependency parse structure of the input sentence, POS tags of tokens, spans, word indices, etc.), semantic annotations (e.g., NER tags, polarity, modality, etc.) and provenance information (e.g., input sentence, Wikipedia article ID where the extraction was made from, etc.). In subsequent work, Broscheit et al. [2020] used a filtered version of OPIEC—whereas they used the corpus annotations for filtering—to create a benchmark for the newly proposed task of open link prediction.

Together with the data, we also published code for the whole pipeline of constructing OPIEC. In principle, the pipeline of OPIEC can be used on any textual data. Some parts of the pipeline, however, need more work. First, we observed that coreference of the arguments is frequent source of ambiguity, which makes the triples to be less useful for downstream tasks. We believe that one promising direction for future work is to effectively disambiguate OpenIE arguments when possible. Second, we observed that the confidence scores are sometimes misleading. For example, we found the correctly extracted OpenIE triple *(“Monique Leyrac”; “is actress from”; “Quebec”)* has low confidence score (0.34). The reason for the low confidence score of this particular extraction stems from the dependency parse of the input sentence. In particular, the confidence scorer assigns lower confidence when there are one or several conjunctions in the sentence, because one of the most common errors from the dependency parser come from conjunctions. For these reasons, we believe that one promising research direction is the construction of more advanced confidence scorer for any given OpenIE extraction. Such confidence score system would not depend on prior processing (e.g., dependency parsing) and would be trained in unsupervised manner on textual data (e.g., considering only spans or some strategies exploiting the concept of language modeling). If

such a scoring system is scalable, then it could be applied on any large OpenIE corpus, and would override the currently provided confidence scores.

Once constructed, large OpenIE corpora are used for many other downstream tasks for deeper semantic understanding, including question answering, KB completion and slot filling. Since the structure of OpenIE extractions is usually in the form of triples—as it is the case with KBs—, they often are used as resource that complements KBs and, therefore, serve for either extending the KBs themselves or for providing further context to the KBs for other downstream tasks. Such work implies several possible manners in which OpenIE corpora and KBs are semantically related: 1) open relations are mapped to KB relations; 2) an OpenIE triple that is aligned with a KB triple which have the same argument pairs have equivalent meaning (i.e., the distant supervision assumption); 3) each OpenIE triple can be represented with a *single* KB triple. Such assumptions were not studied directly in prior work. Instead, the authors usually work on a downstream task and they either just assume that such assumptions hold or they test them w.r.t. the downstream task. In this thesis (Chapter 5) we directly and (mostly) manually studied these assumptions.

First, through the notion of KB hits, we aggregated the counts for each open relation w.r.t. KB relations. We found that OpenIE relations tend to be highly ambiguous, and one should not map them directly to KB relations. Instead, we found that it is possible in many cases to align OpenIE triples with KB triples that have the same argument pairs, though such alignments need to be done on instance level, because the OpenIE triple itself often provides the necessary context. Even though it is often correctly assumed that such alignments imply that the OpenIE triple and the aligned KB triple have the same meaning, we found that this is not entirely true. In particular, the OpenIE triples are often more specific than the KB triple. This means that the OpenIE triple can imply the KB triple, but the KB triple cannot imply the OpenIE triple. For instance, the OpenIE triple *(“P. J. Kennedy”; “is grandfather of”; “J. F. Kennedy”)* implies the KB triple (P. J. Kennedy; dbo:relative; J. F. Kennedy), but not the other way around. Next, we studied to what extent an OpenIE triple can be expressed with a single KB triple. We found that in most cases it is possible for an OpenIE triple to be expressed with a single KB fact, though the OpenIE triple is more specific. We found that the use of KB formula or multiple KB triples often helps to increase such expressibility. Finally, we studied to what extent the information found in OpenIE corpora is missing in reference KBs which were constructed from the same domain (i.e., Wikipedia). We found that most of the OpenIE triples that are relevant for KBs (i.e., they can be either fully or partially expressed with KB language) are either not present in the KB at all or are only partially present. This observation shows the potential of knowledge found in OpenIE corpora which can be used for extending existing KBs.

Overall, we believe that further research work towards compact OpenIE will make OpenIE systems more useful for downstream tasks. In this thesis, we discussed methods, corpora and analysis of compact Open Information Extraction w.r.t. reference KBs. Empirical evidence provided by this thesis and other work (e.g. KBPearl [Lin et al. 2020] and SalIE [Ponza et al. 2018]) suggest that improving compactness of OpenIE will likely make the OpenIE systems more useful for downstream tasks. Current trends in OpenIE suggest that exploiting deep learning techniques (and supervised learning paradigm in general) are promising directions for solving OpenIE-related problems. For now, such approaches suffer from lack of training data. Therefore, we believe that research for providing large amounts of training data for OpenIE-related problems are a promising future direction of research. Other possible research direction is to study the usefulness of compact OpenIE for other downstream tasks, such as question answering. Because KB population is important for other downstream tasks and Lin et al. [2020] showed empirical evidence for the usefulness of compact OpenIE for KB population, we believe that compact OpenIE is likely to affect other downstream tasks as well. Therefore, we think that this is yet another interesting direction of research for future work. Finally, we observed that, even though there has been research in OpenIE for languages other than English, yet, the research community has spent much less efforts in studying them. So far, there has been no study of investigating compactness of OpenIE for other languages. We believe that studying the methods for compact OpenIE proposed in this thesis for other languages (as well as studying other methods for achieving compactness) are promising direction for future work.

Appendix A

Annotation Guidelines

A.1 General Overview

Our goal is to evaluate Open Information Extraction (OpenIE) systems. OpenIE aims to extract relations and their arguments from unstructured text in unsupervised manner. In its simplest form, an OpenIE system extracts triples (or n-ary tuples) consisting of subject, relation, and object from a given sentence. For example, from the sentence:

"Bell is a telecommunication company which is based in Los Angeles."

an OpenIE system may extract the facts:

("Bell", "is", "telecommunication company")

("Bell", "is based in", "Los Angeles")

In this study, we focus on *clause-based OpenIE*. A *clause* is a part of a sentence that expresses some coherent piece of information. For instance, the sentence above consists of the two clauses:

"Bell is a telecommunication company"

"Bell is based in Los Angeles"

The extractions are generated out of individual clauses, which means that we ignore the information in any other clauses (details later).

A.2 Labeling

Each extraction is labeled with two labels (correct / incorrect):

1. **Fact label:** correctness of the extracted fact itself (together with its factuality and quantity values)

2. **Attribution label:** correctness of the extracted attribution

We discuss these two labels in what follows.

A.2.1 Fact Label

The *fact label* indicates whether the whole extraction (ignoring an attribution, if any) is correct.

Rule F1: An extraction is considered *correct* if it contains all necessary information from the clause from which it has been extracted. In general, we label extractions as *correct* if they are entailed by their clause. For example, from the sentence:

"Bell is a telecommunication company which is based in Los Angeles."

both of the following extractions should be labeled as *correct*:

("Bell", "is", "telecommunication company")
("Bell", "is based in", "Los Angeles")

In contrast, the triple:

("Bell", "is based in", "telecommunication company")

is considered *incorrect*.

Rule F2: Information present in all clauses other the one from which the extraction has been taken must be ignored. This includes subordinate clauses (such as conditionals, clauses connected with "while", and so on). For example, in the sentence:

"If it rains, the grass gets wet."

all of the following extractions should be labeled as correct:

("grass", "gets", "wet")
("grass", "gets wet if", "it rains")
("it", "rains")

Rule F3: Further information present in the clause can be ignored if this information does not change the meaning of the extraction. For example, from the sentence:

"Albert Einstein was born in 1879 in Ulm."

then both of the following extractions should be considered *correct*:

("Albert Einstein", "was born in 1879 in", "Ulm")
("Albert Einstein", "was born in", "Ulm")

If, however, the lack of a constituent of a clause is not sufficient, this should be labeled as *incorrect:*

("Albert Einstein", "was")

Rule F4: In cases when relations/arguments lack words which contain crucial information (i.e. the lacking of word(s) is changing the meaning of the triple), the *factual label* should be *incorrect*. For example, from the sentence:

"Jack likes data mining."

extractions should be labeled as follows:

("Jack", "likes", "data mining") as *correct*
("Jack", "likes", "mining") as *incorrect*

because mining and data mining have different meaning.

Rule F5: Some OpenIE systems output some implicit extractions, which may contain words that are not present in the input sentence. Nevertheless, they should also be labeled as *correct* if they correctly represent the information given in their clause. For example, from the sentence:

"Mr. Mike Johnson lives in Berlin, Germany."

the following extractions should be labeled as *correct*:

("Mike Johnson", "is", "male")
("Berlin", "is in", "Germany")

and the following extraction as *incorrect*:

("Berlin", "is", "Germany")

We refer to the combination of *polarity* and *modality* as *factuality* (more on these later). For example, if the polarity is *"negative" (-)* and modality is *"possibility" (PS)*, then the factuality is *"negative possibility"* (briefly written as *(-, PS)*). *Polarity* distinguishes between *positive (+)* and *negative (–)* instantiations of the triples, that is, it conveys the distinction between affirmative and negative contexts.

Rule F6: The extraction should only be labeled *correct* if it expresses the same polarity (positive or negative) as its clause. For example, suppose we have the sentence:

""John did not need the training."

then the *factual label* on both of the following extractions should be *correct*:

1. ("John", "did need", "training") *2. ("John", "did not need", "training")*
Polarity: NEGATIVE $\Longleftrightarrow$ *Polarity: POSITIVE*

Rule F7: If the polarity value is contained in another clause, then you should ignore the negative context in the extraction. For example, if you have the sentence:

"It's not true that John Smith lives in Italy."

then, we have two clauses:

1. John Smith lives in Italy.
2. It's not true that John Smith lives in Italy.

Clause 2 contains clause 1. However, the negation is within clause 2 and not part of clause 1, which means for the *factuality label* the following extraction should be labeled as *correct:*

("John Smith", "lives in", "Italy")
polarity: positive; modality: certainty; (+, CT)

The *modality* is the part of the factual annotation that gives us information of whether an extraction is a *certainty* or a *possibility* within a clause.

Rule F8: The extraction should only be labeled correct if it expresses the same modality (certainty or possibility) as its clause. Suppose we have the following sentences:

"Dewayne Robertson expects to meet with the Jets"
"Dewayne Robertson probably meets with the Jets"
"Dewayne Robertson will meet with the Jets"

then, the *factual label* of the both of the following extractions should be *correct:*

1. ("D. R.", "meet with", "Jets") *2. ("D. R.", "will meet with", "Jets")*
Polarity: positive $\Longleftrightarrow$ *Polarity: positive*
Modality: possibility *Polarity: certainty*

because in the first sentence, Dewayne Robertson "expects to" meet with the Jets, which is merely a possibility, not a certainty. In the second one, it is a future tense ("will" meet ...), which is also not a certainty, but a possibility.

Rule F9: If the quantity(ies) within the extraction contain the proper phrase for indicating a phrase which expresses some sort of quantity (given that all the other rules F1 to F8 are also correct), the *factual label* is *correct.* For example, if you have the sentence:

"At least two e-mails were marked as confidential."

then the *factual label* for the following extraction is considered as *correct:*

("QUANT_S_1 e-mails", "were marked as", "confidential")
Factuality: (+, CT)
QUANT_S_1 = At least two

Rule F10: If at least one quantity placeholder is not represented correctly in the extraction (i.e. it lacks crucial information), then the *factual label* is *incorrect*. For example, considering the same sentence:

"At least two e-mails were marked as confidential."

The following extraction's *factual label* is considered as *incorrect*:

("QUANT_S_1 e-mails", "were marked as", "confidential")
Factuality: (+, CT)
QUANT_S_1 = At least

Rule F11: Ignore correference resolution parts. For example, if we have the sentence:

"John was home and he opened the door."

then both of the following extractions are considered to be equivalent and *correct:*

("John", "opened", "door") $\Longleftrightarrow$ *("He", "opened", "door")*

Rule F12: Ignore wrong form of a word as long as the lemmas are correct. For example, in the sentence:

"John loves his wife."

the following two extractions are considered to be equivalent and *correct:*

("he", "has", "wife") $\Longleftrightarrow$ *("his", "has", "wife")*

A.2.2 Attribution Label

The *attribution* of a triple is the supplier of the information for the triple. The attributions themselves contain annotations for factuality, which are different from the annotation of factuality for the triple itself.

Rule A1: If attribution is captured within the extraction implicitly (it is within the extraction) or explicitly (it is annotated as an attribution), then the attribution should be labeled as *correct.* For example, for the sentence:

"The State Department does not believe that more than 3 million Americans live outside of the U.S."

both of the following extractions are equivalent and should be labeled as *correct:*

1. *("QUANT_S_1 Americans" "live outside of" "U.S.")*
   Attribution: (T. S. D., Factuality: (-, PS))
   Factuality: (+, CT)
   Quantities: [QUANT_S_1 = more than 3 million]
2. *("T. S. D.", "does not believe that", "3 m. Americans live outside of U.S.")*
   Attribution: no attribution detected
   Factuality: (+, CT)

The attribution's modality is *"possibility"* because the attributer believes the statement. If the predicate was not *"believe"*, but it was a predicate expressing a certainty, like *"knows"*, then the modality would have been *"certainty"*.

Rule A2: If the attribution phrase contains words which are not part of the attribution phrase (e.g. instead of *"The State Department"* to have *"The State Department does"*) or lacks some words which are essential to the meaning of the attribution phrase (e.g. instead of *"The State Department"* to have just *"Department"*) then the attribution should be labeled as *incorrect*.

Rule A3: If one of the attribution's factuality values is wrong (e.g. instead of *"negative polarity"* we have *"positive polarity"* or instead of *"possibility modality"* we have *"certainty modality"*), then the *attribution label* should be *incorrect*.

Appendix B

Further Alignments With DBpedia

Open relation	Frequency in OPIEC-Clean	Frequency in OPIEC-Link	# KB hits	# distinct KB rels.	Top-3 mapped DBpedia rel. and hit frequency
"leave"	130,515	1,356	347 (25.6%)	54	associatedBand 70 associatedMusicalArtist 70 formerBandMember 35
"take"	127,757	660	49 (7.4%)	26	writer 4 previousWork 4 artist 3
"use"	127,537	2,951	123 (4.2%)	53	currency 9 affiliation 8 timeZone 7
"receive"	118,429	2,133	268 (12.6%)	19	award 236 team 4 debutTeam 4
"make"	116,688	1,063	140 (13.2%)	39	director 39 writer 25 producer 12
"be member of"	104,680	11,480	3,361 (29.3%)	80	associatedBand 740 associatedMusicalArtist740 party 584
"return to"	104,392	482	117 (24.3%)	35	team 44 league 9 associatedBand 7
"be at"	102,844	20,328	8,314 (40.9%)	78	ground 1,887 city 1,759 location 1,109
"be species of"	101,846	54,269	13,639 (25.1%)	9	order 5,196 family 4,269 kingdom 2,826
"move to"	100,226	1,409	316 (22.4%)	43	team 124 managerClub 35 ground 16
"be write by"	96,790	6,340	1,956 (30.9%)	50	author 571 writer 457 notableWork 120
"be found in"	95,163	836	110 (13.2)%	17	location 21 city 19 headquarter 17

Table B.1 The most frequent open relations in OPIEC-Clean, along with DBpedia mapping information from OPIEC-Link (continuation of Tab. 5.2)

Appendix C

Reference Corpora and Methodology

C.1 OpenIE Data and Methodology

OpenIE Corpus

One of the major problems of aligning OpenIE triples with KB facts is that the OpenIE triples are consisted of surface patterns, which makes the triples highly ambiguous. To make such alignments possible, it is necessary that the arguments of the OpenIE triples are disambiguated. For these reasons, we chose OPIEC-Linked (described in Chapter 4) as an OpenIE corpus for our study, because it is the biggest OpenIE corpus to date, which consists 6M triples with disambiguated arguments. As explained in Chapter 4, OPIEC-Linked was constructed by running the OpenIE system MinIE-SpaTe (discussed in section 3.7) over the entire English Wikipedia. The links in the text added by Wikipedia authors were kept, which makes the corpus to be consisted of golden disambiguation links for the arguments.

OPIEC Filters

The goal of the study is to investigate the limits of aligning OpenIE triples with KBs. For this reason, we assume both a perfect extractor and perfect alignments between the OpenIE triples and the KB facts. To reduce the noise from OPIEC-Linked, we followed Broscheit et al. [2020] and filtered out the triples having the following properties: 1) confidence score is less than 0.3; 2) extraction type is SVOO, SVOC or extractions are made from the *apposition* dependency parse relation. In a preliminary study, we found these triples to be very noisy. For the remainder of this chapter, we will refer to this data as OPIEC for simplicity.

Sampling Correctly Extracted OpenIE Triples for the DSA Study

The DSA implies that for each (correctly extracted) OpenIE triple which has a KB-hit, the open relation expresses the same information as the KB relation. The goal of the study is to *investigate the limits* of such alignments, which is why we consider only extractions which are correctly extracted. For this reason, we took a random sample of 200 OpenIE triples from the OpenIE triples having KB-hits, which were labeled for correctness by an expert. To ensure that the information of the triple is complete, the triples which are not self-contained were labeled as "incorrectly extracted". For example, the triple *(Pope Clement VII; "named him inquisitor of"; Modena)* is not self-contained, because it is not clear to which entity "him" refers to. The labeler stopped at the 100th correctly extracted triple. Note that these 100 correctly extracted triples are also self-contained. These 100 OpenIE triples were used for our DSA study.

Study Design for *Is-a relation* OpenIE triples

The study for *Is-a relation* triples is similar with the one done on *All relations*. We sampled 100 correctly extracted triples from OPIEC-Typed (i.e. the subset of OPIEC containing triples of the form *(subject, "be"; object)*). For each correctly extracted OPIEC-Typed triple, we matched the subject link with all the DBpedia entries for types. As a result, we have an OpenIE triple *(subject, "be"; object)* and on the KB side we have (subject; type; T). The sampling and labeling logic is the same as the one explained in the previous paragraph.

C.2 KB Data and Methodology for the DSA Study

Reference KB

For the alignments, it is very important that the KB contains the same information as the text corpus from which the OpenIE data was constructed (i.e. that both the OpenIE triples and the reference KB were automatically constructed from the same domain). This ensures that the information in the KB and the information content in the OpenIE triples is the same. In such settings, the OpenIE arguments have the same ID links as the KB entities, which makes the study comparable. For these reasons, we chose DBpedia [Auer et al. 2007] as a reference KB, because it is a well-established KB constructed from Wikipedia (the same resource from which OPIEC is constructed), and because it is the largest KB to date which is automatically constructed from Wikipedia. Prior work for aligning OpenIE triples with KB facts also exploited the combination of Wikipedia and DBpedia [Wu and Weld 2010; Dutta et al. 2013; 2014; 2015; Yu et al. 2017; Gashteovski et al. 2019].

DBpedia-filtered

For our study, it is essential that both of the KB triple arguments are disambiguated. Therefore, from DBpedia, we filtered out any triples containing literals, abstracts, dates, etc. Many of the relations in DBpedia are extracted with generic infobox extraction. These KB relations tend to be noisier — sometimes even ambiguous — and they often lack important information describing the precise semantics of the KB relation [Bizer et al. 2009] (e.g. domain/range types or descriptions are often missing.). For these reasons, we filtered out these KB triples as well. We retained only the triples that were extracted with mapping-based infobox extraction (i.e. with namespace `http://dbpedia.org/ontology`), because of their higher extraction quality and higher level of details they provide. This way, it is much clearer to an expert labeler to assess the alignments.

List of figures

List of tables

References

Prabal Agarwal, Jannik Strötgen, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. DiaNED: Time-Aware Named Entity Disambiguation for Diachronic Corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 686–693, 2018.

Alan Akbik and Alexander Löser. KrakeN: N-ary Facts in Open Information Extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX@NAACL-HLT)*, pages 52–56, 2012.

Geoffrey Andogah, Gosse Bouma, and John Nerbonne. Every Document has a Geographical Scope. *Data & Knowledge Engineering*, 81:1–20, 2012.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 344–354, 2015.

Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. Enriching Knowledge Bases with Interesting Negative Statements. In *Proceedings of the Conference of Automatic Knowledge Base Construction (AKBC)*, 2020.

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

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. Rel-grams: A Probabilistic Model of Relations in Text. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX@NAACL-HLT)*, pages 101–105, 2012.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. Generating Coherent Event Schemas at Scale. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1721–1731, 2013.

Michele Banko and Oren Etzioni. The Tradeoffs between Open and Traditional Relation Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 28–36, 2008.

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

Akim Bassa, Mark Kröll, and Roman Kern. GerIE - An Open Information Extraction System for the German Language. *The Journal of Universal Computer Science (J. UCS)*, 24(1): 2–24, 2018.

Hannah Bast and Elmar Haussmann. Open Information Extraction via Contextual Sentence Decomposition. In *Proceedings of the International Conference on Semantic Computing (ICSC)*, pages 154–159, 2013.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. CaRB: A Crowdsourced Benchmark for Open IE. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6263–6268, 2019.

Nikita Bhutani and H V Jagadish. Online Schemaless Querying of Heterogeneous Open Knowledge Bases. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 699–708, 2019.

Nikita Bhutani, H V Jagadish, and Dragomir Radev. Nested Propositions in Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 55–64, 2016.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - A Crystallization Point for the Web of Data. *Journal of Web Semantics*, 7(3):154–165, 2009.

Eduardo Blanco and Dan I. Moldovan. Some Issues on Detecting Negation from Text. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2011.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 2787–2795, 2013.

Claudio Delli Bovi, Luis Espinosa Anke, and Roberto Navigli. Knowledge Base Unification via Sense Embeddings and Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 726–736, 2015.

Samuel Broscheit, Kiril Gashteovski, and Martin Achenbach. OpenIE for Slot Filling at TAC KBP 2017 - System Description. In *TAC*, 2017.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. Can We Predict New Facts with Open Knowledge Graph Embeddings? A Benchmark for Open Link Prediction. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, pages 2296–2308, 2020.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. ClueWeb09 Data Set, 2009.

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

Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. Graphene: Semantically-Linked Propositions in Open Information Extraction. In *Prooceedings of International Conference on Computational Linguistics (COLING)*, pages 2300–2311, 2018.

Angel X. Chang and Christopher D. Manning. SUTime: A Library for Recognizing and Normalizing Time Expressions. In *Proceedings of the International Language Resources and Evaluation Conference (LREC)*, pages 3735–3740, 2012.

Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, 2014.

Janara Christensen, Stephen Soderland, and Oren Etzioni. An Analysis of Open Information Extraction Based on Semantic Role Labeling. In *Proceedings of the International Conference on Knowledge Capture (K-CAP)*, pages 113–120, 2011.

Lei Cui, Furu Wei, and Ming Zhou. Neural Open Information Extraction. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 407–413, 2018.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. Incorporating Selectional Preferences in Multi-hop Relation Extraction. In *Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC@NAACL-HLT)*, pages 18–23, 2016.

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

Leandro Souza de Oliveira, Rafael Glauber, and Daniela Barreiro Claro. DependentIE: An Open Information Extraction System on Portuguese by a Dependence Analysis. *Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, 2017.

Luciano Del Corro and Rainer Gemulla. ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the International World Wide Web Conferences (WWW)*, pages 355–366, 2013.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. FINET: Context-Aware Fine-Grained Named Entity Typing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 868–878, 2015.

Claudio Delli Bovi, Luis Espinosa-Anke, and Roberto Navigli. Knowledge Base Unification via Sense Embeddings and Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 726–736, 2015a.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. Large-Scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic Analysis. *Transactions of the Association for Computational Linguistics (TACL)*, 3:529–543, 2015b.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 601–610, 2014.

Arnab Dutta, Christian Meilicke, Mathias Niepert, and Simone Paolo Ponzetto. Integrating Open and Closed Information Extraction: Challenges and First Steps. In *Proceedings of Workshop NLP-DBPEDIA@ISWC*, 2013.

Arnab Dutta, Christian Meilicke, and Heiner Stuckenschmidt. Semantifying Triples from Open Information Extraction Systems. In *Proceedings of the European Starting AI Researcher Symposium (STAIRS)*, pages 111–120, 2014.

Arnab Dutta, Christian Meilicke, and Heiner Stuckenschmidt. Enriching Structured Knowledge with Open Information. In *Proceedings of the International Conference on the World Wide Web (WWW)*, pages 267–277, 2015.

Luis Espinosa-Anke, Jose Camacho-Collados, Sara Rodríguez Fernández, Horacio Saggion, and Leo Wanner. Extending WordNet with Fine-Grained Collocational Information via Supervised Distributional Learning. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 3422–3432, 2016.

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.

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

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-Driven Learning for Open Question Answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1608–1618, 2013.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open Question Answering over Curated and Extracted Knowledge Bases. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1156–1165, 2014.

Tobias Falke, Gabriel Stanovsky, Iryna Gurevych, and Ido Dagan. Porting an Open Information Extraction System from English to German. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 892–898, 2016.

Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object Detection Meets Knowledge Graphs. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1661–1667, 2017.

Manaal Faruqui and Shankar Kumar. Multilingual Open Relation Extraction using Cross-lingual Projection. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1351–1356, 2015.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, pages 363–370, 2005.

Corina Forăscu and Dan Tufiş. Romanian TimeBank: An Annotated Parallel Corpus for Temporal Information. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3762–3766, 2012.

Cong Fu, Tong Chen, Meng Qu, Woojeong Jin, and Xiang Ren. Collaborative Policy Learning for Open Knowledge Graph Reasoning. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2672–2681, 2019.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. Canonicalizing Open Knowledge Bases. In *Proceedings of the International Conference on Conference on Information and Knowledge Management (CIKM)*, pages 1679–1688, 2014.

Pablo Gamallo and Marcos Garcia. Multilingual Open Information Extraction. In *Portuguese Conference on Artificial Intelligence (EPIA)*, pages 711–722, 2015.

Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. Dependency-Based Open Information Extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 10–18, 2012.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. MinIE: Minimizing Facts in Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2630–2640, 2017.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. OPIEC: An Open Information Extraction Corpus. *In Proceedings of the Conference on Automated Knowledge Base Construction (AKBC)*, 2019.

Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. On Aligning OpenIE Extractions with Knowledge Bases: A Case Study. In *Proceedings of the Workshop of Evaluation and Comparison of NLP Systems (Eval4NLP@EMNLP)*, 2020.

Rafael Glauber, Leandro Souza de Oliveira, Cleiton Fernando Lima Sena, Daniela Barreiro Claro, and Marlo Souza. Challenges of an Annotation Task for Open Information Extraction in Portuguese. In *International Conference on Computational Processing of the Portuguese Language (PROPOR)*, pages 66–76, 2018.

Fabrizio Gotti and Philippe Langlais. Weakly Supervised, Data-Driven Acquisition of Rules for Open Information Extraction. In *Proceedings of the Canadian Conference on Artificial Intelligence (CCAI)*, pages 16–28, 2019.

Stefan Th. Gries. 50-Something Years of Work on Collocations: What Is or Should Be Next... *International Journal of Corpus Linguistics*, 18(1):137–166, 2013.

Stefan Th. Gries. 50-Something Years of Work on Collocations. *Current Issues in Phraseology*, 74:135, 2015.

Adam Grycner and Gerhard Weikum. HARPY: Hypernyms and Alignment of Relational Paraphrases. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2195–2204, 2014.

Raffaele Guarasci, Emanuele Damiano, Aniello Minutolo, and Massimo Esposito. Towards a Gold Standard Dataset for Open Information Extraction in Italian. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 447–453, 2019.

Raffaele Guarasci, Emanuele Damiano, Aniello Minutolo, Massimo Esposito, and Giuseppe De Pietro. Lexicon-Grammar based Open Information Extraction from Natural Language Sentences in Italian. *Expert Systems with Applications*, 143, 2020.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. CaRe: Open Knowledge Graph Embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 378–388, 2019.

Tom Harting, Sepideh Mesbah, and Christoph Lofi. LOREM: Language-consistent Open Relation Extraction from Unstructured text. In *Proceedings of The Web Conference (WWW)*, pages 1830–1838, 2020.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 643–653, 2015.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep Semantic Role Labeling: What Works and What's Next. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 473–483, 2017.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge Graph Embedding Based Question Answering. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 105–113, 2019.

Seohyun Im, Hyunjo You, Hayun Jang, Seungho Nam, and Hyopil Shin. KTimeML: Specification of Temporal and Event Expressions in Korean Text. In *Proceedings of the Workshop on Asian Language Resources (ALR7@IJCNLP)*, pages 115–122, 2009.

Prachi Jain and Mausam. Knowledge-Guided Linguistic Rewrites for Inference Rule Verification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 86–92, 2016.

Shengbin Jia, Shijia E, Maozhen Li, and Yang Xiang. Chinese Open Relation Extraction and Knowledge Base Establishment. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(3):1–22, 2018a.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. TEQUILA: Temporal Question Answering over Knowledge Bases. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 1807–1810, 2018b.

Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M. Kaplan, Timothy P. Hanratty, and Jiawei Han. MetaPAD: Meta Pattern Discovery from Massive Text Corpora. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 877–886, 2017.

Amina Kadry and Laura Dietz. Open Relation Extraction for Support Passage Retrieval: Merit and Open Issues. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1149–1152, 2017.

Tushar Khot, Ashish Sabharwal, and Peter Clark. Answering Complex Questions Using Open Information Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–316, 2017.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. IMoJIE: Iterative Memory-Based Joint Open Information Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5871–5886, 2020.

Anne Lauscher, Yide Song, and Kiril Gashteovski. MinScIE: Citation-centered Open Information Extraction. In *Proceedings of Joint Conference on Digital Libraries (JCDL)*, pages 386–387, 2019.

William Léchelle, Fabrizio Gotti, and Philippe Langlais. Wire57: A Fine-grained Benchmark for Open Information Extraction. In *Proceedings of the Linguistic Annotation Workshop (LAW@ACL)*, pages 6–15, 2019.

Thomas Lin, Mausam, and Oren Etzioni. Entity Linking at Web Scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX@NAACL-HLT)*, pages 84–88, 2012.

Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. KBPearl: A Knowledge Base Population System Supported by Joint Entity and Relation Linking. In *Proceedings of the Very Large Data Base Endowment (PVLDB)*, pages 1035–1049, 2020.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2395–2405, 2018.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. OpenCeres: When Open Information Extraction Meets the Semi-Structured Web. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3047–3056, 2019.

Alexander Löser, Sebastian Arnold, and Tillmann Fiehn. The GoOLAP Fact Retrieval Framework. In *European Business Intelligence Summer School (eBISS)*, pages 84–97, 2011.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Conference on Innovative Data Systems Research (CIDR)*, 2013.

William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 55–60, 2014.

Mausam. Open Information Extraction Systems and Downstream Applications. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4074–4077, 2016.

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

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. Crowdsourcing question-answer meaning representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 560–568, 2018.

George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant Supervision for Relation Extraction without Labeled Data. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1003–1011, 2009.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, B Dalvi, Matt Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-Ending Learning. *Communications of the ACM*, 61(5):103–115, 2018.

Andrea Moro and Roberto Navigli. WiSeNet: Building a Wikipedia-Based Semantic Network with Ontologized Relations. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 1672–1676, 2012.

Andrea Moro and Roberto Navigli. Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 2148–2154, 2013.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1135–1145, 2012.

Sangha Nam, Younggyun Hahm, Sejin Nam, and Key-Sun Choi. SRDF: Korean Open Information Extraction using Singleton Property. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2015.

Federico Nanni, Simone Paolo Ponzetto, and Laura Dietz. Entity-aspect linking: providing fine-grained semantics of entities in context. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, pages 49–58, 2018.

Federico Nanni, Jingyi Zhang, Ferdinand Betz, and Kiril Gashteovski. EAL: A Toolkit and Dataset for Entity-Aspect Linking. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, pages 430–431, 2019.

María Navas-Loro, Erwin Filtz, Víctor Rodríguez-Doncel, Axel Polleres, and Sabrina Kirrane. TempCourt: Evaluation of Temporal Taggers on a new Corpus of Court Decisions. *The Knowledge Engineering Review*, 34, 2019.

Roberto Navigli and Simone Paolo Ponzetto. BabelNet: Building a very large Multilingual Semantic Network. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 216–225, 2010.

Neha Nayak, Mark Kowarsky, Gabor Angeli, and Christopher D. Manning. A Dictionary of Nonsubsective Adjectives. Technical report, 2014.

Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-REED: Joint Relation Extraction and Entity Disambiguation. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, pages 2227–2230, 2017.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 11, pages 809–816, 2011.

Harinder Pal and Mausam. Demonyms and Compound Relational Nouns in Nominal Open IE. In *Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC@NAACL-HLT)*, pages 35–39, 2016.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. YAGO 4: A Reason-able Knowledge Base. *The Semantic Web*, 2020.

Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. CORE: Context-Aware Open Relation Extraction with Factorization Machines. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1763–1773, 2015.

Marco Ponza, Luciano Del Corro, and Gerhard Weikum. Facts that matter. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1043–1048, 2018.

Rifki Afina Putri, Giwon Hong, and Sung-Hyon Myaeng. Aligning Open IE Relations and KB Relations using a Siamese Network Based on Word Embedding. In *Proceedings of the International Conference on Computational Semantics (ICCS)*, pages 142–153, 2019.

Likun Qiu and Yue Zhang. ZORE: A Syntax-based System for Chinese Open Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1870–1880, 2014.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, Jan Svartvik, and David Crystal. *A Comprehensive Grammar of the English Language*. Cambridge University Press, 1985.

Mahmoud Rahat and Alireza Talebpour. Open Information Extraction as an Intermediate Semantic Structure for Persian Text Summarization. *International Journal on Digital Libraries*, 19(4):339–352, 2018a.

Mahmoud Rahat and Alireza Talebpour. Parsa: An Open Information Extraction System for Persian. *Digital Scholarship in the Humanities (DSH)*, 33(4):874–893, 2018b.

Mahmoud Rahat, Alireza Talebpour, and Seyedamin Monemian. A Recursive Algorithm for Open Information Extraction from Persian Texts. *International Journal of Computer Applications in Technology (IJCAT)*, 57(3):193–206, 2018.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation Extraction with Matrix Factorization and Universal Schemas. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 74–84, 2013.

Youngbin Ro, Yukyung Lee, and Pilsung Kang. Multi2OIE: Multilingual Open Information Extraction based on Multi-Head Attention with BERT. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Ade Romadhony, Ayu Purwarianti, and Dwi H Widyantoro. Rule-based Indonesian Open Information Extraction. In *International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, pages 107–112, 2018.

Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. Supervising Unsupervised Open Information Extraction Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 728–737, 2019.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You *can* Teach an Old Dog new Tricks! On Training Knowledge Graph Embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Swarnadeep Saha and Mausam. Open Information Extraction from Conjunctive Sentences. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2288–2299, 2018.

Swarnadeep Saha, Harinder Pal, and Mausam. Bootstrapping for Numerical Open IE. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 317–323, 2017.

Rishiraj Saha Roy and Avishek Anand. Question Answering over Curated and Open Web Sources. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2432–2435, 2020.

Raana Saheb-Nassagh, Majid Asgari, and Behrouz Minaei-Bidgoli. RePersian: An Efficient Open Information Extraction Tool in Persian. In *International Conference on Web Research (ICWR)*, pages 93–99, 2020.

Evan Sandhaus. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 2008.

Roser Saurí and James Pustejovsky. Are You Sure That This Happened? Assessing the Factuality Degree of Events in Text. *Computational Linguistics*, 38(2):261–299, 2012.

Roser Saurı, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. TimeML Annotation Guidelines Version 1.2.1. Technical report, 2006.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A Gers, and Alexander Löser. RelVis: Benchmarking OpenIE Systems. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2017.

Cleiton Fernando Lima Sena, Rafael Glauber, and Daniela Barreiro Claro. Inference Approach to Enhance a Portuguese Open Information Extraction. In *International Conference on Enterprise Information Systems (ECEIS)*, volume 2, pages 442–451, 2017.

Yongpan Sheng and Zenglin Xu. Coherence and Salience-Based Multi-Document Relationship Mining. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 414–430, 2019.

Yongpan Sheng, Zenglin Xu, Yafang Wang, and Gerard de Melo. MuReX: Multi-Document Semantic Relation Extraction for News Analytics. *WWW J*, 23(3):2043–2077, 2020.

Baoxu Shi and Tim Weninger. Open-World Knowledge Graph Completion. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1957–1964, 2018.

Giovanni Siragusa, Rohan Nanda, Valeria De Paiva, and Luigi Di Caro. Relating Legal Entities via Open Information Extraction. In *Research Conference on Metadata and Semantics Research (MTSR)*, pages 181–187, 2018.

Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. Adapting Open Information Extraction to Domain-Specific Relations. *AI magazine*, 31(3):93–102, 2010.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld. Open Information Extraction to KBP Relations in 3 Hours. In *Proceedings of the Text Analysis Conference (TAC)*, 2013.

Stephen Soderland, Natalie Hawkins, John Gilmer, and Daniel S. Weld. Combining Open IE and Distant Supervision for KBP Slot Filling. In *Proceedings of the Text Analysis Conference (TAC)*, 2015a.

Stephen Soderland, Natalie Hawkins, Gene L. Kim, and Daniel S. Weld. University of Washington System for 2015 KBP Cold Start Slot Filling. In *Proceedings of the Text Analysis Conference (TAC)*, 2015b.

Gabriel Stanovsky and Ido Dagan. Creating a Large Benchmark for Open Information Extraction. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2300–2305, 2016.

Gabriel Stanovsky, Ido Dagan, et al. Open IE as an Intermediate Structure for Semantic Tasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 303–308, 2015.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised Open Information Extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 885–895, 2018.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: a Core of Semantic Knowledge. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 697–706, 2007.

Mingming Sun, Xu Li, and Ping Li. Logician and Orator: Learning from the Duality between Language and Knowledge in Open Domain. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2119–2130, 2018a.

Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. Logician: A Unified End-to-End Neural Approach for Open-Domain Information Extraction. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 556–564, 2018b.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 173–180, 2003.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2071–2080, 2016.

Benjamin Van Durme and Lenhart Schubert. Open Knowledge Extraction through Compositional Language Processing. In *Semantics in Text Processing (STEP)*, pages 239–254, 2008.

Shikhar Vashishth, Prince Jain, and Partha Talukdar. CESI: Canonicalizing Open Knowledge Bases using Embeddings and Side Information. In *Proceedings of the World Wide Web Conference (WWW)*, pages 1317–1327, 2018.

Natalia Viani, Hegler Tissot, Ariane Bernardino, and Sumithra Velupillai. Annotating Temporal Information in Clinical Notes for Timeline Reconstruction: Towards the Definition of Calendar Expressions. In *Proceedings of the BioNLP Workshop and Shared Task (BioNLP@ACL 2019)*, volume 18, pages 201–210, 2019.

Denny Vrandečić and Markus Krötzsch. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, 57(10):78–85, 2014.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. Open Relation Extraction for Chinese Noun Phrases. *IEEE Transactions on Knowledge and Data Engineering*, 2019a.

Xuan Wang, Yu Zhang, Qi Li, Yinyin Chen, and Jiawei Han. Open Information Extraction with Meta-Pattern Discovery in Biomedical Literature. In *Proceedings of the International Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM-BCB)*, pages 291–300, 2018.

Xueying Wang and Meng Jiang. Precise Temporal Slot Filling via Truth Finding with Data-Driven Commonsense. *Knowledge and Information Systems*, pages 1–27, 2020.

Xueying Wang, Haiqiao Zhang, Qi Li, Yiyu Shi, and Meng Jiang. A Novel Unsupervised Approach for Precise Temporal Slot Filling from Incomplete and Noisy Temporal Contexts. In *The World Wide Web Conference (WWW)*, pages 3328–3334, 2019b.

Sebastian Wanner. Development of a Confidence Prediction Model for MinIE. Master's thesis, University of Mannheim, Germany, 2017.

Daniel S. Weld, Raphael Hoffmann, and Fei Wu. Using Wikipedia to Bootstrap Open Information Extraction. *ACM SIGMOD Record*, 37(4):62–68, 2009.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. Pocket Knowledge Base Population. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 305–310, 2017.

Fei Wu and Daniel S. Weld. Open Information Extraction Using Wikipedia. In *Proceed-inwu2010opengs of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 118–127, 2010.

Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, and Pengcheng Yin. Towards Practical Open Knowledge Base Canonicalization. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 883–892, 2018.

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

Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. Assertion-based QA with Question-Aware Open Information Extraction. In *Proceedings of the Conference of the American Association for Artificial Intelligence (AAAI)*, pages 6021–6028, 2018.

Dian Yu, Lifu Huang, and Heng Ji. Open Relation Extraction and Grounding. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 854–864, 2017.

Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache Spark: A Unified Engine for Big Data Processing. *Communications of the ACM*, 59(11):56–65, 2016.

Junlang Zhan and Hai Zhao. Span Model for Open Information Extraction on Accurate Corpus. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9523–9530, 2020.

Dongxu Zhang, Subhabrata Mukherjee, Colin Lockard, Xin Luna Dong, and Andrew McCallum. OpenKI: Integrating Open Information Extraction and Knowledge Bases with Relation Inference. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, pages 762–772, 2019.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative Knowledge Base Embedding for Recommender Systems. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 353–362, 2016.

Sheng Zhang, Kevin Duh, and Benjamin Van Durme. MT/IE: Cross-Lingual Open Information Extraction with Neural Sequence-to-Sequence Models. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 64–70, 2017.

Jie Zhou and Wei Xu. End-to-end Learning of Semantic Role Labeling Using Recurrent Neural Networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1127–1137, 2015.

Qi Zhu, Xiang Ren, Jingbo Shang, Yu Zhang, Ahmed El-Kishky, and Jiawei Han. Integrating Local Context and Global Cohesiveness for Open Information Extraction. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 42–50, 2019.