NEURAL METHODS FOR MONOLINGUAL AND MULTILINGUAL OPEN INFORMATION EXTRACTION

KESHAV SAI KOLLURU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
JULY 2023

NEURAL METHODS FOR MONOLINGUAL AND MULTILINGUAL OPEN INFORMATION EXTRACTION

by

KESHAV SAI KOLLURU

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the

INDIAN INSTITUTE OF TECHNOLOGY DELHI
JULY 2023

Certificate

This is to certify that the thesis titled **Neural Methods for Monolingual and Multilingual Open Information Extraction** being submitted by **Mr. Keshav Sai Kolluru** for the award of **Doctor of Philosophy** in Department of Computer Science and Engineering is a record of bonafide work carried out by him under my guidance and supervision at the **Department of Computer Science and Engineering, Indian Institute of Technology Delhi**. Unless otherwise stated explicitly, the work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma. In particular, some parts of work done in Chapters 3 and 6 was done jointly with undergraduate students and some part of Chapter 7 was done jointly with a master's student. In each case, the part done by the collaborators appeared in their respective theses.

Mausam
Professor
Department of Computer Science and Engg.
Indian Institute of Technology Delhi
New Delhi- 110016

Soumen Chakrabarti
Professor
Department of Computer Science and Engg.
Indian Institute of Technology Bombay
Mumbai- 400076

Acknowledgements

I am profoundly grateful to my advisors, Prof Mausam and Prof Soumen Chakrabarti, whose knowledge, wisdom, and mentorship have guided me unfailingly throughout the challenging but rewarding journey that is a PhD. Their keen insights, tireless dedication, and unwavering belief in my capabilities have played a pivotal role in my academic growth and accomplishment. I am grateful for their patience, constant encouragement, and invaluable advice.

Undertaking a PhD is a long, tough journey, akin to sailing in the vast ocean where one often encounters unanticipated storms and calm, peaceful stretches alike. It tests your strength, resilience, and determination in ways you never anticipated, yet you emerge stronger, wiser, and more resilient at the end of this voyage. You arrive at the shore equipped with a trove of knowledge, skills, and experiences that fortify your capabilities for future endeavours. In this profound journey, the strength I gained and the person I have become, I owe to many who have helped me along the way. I am particularly grateful to my family, who have always stood by me through thick and thin; all my teachers, from whom I have learnt invaluable lessons; my friends who helped me become a better person; and colleagues who went out of their way to help me.

Keshav Kolluru
July 2023

Abstract

Open Information Extraction (Open IE) aims to extract semi-structured information from natural language text in a domain-independent fashion. It is formulated as extracting a set of tuples of the form (subject, relation, object) where each of the fields corresponds to a phrase in the text. Compared to 'closed' information extraction based on canonical KGs, it avoids the need for experts to define the ontology and data curators, making it scalable across domains. In this dissertation, we describe novel Open IE systems that take advantage of recent advances in deep neural models to tackle multiple challenges associated with building automated systems for the task of Open IE in both monolingual and multilingual settings. We propose solutions that represent significant advances across multiple axes — (1) design of new models, (2) extension to multiple languages, (3) support for linguistic phenomena, (4) downstream application to Knowledge Bases and (5) release of new systems. In models, we build novel deep learning architectures that establish new state-of-art performance by faithful modelling of the Open IE task with pre-trained language models. We experiment with both sequence-to-sequence generation models (named IMoJIE, Gen2OIE) and sequence labeling models (named IGL, CIGL) for the task. IMoJIE (Iterative Memory-based Joint Open Information Extraction) iteratively re-encodes the sentence along with the extractions generated so far to generate the remaining extractions, ensuring diversity in the extractions. Gen2OIE is a two-stage generative model that first generates all the relations in the sentence, followed by generating extractions corresponding to each relation. The IGL (Iterative Grid Labeling) model labels all the words in the sentence in an iterative fashion with tags dictating their position in the Open IE tuples. CIGL improves over IGL by adding constraints in training to increase the coverage of the extractions. In multilinguality, to enable extension of Open IE to other languages, we need training data in the respective language. Therefore, we build a pipeline for translating English Open IE training data and generating high-quality data in Spanish, Portuguese, Chinese, Hindi and Telugu. In linguistic phenomena, noticing that current Open IE systems lack in properly handling certain linguistic phenomena such as noun compounds and conjunctions, we develop systems for noun compound interpretation and coordination analysis which are incorporated into Open IE systems. In applications of Open IE extractions, we build a multilingual fact linking benchmark and model for connecting textual extractions to their knowledge bases while accounting for facts that can exist in multiple languages. In another application, we advance the state of art in Open Knowledge Base Completion by using a two-stage entity-aware pipeline to infer new triples. Finally in systems, we release the OpenIE-6 system that represents the cutting-edge in the line of Open IE software packages.

सार

खुली सूचना निष्कर्षण (ओपन आईई) का उद्देश्य प्राकृतिक भाषा में लिखा हुआ पाठ से क्षेत्र-स्वतंत्र रूप में अर्ध-संरचित जानकारी निकालना है। ओपन आईई (विषय, संबंध, वस्तु) के सेट निकालने के रूप में परिभाषित किया गया है, जहां प्रत्येक फ़ील्ड पाठ में एक वाक्यांश से मेल खाती है। कैनोनिकल केजी पर आधारित 'बंद' सूचना निष्कर्षण की तुलना में, यह ऑन्तोलॉजी और डेटा क्यूरेटर्स की आवश्यकता से बचाता है, जिससे यह सभी क्षेत्र में काम कर लेता है। इस शोध प्रबंध में, हम नए ओपन आईई सिस्टम्स का वर्णन करते हैं जो एकभाषी और बहुभाषी दोनों ही सेटिंग में ओपन आईई के कार्य के लिए ऑटोमेटेड सिस्टम बनाता है। हम ऐसे समाधान प्रस्तावित करते हैं जो कई अक्षों में महत्वपूर्ण प्रगति का प्रतिनिधित्व करते हैं --- (1) नए मॉडलों का डिज़ाइन, (2) कई भाषाओं में विस्तार, (3) भाषाई घटनाओं के लिए समर्थन, (4) नॉलेज बेस के डाउनस्ट्रीम अनुप्रयोग और (5) नए सिस्टम की रिलीज। मॉडलों में, हम नए डीप लर्निंग आर्किटेक्चर बनाते हैं जो प्री-ट्रेनिंग लैंग्वेज मॉडल के साथ ओपन आईई टास्क का ईमानदारी से मॉडलिंग करके नए अत्याधुनिक प्रदर्शन स्थापित करते हैं। हम इस काम के लिए सीक्वेंस-टू-सीक्वेंस जेनरेशन मॉडल (आईएमओजेई, जेन2ओईई नाम) और सीक्वेंस लेबलिंग मॉडल (आईजीएल, सीआईजीएल नाम) दोनों के साथ प्रयोग करते हैं। इमोजी (इटरेटिव मेमोरी-आधारित ज्वाइंट ओपन इंफॉर्मेशन एक्सट्रैक्शन) शेष निष्कर्षण उत्पन्न करने के लिए अब तक उत्पन्न निष्कर्षण के साथ वाक्य को जोड़ता है, जिससे निष्कर्षण में विविधता सुनिश्चित होती है। जेन2ओईई एक दो-स्टेज वाला जेनरेटिव मॉडल है जो पहले वाक्य में सभी संबंध उत्पन्न करता है, उसके बाद हर संबंध के अनुरूप निष्कर्षण उत्पन्न करता है। आईजीएल (इटरेटिव ग्रिड लेबलिंग) मॉडल वाक्य में सभी शब्दों को ओपन आईई टैग के साथ पुनरावृत्त फैशन में लेबल करता है। सीआईजीएल निष्कर्षण का कवरेज बढ़ाने के लिए ट्रेनिंग में प्रतिबंध लगाकर आईजीएल पर सुधार करता है। ओपन आईई के निष्कर्षण को अन्य भाषाओं में सक्षम करने के लिए, हमें संबंधित भाषा में ट्रेनिंग डेटा चाहिए. इसलिए, इंग्लिश ओपन आईई ट्रेनिंग डेटा का अनुवाद करके स्पेनिश, पुर्तगाली, चीनी, हिंदी और तेलुगु में उच्च क्वालिटी का डेटा उत्पन्न करने के लिए हम पाइपलाइन बनाते हैं। भाषाई घटनाओं में, इस बात पर ध्यान देते हुए कि मौजूदा ओपन आईई सिस्टम कुछ भाषाई घटनाओं जैसे कि नोन कंपाउंड और कंजंक्शन को सही तरीके से संभालने में कमी है, हम नोन कंपाउंड व्याख्या और समन्वय विश्लेषण के लिए सिस्टम विकसित करते हैं जो ओपन आईई सिस्टम में शामिल किया जाता है। ओपन आईई निष्कर्षण के अनुप्रयोगों में, हम एक बहुभाषी तथ्य लिंकिंग बेंचमार्क और मॉडल बनाते हैं, जो कई भाषाओं में मौजूद तथ्यों का हिसाब करते हुए उनके ज्ञान के आधार पर नॉलेज बेस और निष्कर्षण को जोड़ सकता है। एक अन्य अनुप्रयोग में, हम नए निष्कर्षण का अनुमान लगाने के लिए पाइपलाइन का उपयोग करके ओपन नॉलेज बेस कंपलेशेशन में अत्याधुनिक तकनीक को आगे बढ़ाते हैं। अंततः सिस्टम में, हम 'ओपनआईई-६' सिस्टम जारी करते हैं जो 'ओपन आईई' सॉफ्टवेयर पैकेजों की श्रेणी में अत्याधुनिक का प्रतिनिधित्व करता है।

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of World Wide Web brought along with it an explosion of information available on the Internet, giving rise to the so-called, "*Information Age*". However, much of this information is unorganized due to the development of the Internet in a decenteralized manner from its inception and preference of content writers to use natural language. Each source on the Internet chooses their own unique way to express information, which may vary right from the media used (e.g., video, audio, text, tables, etc. or any combination of them) to the presentation style (such as markup and encoding formats used). Even within the scope of text, there exists a broad spectrum of content expression style, ranging from the natural language used (such as English, Spanish, and so on) to writing style. These choices may be dependent on factors such as background or cultural norms of the creator and the intended audience.

Proper organization of this vast textual content can help convert *data* into *information*, that can be used effectively by both humans and machines. This has traditionally been a primary goal for both the Information Extraction (IE) and Information Retrieval (IR) communities. They have focussed on different startegies for solving the problem. While the IR community tackles it by building better ways to retrieve documents relevant to a user query, the NLP community has taken a more fine-grained approach of assigning structure to the information latent in text. Knowledge Bases (KBs) are a popular way to store such structured information in a consistent manner. Traditional KBs provide an understanding of the textual content by summarizing them as fact tuples, which often take the form of triples which represent a relationship between a head entity and a tail entity in the format (head entity, relation/predicate, tail entity). Each such predicate comes from a pre-defined canonical list that is curated by ontology experts. Moreover, any extracted triple in typical IE settings must be disambiguated and presented in terms of canonical entity and relation ids. However, this severely limits the rate of growth and coverage of the constructed KBs. Open Information Extraction (Open IE) was introduced to obviate excessive canonicalization and thus improve coverage.

Open Information Extraction (Open IE) (Banko et al., 2007; Mausam, 2016), is a general purpose, domain agnostic Information Extraction (IE) paradigm that has been designed to handle the scale and variety of text in the Internet age. Unlike Knowledge Base curation (Suchanek et al., 2007; Auer et al., 2007; Bollacker et al., 2008), which relies on human-built ontologies to provide a framework for the types of information extracted, Open IE has been designed to operate in an entirely ontology-free manner. To achieve this, Open IE uses possibly mildly edited spans from the input text itself in order to extract the information that is expressed in text. By using the generally accepted notion of a fact as expressing the binary relation between two entities, an Open IE system aims to extract all possible facts from an input sentence, where each fact is expressed as a tuple containing (subject; relation; object) along with optional fields like

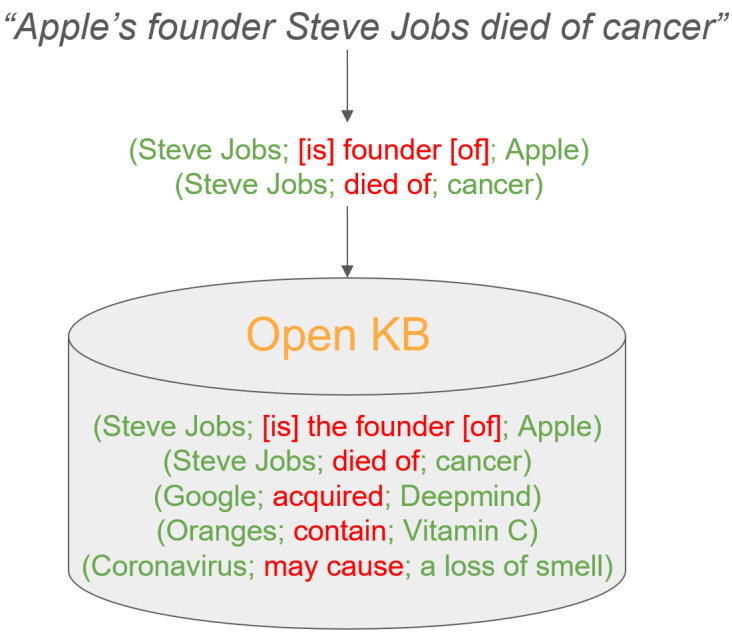

Figure 1.1: Open Information Extraction (Open IE) systems extract tuples of the format (subject, relation, object) from a sentence. A collection of such tuples form an Open Knowledge Base, which can be used as a source of factual information. They provide additional value over using raw-text due to the possibility of aggregating extractions from multiple source sentences via clustering (Fan et al., 2019a).

additional arguments, location, time or context, depending on the Open IE system being used.[1] Since all these phrases are extracted from the input text itself (or using words from a general language vocabulary, in the case of relations not explicitly mentioned in the text), no ontology is needed for expressing the information as facts.

An example is shown in Figure 1.1, where the following two tuples are generated the sentence "Apple's founder Steve Jobs died of cancer" - (Steve Jobs; [is] founder [of]; Apple) and (Steve Jobs; died of; cancer). The square brackets indicate words that are not present in the original sentence and have been added to make the extraction well-formed.

1.1 Semi-structured nature of Open IE

In the NLP literature, various schemes have been proposed to introduce structure into natural language text, often converting it to some form of text-annotated graphs. Converting unstructured text to structured forms has both linguistic and computational advantages. From a linguistic standpoint, these structures can help give a better picture for understanding the meaning of the sentence. From a computational standpoint, they provide additional features or make hidden relations explicit, making it easier for downstream tasks. Some popular examples of structured representations are explained as follows:

- Constituency parses (Younger, 1967; Aho and Ullman, 1973) extract tree-style parse structures that are based on context-free grammers.

[1] In limited cases, even the object phrase is treated as optional, such as, (Ram; sings well;).

- Dependency parses (Mel'cuk et al., 1988; Nivre et al., 2016) find the lexical relation between pairs of words from a set of Universal Dependency tags.
- Semantic Parsing represent text as logical programs using various formalisms such as λ-DCS (Liang, 2013), Answer Set Programming (Baral, 2003), etc.
- Semantic Role Labeling (SRL) (Fillmore, 1985; Carreras and Màrquez, 2005; He et al., 2017) extracts the predicate role and various argument roles.
- Question-Answering guided Semantic Role Labeling (QA-SRL) (He et al., 2015) enables easily generating large scale annotated datasets for SRL using a question and answer framework for identifying the various semantic roles.
- Abstract Meaning Representations (AMR) (Banarescu et al., 2013) uses annotated graphs to express the semantic meaning of sentences.

Open IE chooses a representation where the text is represented as set of tuples. However, it is semi-structured in nature. The tuples are made up of textual phrases that are not often rigourously defined and only have some guiding principles (Stanovsky and Dagan, 2016). This flexibility allows it to capture a richer variety of phenomena compared to other structured tasks like semantic role labeling. For example, while SRL can use only single-word predicates, Open IE often has multi-word predicates which can capture richer relations such as *refused to visit* or *took advantage of*. On the other hand, it also lacks the richer information in SRL, which denotes the exact link between each argument and the relation.

1.2 Relevance of Open IE

Owing to the domain-agnostic nature of Open IE, combined with the simplicity of its design, Open IE has found use in a wide variety of NLP applications like question answering (Khot et al., 2017; Yan et al., 2018), multi-document summarization (Ernst et al., 2021, 2022; Christensen et al., 2014; Fan et al., 2019a), word embeddings (Stanovsky et al., 2015), event schema induction (Balasubramanian et al., 2013) and fact salience (Ponza et al., 2018).

Apart from established uses, Open IE holds a lot of potential in the deep learning era as a rich source of factual knowledge. Explicit reasoning over such factual knowledge stores has several advantages over implicitly storing the knowledge in the parameters of neural language models. Implicit knowledge is hard to change because any change in the knowledge may have non-local consequences. This requires re-training the entire model. Such changes are essential to support due to the '*ephemeral*' nature of many real-world facts. For example, the head of a country is subject to change every few years or new topics of interest may arise with time, such as COVID-19 from the year 2020 onwards. Supporting these additions or deletions of knowledge is more straightforward when we use explicit, readable knowledge stores where the corresponding facts can be added or deleted. Moreover, implicitly storing knowledge in parameters requires increasing the scale of models to support the wide variety of information available. Hence, explicit knowledge stores can help in reducing the model sizes as well (Hoffmann et al., 2022; Rae et al., 2021).

Current knowledge stores often use raw-text itself (Borgeaud et al., 2022) or rely on WikiData (Verga et al., 2021) as the source of factual knowledge. However, due to the unstructured nature of raw text, it is often difficult to control the facts the model uses for reasoning. Moreover, with WikiData, the coverage of facts is highly dependent on the existence of a high-quality

ontology for the particular domain. Open IE tuples provide a promising alternative because the tuples represent knowledge in a domain-agnostic manner, while using the familiar SRO-format (**S**ubject, **R**elation, **O**bject). By searching for tuples that contain the entity in the subject or object position, the facts can be filtered accordingly.

Open IE tuples can also be treated as a factual summary of the text. This can help reduce the size of the textual corpora, which may be used for pre-training language models or as a knowledge source in open-domain question answering. Thus, Open IE is a form of text-corpus distillation, which can reduce the training times or the memory consumed, in the same vein as standard techniques of model distillation that help reduce the size of the neural models.

Moreover, Open IE is even useful as a user-facing representation for its ease of human understandability, especially compared to other structured representations, such as semantic role labeling, which require a detailed understanding of the various role definitions. Open IE tuples often read as a factual sentence, with delimiters typically separating the tuple into clearly-defined relation phrases and argument phrases.

1.3 Thesis Contributions

Approaches to Open IE used to be dominated by statistical and rule-based systems. Successful systems built for the task (Etzioni et al., 2011; Christensen et al., 2011; Mausam et al., 2012; Del Corro and Gemulla, 2013; Pal and Mausam, 2016; Saha et al., 2017; Saha and Mausam, 2018) have relied on a mix of linguistic rules and statistical pipelines. Having been developed before the deep learning wave that swept NLP and many other fields, they do not benefit from the generalization enabled by neural representations and unsupervised pre-training (Devlin et al., 2019; Raffel et al., 2020) and supervised fine-tuning (Krizhevsky et al., 2012) due to the lack of good quality training datasets. In this thesis, we work on enabling the deep learning ecosystem in the context of Open IE.

Moreover, Open IE, like many other NLP tasks, has been predominantly developed for high-resource languages like English, although the task definition imposes no such restrictions. Due to the open nature of the task definition, it can be applied as it is to other languages as well. Because early generations of Open IE systems often used language-specific insights, development of systems was limited to popular languages which had support in the Open IE research community. However, recent years have seen an increasing focus on multilingual NLP (Liu et al., 2020b; Xue et al., 2020). This is due to the recognition of the importance of supporting a wider population of the globe, along with the ease of using language-agnostic embeddings for easily extending NLP support to low-resourced languages. In this thesis, we attempt to bring this trend to the task of Open IE by idenfitying the critical challenges and proposing techniques for solving them.

This thesis contributes to several parts of the Open IE, ranging from:

Models: Building better neural models specifically designed for the task.

Linguistic Phenomena: Enabling proper handling of coordinations and noun compounds in Open IE extractions.

Multilinguality: Constructing strong translation pipelines for generating training data in other languages.

Applications: Investigating the application of extracted triples in the context of knowledge bases.

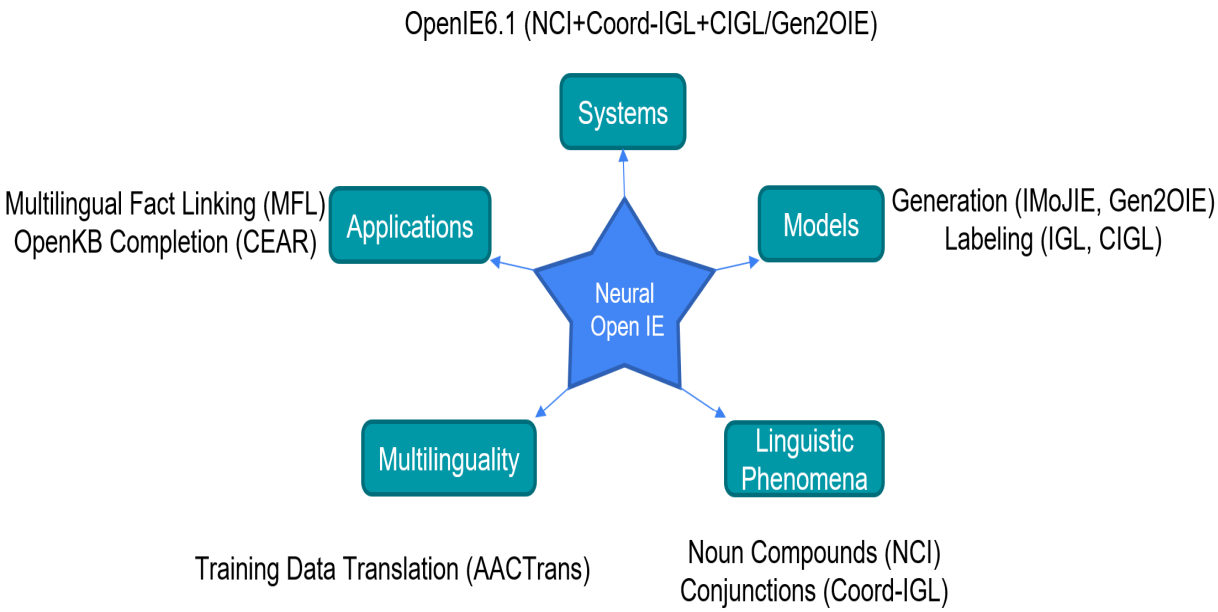

Figure 1.2: Schematic of the overall contributions of the thesis. We introduce new Open IE models, which are generation-based (IMoJIE, Gen2OIE) in Chapter 3 and labeling-based (IGL, CIGL) in Chapter 4.We handle special linguistic phenomena in Open IE extractions such as noun compounds (NCI) coordination analysis (Coord-IGL) in Chapter 5. We extend Open IE to other languages by creating a novel training data translation technique (AACTrans) in Chapter 6. We use Open IE in downstream applications of multilingual fact linking (MFL) and Open KB completion (CEAR) in Chapter 7.

Systems: Releasing new Open IE systems that combine various functionalities developed in this thesis.

The schematic of the overall contributions is shown in Figure 1.2. In the remaining part of the introduction, we discuss these contributions further.

1.3.1 Models

A primary focus of this thesis is on progressively building strong neural models that establish a new state of the art in performance on the task of English Open IE while balancing for fast inference times. Extraction Open IE tuples can be posed as either a labelling task (Stanovsky et al., 2018) or a generation task (Cui et al., 2018). In the labeling framework, each word of the sentence is annotated with a tag indicating its role in a tuple as part of either the subject, relation, object or absent from the tuple. Alternatively, the generative framework generates a tuple word by word using a sequence-to-sequence model. Special delimiters are output to separate the different parts of the tuple. Generative modeling represents a more powerful representation for the task as it can introduce new words in the extraction that are absent in the original sentence. This comes at the cost of increased computation due to auto-regressive nature of generative models. For example, generating the extraction (Steve Jobs; [is] founder [of]; Apple) from the sentence, "Apple's founder Steve Jobs attended Reed College," requires generating two additional words, 'is' and 'of' in the relation.

Therefore, we explore both the types of modeling while building neural models, with IMoJIE (Kolluru et al., 2020b) and Gen2OIE (Kolluru et al., 2022a) being sequence generation models

while IGL and CIGL (Kolluru et al., 2020a) being sequence labeling models. The sequence generation models are covered in Chapter 3, and the labeling models are covered in Chapter 4.

1.3.1.1 Generation models

IMoJIE: IMoJIE is an iterative sequence generation model that relies only on a pre-trained encoder while allowing the decoder to be randomly-initialized at the beginning of training the model. This enables the use of encoder-only pretrained models (like BERT (Devlin et al., 2019), MuRIL (Khanuja et al., 2021), etc.) for the task of generative Open IE. It repeatedly appends the generated extraction with the input sentence and the remaining extractions generated so far, using a special separator token, *[SEP]*, that helps to demarcate the different concatenated inputs. This process converts a set of sequence generation problem (set of Open IE extractions) into an iterative sequence generation approach (generating Open IE tuples, one extraction at a time). IMoJIE prevents generation of redundant extractions that plagued previous neural generative Open IE systems (Cui et al., 2018). This leads to an overall superior quality of generated extractions, improving upon the previously published state-of-art model by 50.28% due to the faithful modeling of the task. At the time of publication, the model represented the first neural Open IE model to makes use of advances in semi-supervised pre-training such as BERT.

Since Open IE lacks a large scale corpus of manually annotated training data, the training data for IMoJIE was bootstrapped from extractions generated by prior high quality Open IE systems like OpenIE-4 (Christensen et al., 2011; Pal and Mausam, 2016). Training on this data enables the neural models to learn by imitating previous systems. Neural models are able to correct some of the errors due to the power of neural generalization that takes advantage of the vast amount of knowledge gained during the pre-training phase.

The IMoJIE architecture is described in Section 3.1.

Gen2OIE: In our next iteration of generative models, we develop the Gen2OIE model, which represents the current state-of-art Open IE model. Extending Ro et al. (2020), Gen2OIE uses a two-stage pipeline to first generate all relations present in the sentence using a Stage-1 Sequence-to-Sequence (Seq2Seq) model and then a Stage-2 Seq2Seq model to generate all the extractions containing the specific relation. This two-stage approach allows independent optimization of each module, enabling the addition of a Relation Coverage (RC) heuristic that improves the coverage of generated extractions in a language-agnostic manner. The model achieves a new state-of-art performance in a total of six languages, including English. The Gen2OIE architecture is described in Section 3.2.

1.3.1.2 Labeling models

IGL: We also developed the Iterative Grid Labeling (IGL) architecture that iteratively labels each extraction as a sequence of subject, relation or object tags over the sentence words. The iterative nature of the algorithm allows for capturing dependencies between the extraction, while the architectural choices of word-level tagging enables an extremely fast model compared to sequence generation models that outputs each word of the extraction in an autoregressive fashion. This results in a speedup of as much as 25$\times$ in terms of number of sentences processed per second, compared to the IMoJIE model. The IGL architecture is described in Section 4.1.

CIGL: The speedup using IGL architecture is accompanied by a modest decrease in performance. Therefore, we introduce constrained training of the model to balance this. The con-

strained training increases the relation coverage by ensuring that certain words, such as verbs, are always labelled as relation in at least one of the generated extractions. Since the constraints are only imposed during training and do not effect the inference speed of the model, it results in a significantly fast system with performance matching that of generative models. The CIGL architecture is described in Section 4.2.

1.3.2 Linguistic Phenomena

We extend the capability of Open IE systems to handle two types of linguistic phenomena — (1) conjunctions and (2) proper noun compounds. These are further discussed in Chapter 5.

1.3.2.1 Conjunctions

To handle conjunctions appropriately, we need to resolve the coordination structure ambiguity (Ficler and Goldberg, 2016a; Saha and Mausam, 2018) where the exact boundaries of each of the conjuncts should be identified. For example, the sentence "Jeff Bezos founded Amazon and Blue Origin and invested in Google, Grail and ZocDoc" should correctly identify the conjuncts associated with the two conjunctions – Amazon, Blue Origin for the first "and", and Google Grail, ZocDoc for the second "and" in the sentence. This allows generation of correct Open IE extractions such as (Jeff Bezos; founded; Amazon), (Jeff Bezos; invested in; Google), etc.

By identifying that the problem can also be posed as an iterative labeling scheme, we use the Iterative Grid Labeling (IGL) architecture from Section 4.1 and use it for identifying the (possibly nested) coordination structure elements. The IGL scheme results in a gain of 12.3% F1 compared to prior coordination analysis models (Teranishi et al., 2019). Once correctly identified, we use heuristics to split the sentence into multiple sub-sentences, each of them involving individual conjuncts. Open IE then generates the final set of extractions by aggregating extractions generated from each of these sub-sentences. Properly handling conjunctions results in a gain of 3.6% F1 in final performance. Handling conjunctions for Open IE are further described in Section 5.1.

1.3.2.2 Proper Noun Compounds

Proper Noun Compounds (PNCs) often express implicit relations that are currently ignored by Open IE systems. For example, a "Covid vaccine" is a "vaccine that *immunizes against* the Covid *disease*". Adding this understanding helps generate better extractions. In the sentence, "Researchers at Oxford successfully developed a Covid vaccine", the following extraction can be generated (Researchers at Oxford; successfully developed a vaccine that immunizes against; the Covid disease). In absence of the interpretation, the extraction would be limited to (Researchers at Oxford; succcessfuly developed; a Covid vaccine). We collect a new dataset called ProNCI (Kolluru et al., 2022b) that contains 25K proper noun compounds (first word is a proper noun) and their manually annotated semantic interpretations. We train generative models for the task of interpretation and integration into the sentence. The Open IE extractions with proper compound noun at the start of an object are integrated with the compound noun interpretation and post-processed to get the final set of extractions. This results in an increased yield of 7.5%, where the added extractions have a precision of 92%. Adding proper noun compound interpretations to Open IE is further described in Section 5.2.

1.3.3 Multilinguality

The primary challenge in extending Open IE to multiple languages is the lack of good-quality training data in the respective languages. The data which is used for training neural models is quintessential for determining the quality of the final system. This is a particularly important factor in the case of Open IE, which is a "resource-poor" task and lacks access to a high quality training corpus. The current state is likely due to the facts that at its inception, Open IE was pitched as a completely unsupervised paradigm, because using training data was regarded as critical requirement for making the system domain-specific (Banko et al., 2007). However, this may no longer a constraint for neural models that can exhibit more substantial out-of-domain generalization than prior statistical/symbolic systems.

The lack of high quality training data is particularly acute in languages other than English, which do not often have strong Open IE systems that can be used for bootstrapping neural systems. Hence, neural Open IE systems have been developed primarily for English. To overcome this problem, we devise a novel translation technique called AACTrans (Kolluru et al., 2022a). It translates Open IE examples from English to other languages. The AACTrans methodology ensures that the sentence and extractions are translated in a consistent fashion, avoiding any lexical or semantic/syntactic inconsistencies that may result from using independent translations of the sentence and extractions. We find that this style of translation consistently outperforms independent translations. When we test this technique on five languages — Spanish, Portuguese, Hindi, Telugu and Chinese — we find that translating the data using AACTrans results in a consistent improvements with as much as 1.7% F1 in Spanish and 1.3% F1 in Telugu. Moreover, the final systems are significantly better than previous state-of-art zero-shot system, Multi2OIE (Ro et al., 2020), by a large margin of 6-25% F1. The AACTrans technique is further described in Section 6.1.

1.3.4 Applications

Open IE triples are helpful in constructing Open KBs (Gashteovski et al., 2019; Galárraga et al., 2014), that can be used as a source of general knowledge in various knowledge-seeking applications. As part of this thesis, we worked on inferring new facts in Open KBs using a novel Knowledge Base Completion (KBC) method called Cross-Entity Aware Re-ranking (CEAR) (Kolluru et al., 2021a). The CEAR model uses a two-stage architecture that re-ranks top-k predictions of an existing Knowledge Graph Embedding (KGE) model. This architecture establishes a new state of the art in a publicly available Open KB dataset - OLPBench (Broscheit et al., 2020), improving performance by as much as 5.3% HITS@1, compared to prior systems. The CEAR model is described in Section 7.2.

We further explore how to align natural language sentences or Open IE triples with standard Knowledge Bases such as WikiData[2] that contain ontologically-grounded triples. The goal of developing the system was to further handle language disparities in both the input and the facts that have to be linked, as facts in WikiData are often biased towards high resource languages like English while the input queries may come from any language (Kaffee et al., 2017). For this purpose, we release a new testing dataset and model architecture for the task Multilingual Fact Linking (MFL) (Kolluru et al., 2021b) in English and six Indian languages. MFL is described in Section 7.1.

[2]https://www.wikidata.org/wiki/Wikidata:Main_Page

1.3.5 Systems

As part of the thesis, we released a new Open IE system which we call OpenIE6 (Kolluru et al., 2020a), which represents the latest in the line of Open IE systems, OpenIE-4 (Christensen et al., 2011; Pal and Mausam, 2016) and OpenIE-5 (Saha et al., 2017; Saha and Mausam, 2018). It uses the CIGL labeling architecture for generating Open IE extractions and the IGL architecture for handling examples with coordination structures. This system has been released at https://github.com/dair-iitd/openie6 and has been downloaded more than 2K times at the time of writing this dissertation.

In an updated version of the system, called OpenIE6.1, we release a repository which contains code for all three systems (IMoJIE, Gen2OIE and CIGL) in a single framework, along with handling coordinations and noun compounds, as dictated by user inputs. This also allows the user to optimize between accuracy and speed, as needed. We describe this final system in Chapter 5. All of the individual code bases have also been released, with the links given in the respective chapters.

1.4 Thesis Outline

The remaining part of the thesis is organized as follows:

(i) Chapter 2 gives a detailed overview of various Open IE systems proposed so far and their applications,

(ii) Chapter 3 describes generative approaches to Open IE, discussing the IMoJIE and Gen2OIE neural models,

(iii) Chapter 4 describes labeling approaches to Open IE, discussing the CIGL neural model,

(iv) Chapter 5 describes how to handle linguistic features such as coordinations and proper noun compounds,

(v) Chapter 6 describes the efforts in developing Open IE for multiple languages,

(vi) Chapter 7 describes two applications where the Open IE triples are used in context of Knowledge Graphs, and

(vii) Chapter 8 describes the final conclusions of the thesis and potential research problems that remain open in this area.

Chapter 2

Related Work

In this chapter, we first define the task of Open Information Extraction (Section 2.1) and discuss some of the existing systems built for the task. We classify the existing systems into two categories – those which are built primarily for the English language (Section 2.3) and those which are built for other languages (Section 2.4). Within each of the two categories, we divide it further into two sub-groups of models – the previous generation syntactic/statistical models and the more recent deep learning models. We also look at the evaluation of Open IE systems (Section 2.2), applications of Open IE (Section 2.5) and other tasks related to Open IE (Section 2.6).

There have been multiple efforts in the past to build a comprehensive summary of Open IE systems. Pai et al. (2022) present a recent survey of methods used in Open IE, with a significant focus on neural systems. Mausam (2016) also presents a comprehensive summary of Open IE, including the downstream applications of Open IE. Downstream applications include event schema induction, traditional information extraction, text comprehension, sentence similarity (summarization), lexical similarity and analogy tasks. Niklaus et al. (2018) also covers non-neural Open IE systems. The current chapter builds on these surveys by highlighting the aspects relevant to understanding the various methods proposed in the dissertation.

2.1 Task Definition

The task of Open Information Extraction (Open IE) aims to extract all possible *relational tuples* from text, which may be a single sentence or a collection of sentences in a document. Each tuple primarily conveys a relation expressed in text, often involving a subject and an object, along with possibly additional information that describes the relation under consideration. In general, each tuple (also referred to as an *extraction*) has the format (*subject*; *relation*; *object*; *optional arguments*), where the optional arguments may include remaining arguments for an *n-ary* relation or have attributes such as location, time or context which add further information about the relation. Each field in the tuple can be nouns, clauses or even complete sentences.

Each field of the tuple uses phrases directly from the text. This choice ensures that relevant tuples can be extracted from as wide a corpus as the Web.

There is some allowance for the use of additional words as sparingly as possible to ensure that the fields in a tuple can be made easier to understand. For example, from "U.S. president Joe Biden", the extraction (Joe Biden; [is] president [of], U.S.) needs addition two words, *is* and *of*, to form a grammatical relation. For the case of optional arguments, consider the example sentence "U.S. president Joe Biden inaugrated the building at Washington in 2022". The extraction (U.S. president Joe Biden; inaugrated; the building; T: in 2022, L: at Washington)

containing the additional fields of location (L) and time (T) tags provides valuable information to the end user/application. To ensure domain independence, a tuple is not defined semantically rigorously, but is left to interpretation of system implementations. For example, in the case of prepositions that occur in the field boundaries, as shown in the above example, we prefer to include them in the relation field by default. However, when optional arguments are present, such as location/time, then they are kept as part of the argument itself.

2.2 Evaluation

Evaluation of Open IE extractions poses a unique challenge from both a linguistic and a computational standpoint. The linguistic definition of Open IE has been left open-ended in the literature, with the objective of allowing the handling of a wide variety of domains. However, this leads to each system choosing its own implementation strategy, which makes it challenging to compare extractions from multiple systems due to the varying representational choices. For example, some systems choose to output additional arguments that represent the location, time and context associated with the extractions, and some other systems use nested representations to represent complex extractions. Even after developing reasonably standard guidelines for Open IE and annotating gold extractions according to it, the computational challenge of evaluating the predictions of a system is non-trivial. This is due to the nature of the task, which requires matching of a gold set of tuples/extractions with a variable number of predicted tuples. It can lead to complications such as having similar content being expressed in multiple extractions, thus necessitating penalization of near-redundancy. A broad variety of evaluation schemes and benchmarks have been proposed in literature over the years, which is summarized in this section.

Early works have often relied on manual evaluation to determine the precision of the system, often ignoring recall due to the challenge in annotating the complete set of valid extractions. For example, Mausam et al. (2012); Del Corro and Gemulla (2013); Saha et al. (2017); Gashteovski et al. (2017) use annotators to evaluate the quality of generated extractions without using a reference set of gold extractions. Hence, only the precision and yield (total number of extractions) are used as the performance metrics.

Stanovsky and Dagan (2016) is the first work to recognize the challenges associated with prior Open IE evaluation schemes commonly used in literature, which relied on subjective human-judgements. In particular, they identify a lack of consistent evaluation guidelines as well as a lack of large-scale gold-annotated corpus for automated benchmarking of multiple Open IE systems. As part of the work, they identify three main properties of good Open IE extractions, including assertedness, minimal propositions and completeness, and they create a corpus of 3200 Open IE sentences and their gold extractions using an automated procedure for converting from QA-SRL (He et al., 2015) gold annotations to their corresponding Open IE tuples. In their proposed scheme, commonly referred to as the OIE2016 benchmark,[1] the correctness of an extraction is evaluated by matching the grammatical head of each field with that of the gold. OIE2016 creates a one-to-one mapping between (gold, system) pairs by serializing the extractions and comparing the number of common words within them. Hence the system is not penalized for misidentifying parts of one argument in another.

Precision and recall for the system are computed using the one-to-one mapping obtained, i.e. precision is (no. of system extractions mapped to gold extractions)/ (total no. of system extractions), and recall is (no. of gold extractions mapped to system extractions)/(total no. of gold extractions). These design choices have several implications (Léchelle et al., 2018; Bhardwaj

[1] `https://github.com/gabrielStanovsky/oie-benchmark`

et al., 2019). Overlong system extractions, which are mapped, are not penalized, and extractions with partial coverage of gold extractions, which are not mapped, are not rewarded at all.

Wire57[2] (Léchelle et al., 2018) attempts to tackle the shortcomings of OIE2016. They manualy annotate a relatively small corpus of 57 sentences with their gold tuples. Extractions are written while adhering to the principles of minimality, informativeness, exhaustiveness and allowing light inferences. The exercise results in a total of 347 tuples. They also propose a new function to score system predicted extractions. For each gold extraction, a set of candidate system extractions are chosen on the basis of whether they share at least one word for each of the arguments[3] of the extraction, with the gold. It then creates a one-to-one mapping by greedily matching gold with one of the candidate system extraction on the basis of token-level F1 score. Token level precision and recall of the matches are then aggregated to get the score for the system. Computing scores at token level helps in penalizing overly long extractions.

Wire57 ignores the confidence of extraction and reports just the F1 score (F1 at zero confidence). One way to generate AUC for Wire57 is by obtaining precision and recall scores at various confidence levels by passing a subset of extractions to the scorer. However, due to Wire57's criteria of matching extractions on the basis of F1 score, the recall of the system does not decrease monotonically with increasing confidence, which is the norm when calculating AUC.

OIE2016 and Wire57 both use one-to-one mapping strategy, due to which a system extraction that contains information from multiple gold extractions is unfairly penalized.

To resolve such unfair penalization, CaRB[4] (Bhardwaj et al., 2019) was proposed. CaRB also computes similarity at a token level, but it is slightly more lenient than Wire57 — it considers number of common words in (gold,system) pair for each argument of the extraction. However, it uses one-to-one mapping for precision and many-to-one mapping for computing recall. While this solves the issue of penalizing extractions with information from multiple gold extractions, it inadvertently creates another one — unsatisfactorily evaluating systems that split on conjunctive sentences, such as, "I ate an apple *and* an orange". We explore this in detail in Section 5.1.2.1. Along with the scoring function, they also release a more extensive evaluation set of 1200 sentence extraction pairs that have been annotated using crowdsourcing after carefully training the workers on the Open IE task. Using manual verification, they demonstrate that their dataset and scoring function marks relative performance of systems more accurately compared to prior proposed benchmarks. Hence, we use CaRB as the primary evaluation metric throughout the dissertation.

2.3 Models for English Open IE

2.3.1 Syntactic and Statistical Models

Traditional open extractors are rule-based or statistical, e.g., Textrunner (Banko et al., 2007), ReVerb (Fader et al., 2011; Etzioni et al., 2011), OLLIE (Mausam et al., 2012), Stanford-IE (Angeli et al., 2015), ClausIE (Del Corro and Gemulla, 2013), OpenIE-4 (Christensen et al., 2011; Pal and Mausam, 2016), OpenIE-5 (Saha et al., 2017; Saha and Mausam, 2018), MinIE (Gashteovski et al., 2017) and NestIE (Bhutani et al., 2016). These use syntactic or semantic parsers combined with rules to extract tuples from sentences. We briefly discuss these systems

[2]https://github.com/rali-udem/WiRe57

[3]We refer to *subject*, *relation* and *object* as *arguments* of the extraction.

[4]https://github.com/dair-iitd/CaRB

in this section.

2.3.1.1 TextRunner

Banko et al. (2007) is the first system designed for Open Information Extraction and defines some of the important features required for this task, as conceived then:

- It must have ideally no supervision or a minimum amount of supervision, in-order to keep its domain independence.
- It must be fast to scale to the magnitude of information available on the Web. Hence, it must use shallow features.

The system is designed to operate in a self-supervised fashion. It generates labels for automatically parsed sentences based on certain heuristics defined. Then using unlexicalized features from the sentences, it learns a Naive Bayes classifier to predict the valid relations in the sentence. It uses noun chunks to come up with argument phrases and then iterates over possible relation phrases, passing it through the classifier to determine the confidence of the extracted tuple.

2.3.1.2 ReVerb

ReVerb (Fader et al., 2011) improves on the previous Open IE systems, such as TextRunner, by using a very simple heuristic rule that matches relations phrases with POS patterns. These POS patterns are shown to cover up to 85% of naturally occurring verb-based relations in the text. In addition to these syntactic patterns, a lexical constraint is introduced to avoid overly-specific relations permitted by the syntactic pattern. The lexical constraint is to consider only relations less than a pre-specified maximum length.

In order to assign confidence to each extraction, a logistic regression model is trained on 1000 extractions which are manually labelled. The features for the logistic regression are relation independent and shallow to allow for scale and domain-independence.

2.3.1.3 OLLIE

OLLIE (Mausam et al., 2012), Open Language Learning for Information Extraction, is an Open IE system that uses bootstrapped data from high-quality extractions of the ReVerb system to learn extractions from dependency parse trees. The improvements over the previous Open IE systems is to include noun-mediated relations in its extractions and provide further contextual information about the extraction, which may, in fact be a supposition, a held-belief, or conditionally asserted, provided the occurrence of future events.

OLLIE learns extractions using deeper features, such as dependency parse trees, compared to previous systems like ReVerb, which use only shallow PoS tags. It learns pattern templates over the dependency parse trees, using training data generated using bootstrapping. ReVerb extractions are taken from a large corpus, such as ClueWeb (Callan et al., 2009), and only highly confident extractions are retained. The set of extractions is further refined by removing those which occur less than twice in the entire dataset.

These form a set of seed tuples which is used for bootstrapping the training data. All sentences which have the keywords present in the tuples are retrieved and filtered using constraints over the dependency parse trees. All the sentences retrieved for extraction are assumed to have

it as the correct extraction. Using this noisy training data, patterns over the dependency parse trees are learnt to produce extractions.

At test time, given a new sentence, it's parse tree is matched against the learnt pattern templates and based on the slots in the pattern template and their corresponding value in the sentence's dependency parse tree, the extraction is generated. The generated extractions are then passed through a contextual analysis step which generates the additional context (based on certain edges of the dependency parse tree), under which the fact the can be asserted as true.

2.3.1.4 StanfordIE

Most of the previous Open IE extractors are based on learning patterns, either on shallow features such as POS tags (Banko et al., 2007) or on deeper features (Mausam et al., 2012) such as dependency parse trees. The current paper takes a different approach by first extracting clauses (which are syntactically and semantically independent) entailed by the original sentence. Then using Natural Logic (Valencia, 1991), these clauses are reduced to minimalistic ones in order to be more valuable in downstream applications. Once these set of short, independent clauses are extracted, a small set of hand-written rules over the dependency parse trees are used to get the Open IE extractions.

Dependency-tree edge classification is used to extract the clauses. Once independent clauses are extracted, they are minimized to be of maximum utility. The expectation is that this minimization provides a simplistic way for normalizing Open IE extractions, where the downstream applications (such as Question Answering or Relation Extraction) can then easily identify similar extractions. Since these extracted clauses are typically simple, 14 manually written patterns are sufficient to extract high-quality Open IE extractions from them.

2.3.1.5 ClausIE

Similar to StanfordIE, ClausIE Del Corro and Gemulla (2013) uses the notion of clauses for generating Open IE extractions. It identifies a set of clauses in a input sentence. The grammatical function of the clause is classified based on their constituents. Each clause can be broken down into multiple extractions or prepositions. The postprocessing of clauses to extractions can be done in a application-specific manner. Thus, ClausIE maintains a separate knowledge represention in the form of clauses that is distinct from the final extractions. This feature of ClausIE allows greater flexibility compared to other systems.

The identification of clauses in sentences is done using grammatical rules on dependency parses. Each clause is then classified into seven types based on their consitutents. The clause types are then used for minimizing any additional information in the clause, thus reducing them to a set of minimalistic clauses. With the set of minimal clauses identified, the extractions are generated by initially identifying the constituents that should belong to an extraction. Then the constituents are classified into subject, relation and additional arguments to form the final extraction. The final system generates 2.5-3.5$\times$ the number of correct extractions as OLLIE. They also note that they don't use any sort of global post-processing on top of the generated extractions to filter out some extractions, thus making their system highly parallelizable.

2.3.1.6 OpenIE-4

OpenIE-4 consists of a combination of two systems, SRL-IE (Christensen et al., 2011) and RelNoun (Pal and Mausam, 2016). SRL-IE forms the core web-based information extractor, and RelNoun adds noun-mediated relations to the overall mix.

Christensen et al. (2011) explores the possibility of using very deep semantic features, as found in the task of semantic role labelling for the purpose of generating open information extractions. It uses an existing SRL system and repurposes the output to generate open extractions. It does this by combining the modifier associated with the verb to form the relation, and the various fields associated with the verb form both the arguments. The dependence on semantic role labeling comes at the cost of reduced speed, compared to using shallow features like PoS tags but benefits the system in case of complex extractions.

Pal and Mausam (2016) improves the extractions on noun-mediated relations by using additional lexicons generated from ClueWeb corpus and Wikipedia. It shows that current systems suffer from wrong extractions on compound nouns, as they struggle in cases of organization names, denonyms and compound relational nouns. By semi-automatically curating a set of 160 organization words, 2143 denonym, location enteries and 5,606 common relational noun prefixes, they create a new system, RelNoun 2.2.

2.3.1.7 OpenIE-5

OpenIE-5 builds on top of OpenIE-4 by adding two components for proper handling of coordination structures and numerical facts in the input.

Performance of Open IE systems can be improved by identifying coordinating structures governed by conjunctions (e.g., 'and') and splitting conjunctive extractions. CalmIE (Saha and Mausam, 2018), which is part of OpenIE-5 system, splits a conjunctive sentence into smaller sentences based on detected coordination boundaries, and runs Open IE on these split sentences to increase overall recall. The paper proposes a novel approach to dealing with conjunctions for the task of open information extractions by first parsing the sentence to a tree representation using its dependency parse. Following this, the natural conjuncts are identified, and the sentence is split into simpler sentences using a language model to understand the validity of the generated sentences. The simpler sentences are then passed to Open IE to generate extractions. They name their system CalmIE.

Numerical facts have been ignored so far in state of the art open extractors. Although they may produce some numerical facts, they are not number-aware and hence make many mistakes. Saha et al. (2017) proposes a new system that generates numerical facts using a bootstrapping process similar to the one used by OLLIE. They manually specify a set of seed patterns to extract high precision numerical facts. Using this as the seed tuples, they retrieve additional sentences from the web. After enforcing additional lexical constraints, they assume that the seed tuple is the correct extraction for the retrieved sentence and learn to map dependency patterns to the extractions. This gives a high precision method for extracting numerical facts from raw text.

2.3.1.8 MinIE

MinIE (Gashteovski et al., 2017) has been proposed to increase the compactness of the Open IE extractions. It is built on top of the ClausIE system, which they identify as sufferring from overly-specific extractions. In order to achieve this minimality, they purposefully leave out some information from the actual extraction and annotate that as special fields associated with the extraction. They also use statistical modules to identify aspects of the extractions that are overly specific, either universally or in a domain-specific manner. Once identified, the extractions are re-written after removing them. This results in system outputs that are much more compact while being competitive in precision to other state of the art Open IE systems like OLLIE and ClausIE.

2.3.1.9 NestIE

NestIE (Bhutani et al., 2016) is a rule-based open extractor that aims to address the issue with previous extractors that only produce binary tuples. Binary tuples leads to the inability to express relations in complex sentences that may need tuple nesting. Thus NestIE proposes an approach to generate various tuples which are linked together to get nested tuples. They use hand-written templates to get highly confident seed data of dependency patterns and tuple representations. By bootstrapping from the seed data, they learn patterns over dependency parses hence generate extractions from a wide variety of sentences. NestIE achieves higher informativeness with more minimalness when compared to previous systems.

There exists many more non-neural Open IE systems that have been published over the years. We have only described the ones that are prominent and relevant to the dissertation. In the next section, we discuss the Open IE systems that are built using deep learning techniques.

2.3.2 Deep Learning Models

Traditional Open IE systems discussed in the previous section are either statistical or rule-based. They often consist of pipelines of several components like POS tagging, and syntactic parsing. To bypass error accumulation in such pipelines, recent systems use end-to-end neural models. These neural models are end-to-end becuase they can generate the extractions given the sentence by directly learning the associations between them from the training data. The training data for these models are usually bootstrapped from extractions made by earlier systems. Each system tends to make it's own choice of the training data used, and hence, we describe them along with the model descriptions.

The existing neural Open IE methods belong to two categories: sequence *labeling* and sequence *generation*.

Generation systems generate extractions one word at a time. The generated sequence contains field demarcators, which are used to convert the generated flat sequence into a tuple.

Labeling systems label each word in the sentence as either *S* (Subject), *R* (Relation), *O* (Object) or *N* (None) for each extraction. The final extraction is obtained by collecting labeled spans into different fields and constructing a tuple. Such models are much faster but often less accurate due to lack of explicit dependencies between the labels. All the labels are generated parallely in a non-autoregressive manner.

In principle, generation is more powerful than labeling systems because it can introduce auxiliary words or change the word order as necessary. For example, for producing the extraction (Mary; fought with; John) from the sentence, "Mary and John fought each other" requires changing the word order to bring "fought" into the relation and introducing an additional word "with" to make the extraction grammatical. However, this additional power comes with a significant reduction in speed due to the autoregressive nature of sequence generation models.

Sequence Labeling

Sequence Labeling models have been commonly used for related tasks like Semantic Role Labeling (SRL) (Marcheggiani and Titov, 2017). Hence, sequence labeling paradigm has commonly been used for the task of Open IE as well. The major challenge in the adoption of the sequence labeling paradigm for the task of Open IE is the labeling of multiple extractions that may have overlapping fields. For example, consider the sentence, "Shyam presented a gift to Ram who was overjoyed", which has two extractions (Shyam; presented a gift; to Ram) and (Ram; was; overjoyed). The word "Ram" receives both the subject and object labels in the two extractions.

2.3.2.1 RnnOIE

RnnOIE (Stanovsky et al., 2018) is the first sequence labeling system proposed for the task of Open IE. It first identifies the syntactic heads of relations present in the sentence. It does this by collecting all verbs using PoS tagging and collecting nominalizations (nouns that can act as relations) using Catvar's subcategorization frames (Habash and Dorr, 2003). For every identified predicate head, it then uses sequence labelling to get their arguments. Since multiple extractions are possible for a given predicate head, it identifies all of them using a single BIO tagging scheme. Multiple arguments are marked for the predicate, and all possible combinations of the marked arguments are taken as the final set of extractions corresponding to the predicate. This scheme has the drawback of being incapable of supporting overlapping arguments for the same predicate head. For example, from the sentence "Barack Obama, a former U.S. president was born in Hawaii", RnnOIE cannot generate both the extractions (Barack Obama; was born in; Hawaii) and (Barack Obama, a former U.S. president; was born in; Hawaii) as they both have the same predicate head, "born" and overlapping subjects. It is trained on OIE2016 dataset (Stanovsky and Dagan, 2016), which postprocesses QA-SRL data (He et al., 2015) for Open IE. QA-SRL poses the SRL task as a set of question-answer pairs that helps in acquiring crowd-sourced annotations at scale. They also add to the training set QAMR (Michael et al., 2018), an open variant of QA-SRL, after converting it to Open IE format. Evaluating on the OIE16 benchmark, the BiLSTM-based model demonstrates superior performance to non-neural systems usch as OpenIE-4 and ClausIE.

2.3.2.2 SenseOIE

SenseOIE (t et al., 2019) is another sequence labeling system that improves upon RnnOIE by using the extractions of multiple Open IE systems as input features to the model. They consider the tag assigned to the word in each of the k Open IE systems used and pass them as k embeddings to the model. Along with this, they also add additional features associated with each word, such as embedding of the associated pos-tag, semantic role label and dependency parent and siblings. In order to generate multiple extractions, they make use of a beam search strategy. Instead of labeling in a greedy fashion, they choose the top-k labels at every time step. This leads to k set of extractions where a single word may be assigned to multiple labels, overcoming the challenges associated with RnnOIE. However, it also leads to generation of a fixed number of extractions irrespective of the sentence, and their training requires manually annotated gold extractions, which is not scalable for the task. This restricts SenseOIE from training on a dataset of 3,000 sentences.

2.3.2.3 Iterative Rank-Aware Learning

To measure the confidence of the model for a generated extraction, log-likelihood loss is often used as a measure. But it is not optimal as it leads to a difference in training and testing time behaviour. At training time, the confidence of gold extractions are maximized using log-likelihood loss. But at test time, log-likelihood loss is applied to extractions that may be incorrectly generated by the system. Therefore, Jiang et al. (2020) introduced an iterative rank-aware learning approach to calibrate the confidence of Open IE tuples and make them comparable across sentences. Hence, they use a binary classifier to predict if the extraction is correct or wrong. The score of this binary classifier is thus assigned as the score for the extraction. It is trained using classification loss to boost the confidence of correct tuples and alleviate that of incorrect ones, which are synthetically generated by replacing valid arguments in an extraction with randomly

sampled ones. When the classification loss is added to the model, it results in generation of more accurate extractions. These new extractions are added back to the training data for the next version of the model.

They use sequence-labeling RnnOIE as the base model, but the technique can also be applied on other models. The iterative scheme of adding training data helps improve the model performance by as much as 11% F1 on the OIE16 benchmark.

2.3.2.4 SpanOIE

SpanOIE (Zhan and Zhao, 2020) uses a span selection model, a variant of the sequence labelling paradigm. Firstly, the predicate module finds all the predicate spans in a sentence after classifying all possible spans. Subsequently, the argument module outputs the arguments for the detected predicates by iterating through all possible spans and classifying each one of them. Also, for each span embedding, a predicated embedding is added as an additional feature, based on which the argument role is defined. Since considering all possible spans would be infeasible, they make use of three constraints to limit computations on unlikely spans. These constraints include a limit on the length of arguments and predicate, removing syntactically invalid spans that do not contain any head word in it and spans that overlap with the given predicate span. The constraints are applied only during training and not during inference to avoid missing certain types of spans. However this comes with a corresponding increase in inference time.

2.3.2.5 Systematic Comparison

Hohenecker et al. (2020) performs a detailed comparison of various design choices for building sequence labeling systems. They experiment with differnt types of embedding, encoding and decoding blocks used. The embedding block produces an embedding vector for each of the input tokens. The encoding blocks re-contextualizes the embeddings, and the decoding block generates the sequence of labels. They choose between randomly initialized word-piece embeddings or embeddings given by ALBERT, BiLSTM, CNN or Transformer encoders and LSTM, CRF or MLP decoders. They also introduce a novel training scheme for sequence tagging where the loss corresponding to the "Other" tag is ignored. This is to prevent the loss term from being swamped by "Other" tag, which dominates Open IE tagging, compared to "Subject", "Relation" and "Object" tags. They additionally compare with the SpanOIE formulation in a similar setting.

The paper finds that on the OIE16 benchmark, the labeling-based systems achieves best performance when ALBERT, Transformer and LSTM are used as the embedding, encoding and decoding blocks, respectively.

In summary, we have described many sequence labeling systems that have been proposed in literature. The common factor is the use of labeling paradigm for assigning a label to each word in the sentence. The various systems differ in the way they label multiple extractions. In Chapter 4, we propose a novel labeling architecture, Iterative Grid Labeling (IGL), that treats the problem as labeling a grid in an iterative fashion. On addition of soft constraints, our architecture remains fast while reaching performance levels close to that of sequence generation Open IE systems that are described next.

Sequence Generation

Sequence-labeling based models lack the ability to change the sentence structure or introduce new auxiliary words while uttering predictions. For example, they cannot extract (Trump; is the

President of; US) from "US President Trump" since 'is', 'of' are not in the original sentence. Also, they assume that all words in one field of the extraction would occur in the same order as they are in the original sentence. But this may not be ideal in some cases, as shown previously. On the other hand, sequence-generation models are more general and, in principle, subsume the former type of models. They comes with the added ability to generate words that are not present in the sentence as well as the ability to mutate the sentence structure.

2.3.2.6 CopyAttention

Cui et al. (2018) is the first neural Open IE system that has been developed using the sequence generation paradigm. It contains an LSTM encoder-decoder architecture that is augmented with a copy module and an attention module. Hence, we refer to it as the *CopyAttention* model. The decoder generates the extraction one word at a time, using delimiters to identify the various parts of the tuple. In the task of Open IE, most of the words in the extraction are directly taken from the sentence. Therefore, the added copy module allows the model to copy words from the input directly. Similarly, the attention module allows the decoder to focus on important words in the input at each decoding step. This is required to overcome the limitations of LSTMs which face difficulty in remembering longer contexts. During inference, CopyAttention uses beam search to get the final set of predicted extractions.

The model is trained over bootstrapped data that is generated by running OpenIE-4 on all sentences of Wikipedia with less than 40 words. The final training data consists of 36M (sentence, extraction) pairs.

Since the model uses a fixed-size beam search, it limits the output to a constant number of extractions irrespective of the length of the sentence. Moreover, our analysis shows that CopyAttention extractions severely lack in diversity, as illustrated in Table 3.1. We attribute this observation to the dependence on beam search, which can only ensure that exact duplicates are avoided.

2.3.2.7 MCTS

Liu et al. (2020a) proposes a Monte-Carlo Tree Search (MCTS) based RL formulation to generate Open IE extractions from sentences. The paper introduces a way to model the knowledge extraction task of Open IE as a Markov Decision Process (MDP) by defining appropriate state and action spaces along with a reward function. The action space is defined as the set of words in the vocabulary and special symbols that are added to demarcate the various fields in the facts. The extraction generated so far forms the state space for the MDP. The similarity of the predicted fact with the gold fact is used for computing rewards at the training time. Moreover, a simulator is trained that takes as input the partially generated fact and outputs a reward signal to compute the reward at inference time when gold facts are absent. The tree search is also guided using probabilities from an already trained Seq2Seq-based Open IE model. By relying on tree search, they overcome the issues associated with standard sequence generation by performing a global optimization instead of local greedy optimizations. The paper also proposes a parallelized version of MCTS to improve inference speeds.

2.3.2.8 DocOIE

DocOIE (Dong et al., 2021) proposes a *document-level* generative Open IE system that can produce extractions of a sentence in context of the document it appears in. The paper shows that this allows the system to resolve certain types of parts of speech or syntactic ambiguities that may

prove difficult looking only at a specific sentence. For this purpose, they use a Seq2Seq model with BERT as the encoder and LSTM as the decoder augmented with copy and attention modules similar to Cui et al. (2018). The model takes as input the source sentence and specially marked context sentences. The two types of sentences are differentiated using segment embeddings. The proposed model also differentiates between the bootom and top layers of BERT. The bottom layers of BERT take the combined input, while the top layers take only the source sentence embeddings for generating the final extractions.

The paper also releases a dataset of 800 sentence and expert-annotated extractions along with context sentences based on the document in which the sentence occurs. They choose a sample of 80 patents from transportation and healthcare domain to form the dataset. These 800 examples are released as the DocOIE evaluation set. For the training set, they use OpenIE-4 to generate pseudo labels for bootstrapping the systems. Extractions are generated from 100K sentences picked from the 1200 patents to keep it in the same domain. Context sentences are then provided for each of the examples bootstrapped from OpenIE-4. Since this procedure relies on training data generated from context-independent systems, it fundamentally limits the performance of their approach.

In summary, we discussed various models proposed in literature that autoregressively generating one word of the extraction at a time. In Chapter 3, we introduce two new novel sequence generation models, IMoJIE and Gen2OIE, that outperform the extraction quality of existing generation models.

2.4 Models for Non-English Open IE

Many of the Open IE systems described so far, both non-neural and neural, have been deployed exclusively for English. Open IE systems built for other languages often work only for a single language due to their reliance on language-specific resources. Such a reliance makes it infeasible to develop systems for the plurality of languages in the world due to the cost and effort involved. In this section, we explore the systems that operate on a single non-English language and multilingual approaches. For example, Bassa et al. (2018); Rahat and Talebpour (2018); Romadhony et al. (2018); Guarasci et al. (2020); Papadopoulos et al. (2021) focus on German, Persian, Indonesian, Italian, and Greek, respectively.

We first introduce the language specific Open IE models in this section, followed by multilingual Open IE models that work for multiple languages.

2.4.1 Open IE models for German

Bassa et al. (2018) uses German dependency parser and handwritten rules to build an Open IE system in German languages. They perform a detailed analysis of the difference between German and English languages, which can affect the open extraction in German language. They note that the differences in capitalization, gender, cases, and word order needs to be accounted for in designing German Open IE systems. Their system achieves upto 89% F1 performance on the 506 gold facts manually annotated by two German experts.

2.4.2 Open IE models for Italian

Guarasci et al. (2020) builds an Open IE system for Italian and proposes a gold standard benchmark for evaluating the generated extractions. The benchmark consists of 195 sentences annotated by four native Italian speakers with the corresponding n-ary tuples. Their system, which

builds on behavioural patterns of verbs in Italian while identifying uninformative extractions, achieves a precision of 0.79 and recall of 0.84 on their benchmark.

2.4.3 Open IE models for Greek

PENELOPIE (Papadopoulos et al., 2021) is an Open IE system built for the Greek language. It makes use of a Greek-to-English and English-to-Greek NMT systems to generate extractions for a given sentence. It translates the Greek sentence into English and generates extractions using three Open IE systems, Open IE 5.0, ClausIE and RnnOIE. The generated English extractions are back-translated into Greek language using the Greek-to-English NMT system. Each phrase of the extraction is indpendently translated in their scheme.

2.4.4 Open IE models for Chinese

Neural models like Logician (Sun et al., 2018b) and Orator (Sun et al., 2018a) use language-specific training data that limits their system to Chinese.

2.4.4.1 Logician

Sun et al. (2018b) presents a new format for extracting information from text in an ontology-free, open-domain fashion, called SAOKE format and releases a dataset (called the SAOKE dataset) of 48,000 Chinese language sentences and their extractions in the SAOKE format. They also developed a neural Seq2Seq based model, called Logician, that outperforms competitive baselines on the SAOKE dataset.

The SAOKE format, following the previous Open IE extractors, has the desired properties of completeness, atomicity, compactness and accurateness. It almost always uses words from the sentence directly to express the facts in the sentence, except in few cases where it uses special symbols (to state abbreviations, birth and death dates, descriptions, etc.), and hence aptly named as Symbol Aided Open Knowledge Expression.

Based on a comprehensive analysis, the authors divide the extractions into four preliminary types. Facts may

- express the relation between two entities,
- give the attributes of an entity,
- provide descriptions of entity phrases, and
- state hypernymy or synonymy relations between instances.

The proposed neural model, name Logician, convert sentences into "natural logic" in the form of extractions. Similar to CopyAttention, it uses attention and copy mechanism to copy most of the words from the sentences. Only a limited vocabulary of symbols are generated. The model uses a coverage mechanism to avoid the issue of under-extraction or over-extraction of facts from the sentence. The coverage mechanism also provides a normalization scheme across facts. When including a word in a fact, the model is aware if the word is already included in other facts or not. It also uses a gated-dependency mechanism to include dependency information from the parse trees generated using off-the-shelf syntactic parsers.

The compare their system with competitive baselines, include a SoTA SRL-based extractor which learns sequence based tagging on SAOKE dataset. They show that Logician, a Seq2Seq

approach, outperforms the SRL-based system as it is cognizant of the remaining facts while generation.

2.4.4.2 Orator

The Orator (Sun et al., 2018a) is an open-domain narration system that converts a given set of facts into a consistence sentence that contains all the facts expressed in it. Thus, Orator and Logician operate on dual aspects of the same problem where Logician goes from sentence to facts, and Orator goes from facts to sentence. The paper proposes a reinforcement learning loss that uses this duality to enable both the systems to improve each other. This results in the Logician system generating better quality extractions in Chinese language when trained and evaluated on SAOKE dataset.

2.4.5 Models for multilingual Open IE

Prior literature propose only a few multilingual Open IE systems that target Open IE extractions in multiple languages. Claro et al. (2019) lists various challenges involved with building such multilingual systems. They note that the absence of multilingual benchmarks is one of the key reasons hindering advancement in the field, with current systems evaluated only on one or two languages. Moreover, since the defintion of a relation is left open-ended, it complicates the development of such benchmarks where annotators in different languages may choose varying design choices. Hence this neccessiates a development of an open-standard for creating effective annotations in different languages. The paper also points to the development and utilization of multilingual resources such as machine translation systems and cross-lingual knowledge transfer as key components for building strong multilingual Open IE systems.

We now summarize some of the important multilingual Open IE systems proposed in literature.

2.4.5.1 Cross Lingual Projection (CLP)

Cross Lingual Projection (CLP) (Faruqui, 2015) adopts an unsupervised two-step approach for generating Open IE triples from a sentence in a source language such as Spanish. A machine translation system is used to translate the sentence into a target language, such as English. An Open IE system such as OLLIE (Mausam et al., 2012) is used to generate the extractions from the translated English sentence. The words from the generated English extractions are *projected* back into the original source sentence to generate the final set of extractions in Spanish.

The projection algorithm uses automatically detected alignments between the words in the two languages. Since the word alignments are a many-to-many mapping and the subject, relation, object fields are mostly contiguous phrases, they use a phrase-extract algorithm to detect all possible mappings between contiguous phrases in the source and target languages. For the projection of the relation field, the source phrase from the extracted phrase map with the highest BLEU score is chosen. The correspondingly mapped target phrase in Spanish language is marked as a relation.

We describe the CLP algorithm for projecting labels from English extraction to other language with the help of an example. Consider English sentence, E: *Dutil - Dumas experiment was promoted by an organization called Encounter 2001 denotes* and Spanish sentence, S: *Experimento Dutil - Dumas fue promovido por una organización llamada Encounter 2001*. The word alignments between these sentences are listed in Figure 2.1, and equivalent phrases from the phrase extract algorithm are shown in Table 2.1. Consider the English extraction, *(Dumas*

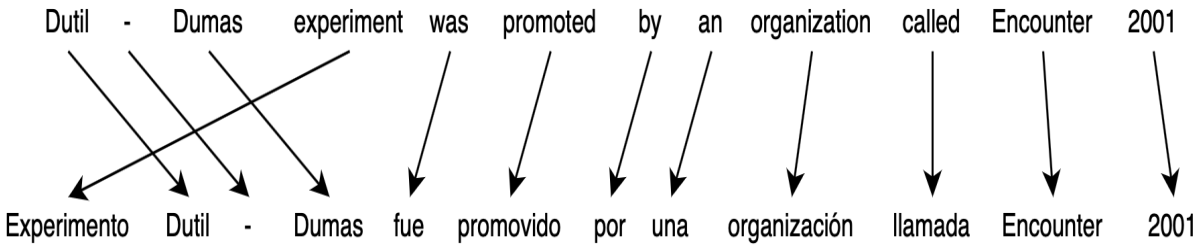

Figure 2.1: Equivalent English and Spanish sentence with corresponding word alignments between them

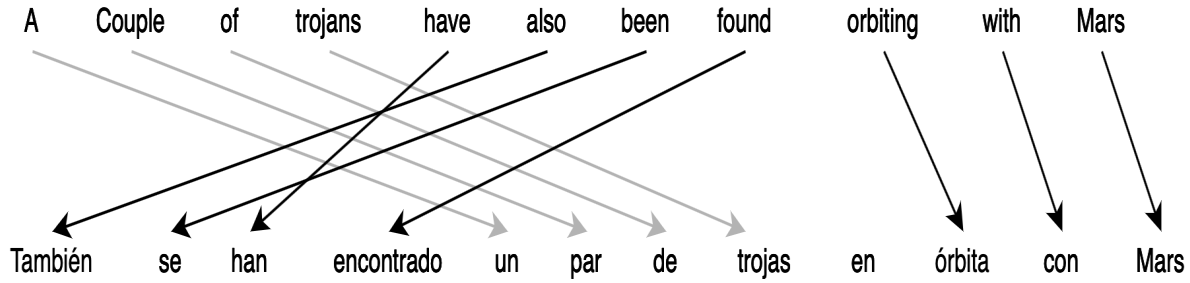

Figure 2.2: Equivalent English and Spanish sentence with corresponding word alignments between them

experiment; was promoted; by an organization). For each phrase in the tuple, CLP algorithm looks for the highest BLEU match phrase from Table 2.1. The subject phrase *Dumas experiment* has best BLEU match to *Dutil - Dumas experiment*, and so the corresponding Spanish phrase *Experimento Dutil - Dumas* will be marked as subject. Note that the phrase *Dumas experiment* is not present in Table 2.1 because its aligned phrase is not continuous in Spanish sentence as can be seen in Figure 2.1. Similarly for the relation phrase *was promoted*, we find *fue promovido* from Table 2.1. Continuing the same algorithm, we get *(Experimento Dutil - Dumas; fue promovido; por una organización)* as the final Spanish extraction.

The work is originally motivated by open relation extraction, where only the relation phrase is projected. But the method is trivially extendible to the complete Open IE task itself by projecting the subject and object as well, which is used in Section 6.2.3.

English Phrases	Spanish Phrases
Dutil - Dumas experiment	Experimento Dutil - Dumas
Dumas	Dumas
experiment	Experimento
was promoted	fue promovido
....	....

Table 2.1: Mapped continuous phrases between English (E) and Spanish (S) language sentences from the phrase extract algorithm

2.4.5.2 PredPatt

PredPatt (White et al., 2016) builds on top of universal dependency parses[5] to extract predicate-argument structures, similar to Open IE. They reduce the deep semantic notions associated with universal dependencies to a shallow semantic framework of predicates and their corresponding arguments. It uses interpretable rules for this conversion and results in a system that achieves significantly better P-R curves compared to other systems. Since the rules operate on top of universal dependency tags, which remain same across languages, PredPatt can be easily extended to work for other languages as well. The performance of PredPatt in multilingual setting has been explored in Ro et al. (2020).

2.4.5.3 ArgOIE

ArgOIE (Gamallo and Garcia, 2015) works for multiple languages by using dependency parses in CoNLL-X format (Buchholz and Marsi, 2006) to generate Open IE extractions. CoNLL-X is a language independent way to represent the dependency tags. They propose a set of rules to convert the parses into a set of prepositions. Since the parses are language-agnostic in nature, the ArgOIE system can be applied to any language. They perform experiments on three languages – English, Spanish and Portuguese.

2.4.5.4 CrossOIE

CrossOIE (Cabral et al., 2020) proposes a multilingual classifier that denotes the validity of a given Open IE triple. They use extractions in three languages – English, Spanish and Portuguese, manually annotated with their validity or invalidity to train a language model based classifier. They experiment with two types of language models, mBERT and XLM. They observe strong zero-shot performance even in the absence of training data for a particular language. This model can be used to infer the quality of triples that can be used for boostrapping Open IE systems in the corresponding language.

2.4.5.5 Multi2OIE

Owing to their pipelined nature, PredPatt and ArgOIE performance is below that of neural systems. Ro et al. (2020) proposed Multi^{2}OIE, a sequence-labeling model for Open IE, which first predicts all the relation arguments using BERT, and then predicts subject and object arguments associated with each relation using multi-head attention blocks. Their model cannot handle nominal relations and conjunctions in arguments. The underlying mBERT encoder in Multi^{2}OIE allows for cross-lingual generalization across various languages even after training with only English supervised data. However, dependence on zero-shot generalization also limits the performance of the model.

In summary, we discuss the various Open IE systems that have been proposed for handling the task of Open IE in languages other than English. In Chapter 6, we discuss our contributions to this space by proposing a technique to generate Open IE training data in multiple languages.

[5] https://universaldependencies.org/

2.5 Applications of Open IE

Apart from the intrinsic value of Open IE extractions, they have also proven to be helpful in various downstream tasks. They are briefly summarized in this section.

2.5.1 Text Summarization

Multi-document summarization involves generating a human-readable summary using the information contained in multiple documents. Christensen et al. (2014) proposed a hierarchical scheme for summarizing multi-documents. The hierarchical scheme makes use of Open IE to measure similarity between sentences as the number of shared tuples. These scores are used to ensure that the summary contains non-redundant sentences. Fan et al. (2019a) also uses Open IE for the task of multi-document summarization. Since transformer models scale quadratically with the size of the input, they are not capable of handling very large inputs. Processing multiple documents for summarization using a transformer model would be infeasible. Therefore, they dynamically construct open knowledge graphs by generating Open IE extractions from multiple documents. These knowledge graphs are linearized and passed to a Seq2Seq model, which then generate the final summary. Their technique shows strong results in two datasets, WikiSum (Liu et al., 2018) and ELI5 (Fan et al., 2019b).

Ribeiro et al. (2022) aims to evaluate the factuality of the generated extractions in an interpretable fashion. To achieve this, they extract semantic representations of both the source document and the generated summary. These semantic representations capture the entities and relavant relations among them, which are structured in the form of graphs (FactGraphs). By comparing the similarity across graphs, they achieve the desired goal of evaluating factuality. They experiment with both Open IE and Abstract-Meaning Representation (AMR) graphs and find Open IE to be competitive in performance, when evaluated on CNN/DM and XSum datasets.

2.5.2 Question Answering

Fader et al. (2013) presents a seminal approach for using Open IE facts for answering open-domain questions. They introduced a completely unsupervised scheme to learn lexicon matching between the query and the database of knowledge facts. Instead of using training data, they rely on 16 manually annotated seed templates, which are then used to bootstrap additional patterns from WikiAnswers. Using learned lexicon equivalences, they are use to convert the given query into terms that can be easily matched to the database for extracting the most relevant answer.

Khot et al. (2017) proposes a scheme for answering complex multiple choice questions based on the information expressed in Open IE tuples. They first identify the top-1000 tuples that are most closely linked to the question based on lexical word overlap. Then they use a graph matching algorithm to see how well a graph can be formed with the given question, choice and chosen Open IE tuple to determine the score of each choice. The approach relies on using a single tuple for each question, thus preventing multi-hop reasoning.

2.5.3 Event Extraction

Balasubramanian et al. (2013) focuses on extracting open-domain schemas for specific events, such as a “terrorist bombing”. They aim to identify the various actors and how they are linked to each other through semantic relations. Open IE is used to construct knowledge base of tuples

from the corpus of texts relevant to the event. Analysis on these tuples and their co-occurrence are used to identify the various types of actors that occur in the events. The co-occurrence information is referred to as Rel-grams and represents relations between abstracted Open IE tuples.

Pratapa et al. (2021) creates a new dataset for linking events across documents, referred to as the Cross-Document Event Coreference (CDEC) dataset. They identify all open-domain events in the chosen set of Wikipedia documents using RnnOIE and provide dense annotations whether each pair of identified events are linked to each other or not.

2.5.4 Entity and Relation Linking

TENET (Lin et al., 2021) proposes a method for joint entity and relation linking that considers the coherence of mentions within a document, i.e., the entities repeated in a document are usually similar. In order to take this coherence into account, they build a Knowledge Coherence Graph that contains the detected entity mentions, relation mentions and independently identified entities and relations. The relation mentions are identified using the MinIE system. Using this dynamically constructed KG for each input document, they employ a global optimization to assign coherent entities and relations to the various detected mentions using tree-coverage techniques.

2.5.5 Video Grounding

Video Grounding (Hendricks et al., 2017) refers to the task of extracting the video frame that is most relevant to a given textual caption. Many methods proposed for the task of video grounding use image-text similarity matching algorithms (Gao et al., 2017; Zhang et al., 2020b). However they often result in spurious correlations that are learnt from the training data. Nan et al. (2021) propose a novel paradigm of using causal inference for removing the selection bias resulting from the training data. Since the sampling distribution for the training data creation is not available, they make use of heuristics to determine the prior probability of a sample. They use a novel method to estimate this prior probability directly from the textual caption. RnnOIE is used to extract the subject, relation and object phrases from the textual caption. The probability of finding each of these phrases in the dataset is assumed to be the prior probability for obtaining the sample in the training data.

2.5.6 Scientific Text

Most of the Open IE extractors are evaluated on text from general domains such as news and Wikipedia. However, the need for such open extractors is very high in this specific domain, considering the amount of research output being generated in the past few years. These extractors would help immensely in processing the information in an intelligent fashion to help research scientists make the best use of the prior research to develop techniques for tackling crucial life-saving problems.

Groth et al. (2018) devise a scheme to check the performance of systems on sentences from scientific publications and compares it with extractions from sentences in Wikipedia. They use crowdsourcing to annotate if a given extraction is correct or not. In this way, they only test for the precision of the systems and not the recall (which would require expert annotaters). They compare two SoTA systems - OpenIE-4 and MinIE and find that both systems suffer from a considerable gap when applied to science publications vs Wikipedia, and among the two,

OpenIE-4 outperforms MinIE in precision. These extractions can help in quickly analyzing vast quantities scientific text and potentially speed up the research process.

In this section, we have summarized the various use cases for Open IE extractions which have found downstream utility in a broad range of tasks. However, the challenge remains for the research community to explore the utility of Open IE extractions in the neural context where dense vector representations and end-to-end modeling outperform various types of semi-structured representations for text.

2.6 Related Tasks

In this section, we discuss some of the tasks that are related to Open IE. We discuss the task of Semantic Role Labeling (SRL), which is closely related to the task of Open IE. Finally, the task of Open IE has given rise to other sub-fields that resolve some of the challenges arising from the usage of open extractions. We discuss two such tasks of Open Link Prediction and Canonicalization.

2.6.1 Ontological/Closed IE

Traditional Information Extraction relies on a pre-defined ontology that guides the conversion of unstructured text to a structured format. The tasks of entity linking and relation classification traditionally fall under this paradigm and they are briefly described below:

Entity Linking: Entity linking involves finding the KB entity referred to by a mention marked in the input sentence. Multilingual pretrained models have advanced the task of entity linking across languages. MEL (Botha et al., 2020) uses a dual encoder, cross-encoder pipeline trained on hard negatives to achieve strong performance on 104 languages. mGENRE (Cao et al., 2021) is an entity-linking model that autoregressively generates the linked entity name using a Seq2Seq model. Moreover, it augments the decoder with a prefix trie to generate only valid entity names. We find that combining these models using a dual encoder, constrained generation pipeline leads to strong multilingual fact-linking performance.

Relation Classification: The task involves identifying the relation between a pair of entity mentions. It is also referred to as relation extraction. The recently released WebRED (Ormándi et al., 2021) dataset provides a set of English sentences annotated with the corresponding Wikidata relation/predicate and the linked entities. We create IndicLink using these examples. Other multilingual relation classification datasets such as RELX (Koksal and Ozgur, 2020), DiS-ReX (Bhartiya et al., 2022) are unusable for fact alignment as they don't provide the linked entities. From the task-modelling perspective, it has traditionally been posed as a classification task (Xiao and Liu, 2016; Ormándi et al., 2021). However, recent techniques (Nayak and Ng, 2020; Huguet Cabot and Navigli, 2021) have shown strong performance using generative models. Nayak and Ng (2020) treats the entire triple of the two entities and the relation as a single text and generates it using a Seq2Seq model. Huguet Cabot and Navigli (2021) further makes use of the power of pre-trained Seq2Seq models to improve relation extraction performance over 200 relations.

Fact Extraction: The task involves joint entity and relation extraction (Zhong and Chen, 2021; Sui et al., 2020) focusing on discovering new facts that are not present in the KB. Whereas fact linking deals with connecting existing KB facts with text. Therefore, fact-linking models use KB facts (which may be millions), whereas fact extraction systems do not.

2.6.2 Semantic Role Labeling

Semantic role labeling identifies the semantic roles of the words in the sentence, about how they are related to a mentioned verb. Similar to Open IE, they use the notion of agent and recipient to correspond to subject and object. The verb or predicate corresponds to the relation in Open IE. For example, in "Mary loves John", "loves" is the predicate with "Mary" as the agent and "John" as the recipient. However, SRL cannot have multi-word relational phrases. Moreover, they cannot handle noun-based relations or identify implicit relations in the sentence. The use of a semantically strict set of labels requires higher linguistic expertise from the end users. This is not the case with Open IE.

2.6.3 Open Link Prediction

Open Knowledge bases represent a special type of Knowledge Base which don't use a predefined ontology. Instead, they use the triples generated using Open IE systems as the facts. Open KBs have the potential to augment existing ontology-based KBs and make them widely applicable to domains not considered during the construction of the KB. Constructing such Open KBs using Open IE extractions has been a long-standing goal in the IE community. Many methods have been proposed to add new triples to such open knowledge bases by predicting new links between existing nodes. Due to the open-domain nature of the problem, such completion methods are also referred to as Open Link Prediction. The un-normalized surface forms of entities and relations make link prediction challenging.

Broscheit et al. (2020) provides a benchmark for this task called OLPBench. It uses the OPIEC KB (Gashteovski et al., 2019), which was constructed by curating the triples generated from Wikipedia. MinIE was used as the underlying Open IE system for making the KB. OLPBench also provides a version of the test set that ensures no overlapping facts with the train set. No overlap is ensured to understand the true generalization of the link prediction systems.

2.6.4 Canonicalization

The subject, relation, and object phrases extracted from Open IE are not grounded in a common framework, resulting in a large number of duplicates referring to the same ground truth entity. For example, "Barack Obama", and "Barack H. Obama" refer to the same entity but would be considered distinct in the Open IE extractions. Similarly, for the relation phrase "the boss of", syntactic variants such as "boss of" or synonymous variants such as "leader of" refer to the same relation phrase. The task of canonicalization deals with assigning a standard reference for such similarly grounded phrases.

CESI Vashishth et al. (2018) presents a novel way for canonicalization using side information along with Open IE extractions in order to cluster the entities and relations. The heads of the clustered entities and relations are considered canonicalized representations.

A wide variety of side information is used to achieve their final goal. For the entities, linking to Wikipedia pages, Wordnet types, a paraphrase database (PPDP), and morphology-based normalization is used as relevant side-information. For relations, existing KB Populations techniques are used to provide links to existing KB relations.

They introduce a new dataset called ReVerb45K, which contains 45,000 high-quality extractions that are used to construct the Open KB. The evaluation is performed in terms of the precision of the generated clusters.

In this chapter, we have defined the task of Open IE and discussed various Open IE systems that have been proposed in the literature. We looked at them from the point of view of English and non-English Open IE systems, non-neural and neural Open IE systems, and how the Open IE systems are evaluated. We also described some downstream applications where Open IE extractions have been found to be helpful and the tasks which are closely related to Open IE.

Chapter 3

Generative Models for Open IE

As discussed in Chapter 1, the task of Open IE can be modeled as either a generative (Cui et al., 2018) or a labeling (Stanovsky et al., 2018; Ro et al., 2020) task. This chapter describes two neural architectures for Open IE that belong to the generative modeling paradigm. In Chapter 4, we describe neural architectures that belong to the labeling paradigm.

In generative modeling, Seq2Seq models are typically used to generate extractions from the original sentence. Models in this paradigm include a decoder to generate the words in an extraction. The different fields in the extraction tuple are usually demarcated by unique delimiter tokens, output by the decoder. Models usually differ in how they handle the generation of multiple extractions for each sentence. However, independent generation of extractions may result in redundancy, where the same information is expressed through similar or identical triples. Moreover, the data used for training the models often miss extractions due to a lack of high-quality data, resulting in reduced coverage of the trained models.

In Section 3.1, we introduce IMoJIE (Iterative Memory-Based Joint Open Information Extraction), an architecture that captures dependencies among extractions by repeatedly re-encoding the sentence along with the extractions generated so far before generating the subsequent extraction.[1] In Section 3.2, we introduce the Gen2OIE model, a two-stage generative architecture that factorizes the dependencies among extractions based on shared relations. Gen2OIE achieves better performance than IMoJIE due to improved handling of training noise by the use of a Relation Coverage (RC) heuristic that increases the information covered in its extractions.

3.1 IMoJIE: Iterative Memory Joint Open Information Extraction

In this section, we present IMoJIE (Kolluru et al., 2020b), an extension of CopyAttention (Cui et al., 2018). IMoJIE produces the next extraction conditioned on all previously extracted tuples, which results in a variable number of diverse extractions per sentence. We release IMoJIE and all related resources for further research.[2]

Our analysis of CopyAttention (Cui et al., 2018) reveals that it suffers from two drawbacks. First, it does not naturally adapt the number of extractions to the length or complexity of the input sentence. Second, it is susceptible to *stuttering*: extraction of multiple triples bearing redundant information.

[1] Samarth Aggarwal helped with the ideation and implementation of the IMoJIE architecture and included it as a part of his BTech thesis.

[2] https://github.com/dair-iitd/imojie

Sentence	He was appointed Commander of the Order of the British Empire in the 1948 Queen's Birthday Honours and was knighted in the 1953 Coronation Honours .
CopyAttention	(He ; was appointed ; Commander ... Birthday Honours) (He ; was appointed ; Commander ... Birthday Honours and was knighted ... Honours) (Queen 's Birthday Honours ; was knighted ; in the 1953 Coronation Honours) (He ; was appointed ; Commander of the Order of the British Empire in the 1948) (the 1948 ; was knighted ; in the 1953 Coronation Honours)
IMOJIE	(He ; was appointed ; Commander of the Order ... Birthday Honours) (He ; was knighted ; in the 1953 Coronation Honours)

Table 3.1: IMoJIE vs. CopyAttention. CopyAttention suffers from stuttering, which IMoJIE does not.

Sentence	Greek and Roman pagans , who saw their relations with the gods in political and social terms , scorned the man who constantly trembled with fear at the thought of the gods, as a slave might fear a cruel and capricious master .
OpenIE-4	(the man ; constantly trembled ;)
IMOJIE	(a slave ; might fear ; a cruel and capricious master) (Greek and Roman pagans ; scorned ; the man who constantly trembled with fear at the thought of the gods, as a slave might fear a cruel and capricious master) (the man ; constantly trembled ; with fear at the thought of the gods) (Greek and Roman pagans ; saw ; their relations with the gods in political and social terms)

Table 3.2: IMoJIE vs. OpenIE-4. Pipeline nature of OpenIE-4 can get confused by long convoluted sentences, but IMoJIE responds gracefully.

These limitations arise because its decoder has no explicit mechanism to remember what parts of the sentence have already been 'consumed' or which triples have already been generated. Even though decoder in earlier methods used a fixed-size beam for inference, beam search can only ensure that the extractions are not exact duplicates.

In response, we design a neural Open IE system that uses sequential decoding of tuples conditioned on previous tuples. We achieve this by adding every extraction generated hitherto to the encoder. This iterative process stops when an *EndOfExtractions* tag is generated by the decoder, allowing it to produce a variable number of extractions. We name our system **I**terative **Me**m**o**ry **J**oint Open **I**nformation **E**xtraction (**IMoJIE**).

At a high level, the next extraction from a sentence is determined in the context of all tuples extracted from it so far. Hence, IMoJIE uses a decoding strategy that generates extractions in a sequential fashion, one after another, each one being aware of all the ones generated prior to it. The architecture of IMoJIE is illustrated in Figure 3.1.

Formally, let us consider a sentence S with N word tokens $[w_1, w_2, \ldots, w_N]$ and set of M extractions generated by the system, $E = \{E_1, E_2, \ldots, E_M\}$, where each extraction E_i contains the tokens [<s>, $e^i_{s1}, \ldots,$, </s>, <r>, $e^i_{r1}, \ldots$ </r>, <o>, $e^i_{o1}, \ldots,$ </o>]. The tags, <s> and </s>, <r> and </r>, <o> and </o> indicate boundaries of the subject, relation and object, respectively.

In the IMoJIE model, given the sentence S and the previously generated extractions $\{E_1 \ldots E_{i-1}\}$, the generation of the next extraction E_i is conditioned on the extractions generated so far. A special extraction, E_{M+1} containing the token *EndOfExtractions* indicates that the model has generated all the extractions for the sentence. Therefore, the probability of generating

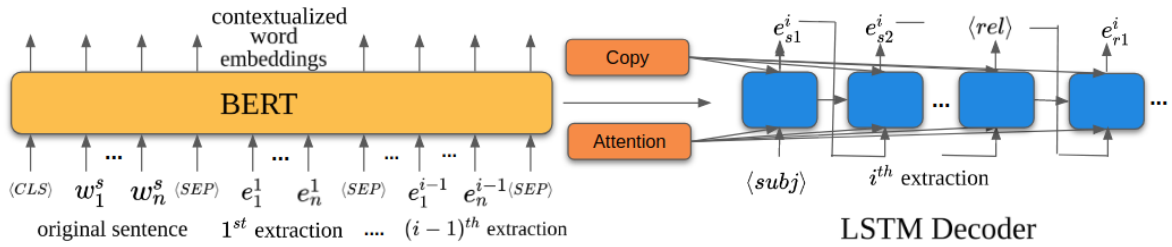

Figure 3.1: One step of the sequential decoding process, for generating the i^{th} extraction, which takes the original sentence and all extractions numbered $1, \ldots, i-1$, previously generated, as input.

the set of extractions E, from the sentence S can be expressed as:

$$\Pr(E|S) = \prod_{i=1}^{M+1} \left(\prod_{j=1}^{|E_i|} \Pr(E_{i,j}|S, E_{i,j-1}, E_{i,j-2}, \ldots, E_{i,1}, E_{i-1}, \ldots, E_1) \right) \tag{3.1}$$

where each extraction E_i is generated conditioned on extractions $E_{i-1} \ldots E_1$ and each token $E_{i,j}$ in E_i is generated autoregressively based on the previously generated tokens $E_{i,j-1} \ldots E_{i,1}$.

This kind of sequential decoding is made possible by the use of an *iterative memory*. Each of the generated extractions are added to the memory so that the next iteration of decoding has access to all of the previous extractions. We simulate this iterative memory with the help of BERT encoder, whose input includes the [CLS] token and original sentence appended with the decoded extractions so far, punctuated by the separator token [SEP] before each extraction.

IMoJIE uses an LSTM decoder, which is initialized with the embedding of [CLS] token. The contextualized embeddings of all the word tokens are used for the Copy (Gu et al., 2016) and Attention (Bahdanau et al., 2015) modules. The decoder generates the tuple one word at a time, producing ⟨*rel*⟩ and ⟨*obj*⟩ tokens to indicate the start of relation and object, respectively. The iterative process continues until the *EndOfExtractions* token is generated.

At the time of development of IMoJIE, pre-trained decoders were not yet well-established. Hence, we chose BERT as the encoder and LSTM as the decoder. However, using pre-trained Seq2Seq models such as T5 (Raffel et al., 2020) in the IMoJIE model results in almost similar performance while adding to the computational cost (shown in Section 3.4). This indicates that the iterative re-encoding strategy can overcome the limitations of a decoder without any pretraining. Hence, we continue to use BERT-LSTM as the backbone architecture for IMoJIE.

The overall process is summarized in Algorithm 1 and described below.

1. Pass the sentence through the Seq2Seq architecture to generate the first extraction.

2. Concatenate the generated extraction with the existing input and pass it again through the Seq2Seq architecture to generate the next extraction.

3. Repeat Step 2 until the *EndOfExtractions* token is generated.

IMoJIE is trained using a cross-entropy loss between the generated probability distribution and the gold token at every *j*th token of the *i*th extraction. The gold set of extractions for a sentence are ordered based on decreasing values of confidence scores that are assigned by the bootstrapping systems. However, we don't notice any statistically significant change in performance of the model, even when trained with the extractions randomly ordered.

Algorithm 1 IMoJIE Model

Input: Sentence $S = [w_1, w_2, \ldots, w_N]$, Encoder model Enc, Decoder model Dec

```
I ← [CLS] S
E_1 ← Dec(Enc(I))
E ← {E_1}
i ← 1
while true do
    if E_i = EndOfExtractions then
        break
    end if
    I ← I [SEP] E_i
    E_{i+1} ← Dec(Enc(I))
    E ← E ∪ {E_{i+1}}
    i ← i + 1
end while
```

Formally, the cross-entropy loss for the *j*th token in the *i*th gold extraction (denoted by *) can be written as:

$$CE_j^i = -\log \Pr(e_j^{*i} | e_1^{*i} \ldots e_{j-1}^{*i}) \tag{3.2}$$

The final cross-entropy loss, which is minimized for a sentence, sums up the above term over all tokens in all extractions as follows:

$$CE = \sum_{i=1}^{M} \sum_{j=1}^{len(E_i^*)} CE_j^i \tag{3.3}$$

3.1.1 Confidence Scoring

To assign a confidence value to every extraction, following previous generation systems (Xue et al., 2020), we compute the inverse of perplexity as assigned by the IMoJIE decoder. The log word probabilities assigned by the IMoJIE decoder are averaged to be used as a confidence score for each of the generated extractions. If E_i is the generated extraction, the confidence score associated with it is given by the following formula:

$$\text{Confidence}(E_i) = \frac{\sum_{j=1}^{len(E_i)} \log \Pr(e_j^i | e_1^i \ldots e_{j-1}^i)}{len(E_i)} \tag{3.4}$$

In the next section, we present another generative model that makes use of a two-stage pipeline that improves upon IMoJIE by using certain coverage heuristics.

3.2 Gen2OIE: Two-Stage Generative Model

In this section, we propose an improved generative approach called Gen2OIE, which, combined with a training heuristic to improve the coverage of extractions, establishes the current state of the art performance on the CaRB benchmark (Bhardwaj et al., 2019).

The model extends the 2-stage design of Multi^{2}OIE (Ro et al., 2020) to a generative paradigm. In the first stage, it uses a Seq2Seq model to generate all possible relations from the input sentence, and in the second stage, it uses another Seq2Seq model to generate all extractions that contain a given relation, i.e., completes the subject and object fields.

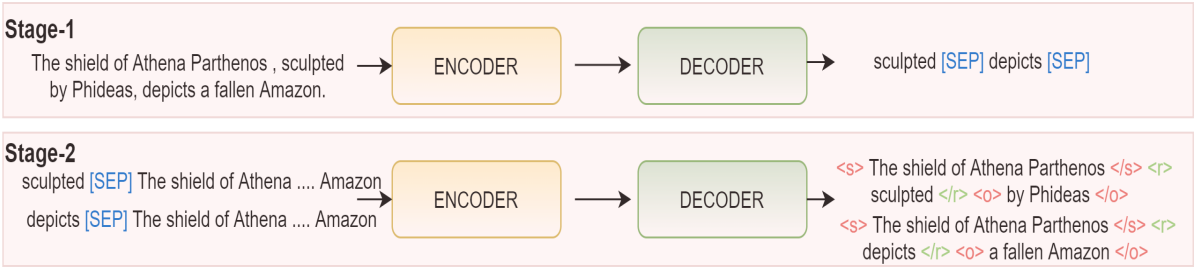

Figure 3.2: Gen2OIE model contains two Seq2Seq models. In Stage-1, it generates all relations in the sentence, separated by an [SEP] token. For each detected relation in Stage-2, it generates extractions containing the relation.

Algorithm 2 Gen2OIE model

Input: Sentence S, Stage-1 Seq2Seq model $RelsM$, Stage-2 Seq2Seq model $ArgsM$

$E \leftarrow \{\}$

$R \leftarrow RelsM(S)$ ▷ Get all relations from Stage-1

for $r \in R$ **do**

$E_{\{r\}} \leftarrow ArgsM(r$ `[SEP]` $S)$ ▷ Get extractions for each relation from Stage-2

$E \leftarrow E \cup E_{\{r\}}$

end for

Gen2OIE can produce overlapping relations and multiple extractions containing the same relation, thus overcoming the limitations of Multi^{2}OIE model. Moreover, due to its generative nature, Gen2OIE can add new words or introduce changes in morphology that may be necessary for producing correct extractions, which cannot be achieved by labeling models.

Both the stages of Gen2OIE (shown in Figure 3.2 and summarized in Algorithm 2) use Seq2Seq models as follows:

Stage-1 Seq2Seq $RelsM$: The input sentence S is passed to the encoder, and decoder generates a string formed by concatenating the set of all extracted relations, separated by `[SEP]` tokens. This concatenated string is referred to as R. During training, the target relations are concatenated in the order in which they occur in the sentence. We find that a deterministic order is important for adding stability to the model training.

Stage-2 Seq2Seq $ArgsM$: To produce extractions corresponding to each relation generated in Stage-1, the relation r is concatenated with the input sentence S and passed to the encoder as "r `[SEP]` S". The decoder is trained to generate all the extractions containing the relation r (referred to as $E_{\{r\}}$). Multiple extractions are separated by an `<e>` token, and hence separating $E_{\{r\}}$ based on `<e>` token gives us the set of individual extractions containing the relation r. Each extraction contains delimiter tokens to identify the various parts of the extraction. The surrounding `<s>...</s>`, `<r>...</r>` and `<o>...</o>` tokens are used to identify the subject, relation and object phrases, respectively.

We introduce a simple parts-of-speech based heuristic during Stage-1 training of Gen2OIE that increases the relation coverage in the generative paradigm. In Chapter 4, we also explore ways to achieve the same goal in the labeling paradigm.

Relation Coverage (RC): Verbs are usually strong expressions of relations. However, the extractions of training data may be incomplete and not satisfy this property. Therefore, during the training phase, we modify the input to the Stage-1 model by removing the verbs in the sentence which are not present in relation of any extraction. Thus the model learns that every verb must be included in some relation and applies this bias during inference as well. This heuristic does

not effect Stage-2 model training. Hereafter, Gen2OIE is understood to include training with Relation Coverage unless explicitly mentioned otherwise. We note that the Relation Coverage heuristic can only be applied to a two-stage pipeline like Gen2OIE because it artificially alters the input sentence, and so makes it infeasible to generate complete extractions from it. Hence, it cannot be applied to the IMoJIE model.

3.2.1 Confidence Scoring

The word log probabilities assigned by the Stage-2 decoder are average to be used as confidence score for the extractions generated by Gen2OIE. The equation for finding the confidence score remains the same as Equation (3.4).

3.3 Experimental Setup

In this section, we discuss the training data used, metrics evaluated and the systems with which IMoJIE and Gen2OIE are compared.

3.3.1 Training Data Construction

To train generative neural models for the task of Open IE, we need a set of (sentence, extraction) pairs. It is ideal for curating such a training dataset via human annotation, but such a dataset is unavailable for the task of Open IE. We follow Cui et al. (2018) and use bootstrapping — using extractions from a pre-existing Open IE system as 'silver'-labeled (as distinct from 'human' or 'gold'-labeled) instances to train the neural model. We first order all of the extractions in decreasing order of confidences output by the original system. We then construct training data assuming that this is the order in which it should produce the extractions.

We obtain our training sentences by scraping Wikipedia, because Wikipedia is a comprehensive source of informative text from diverse domains, rich in entities and relations. Using sentences from Wikipedia[3] ensures that our model is not biased towards data from any single domain. We run OpenIE-4[4] on 91K randomly sampled sentences to generate a set of Open IE tuples for every sentence. This results in a total of 181K extractions.

3.3.2 Evaluation Metric

We use the CaRB data and evaluation framework (Bhardwaj et al., 2019) to evaluate the systems[5] at different confidence thresholds, yielding a precision-recall curve. We identify two important summary metrics from the P-R curve.
F1: We find the point in the P-R curve corresponding to the largest F1 value and report it. This is the operating point for getting extractions with the best precision-recall trade-off.
AUC: This is the area under the P-R curve. This metric is useful when the downstream application can use the confidence value of the extraction.

[3]https://archive.org/details/enwiki-20170920

[4]https://github.com/knowitall/openie

[5]Our reported CaRB scores for OpenIE-4 and OpenIE-5 are slightly different from those reported by Bhardwaj et al. (2019). The authors of CaRB have verified our values.

System	F1%	AUC%
Stanford-IE	23	13.4
OLLIE	41.1	22.5
PropS	31.9	12.6
MinIE	41.9	-*
OpenIE-4	51.6	29.5
OpenIE-5	48.5	25.7
ClausIE	45.1	22.4
CopyAttention	35.4	20.4
CopyAttention + BERT	51.6	32.8
RNN-OIE	49.2	26.5
Sense-OIE	17.2	-*
Span-OIE	47.9	-*
Multi^{2}OIE	52.5	31.6
IMoJIE	53.2	**33.1**
GenOIE	52.1	30.3
Gen2OIE w/o RC	51.9	29.7
Gen2OIE	**54.4**	32.3

Table 3.3: Comparison of various Open IE systems: non-neural, neural and our proposed models. Gen2OIE outperforms all other systems. (*) Cannot compute AUC because Sense-OIE and MinIE do not emit confidence values for extractions, and released code for Span-OIE does not include calculation of confidence values.

3.3.3 Systems Compared

We compare IMoJIE and Gen2OIE against several non-neural baselines, including Stanford-IE (Angeli et al., 2015), OpenIE-4 (Christensen et al., 2011; Pal and Mausam, 2016), OpenIE-5 (Saha et al., 2017; Saha and Mausam, 2018), ClausIE (Del Corro and Gemulla, 2013), PropS (Stanovsky et al., 2016), MinIE (Gashteovski et al., 2017), and OLLIE (Mausam et al., 2012). We also compare with previously proposed neural Open IE models such as CopyAttention with and without using BERT encoder, (Cui et al., 2018), RnnOIE (Stanovsky et al., 2018), SenseOIE (t et al., 2019), SpanOIE (Zhan and Zhao, 2020) and Multi^{2}OIE (Ro et al., 2020).

Probably the most closely related baseline to IMoJIE is the neural generation baseline of CopyAttention. We compare against an English version of Logician (Section 2.4.4.1), which adds coverage attention, a module to ensure that all the important words in the input are covered, to a single-decoder model that emits all extractions one after another. We also compare against CopyAttention augmented with diverse beam search (Vijayakumar et al., 2018) — it adds a diversity term to the loss function so that new beams have smaller redundancy with respect to all previous beams.

To further analyze the effectiveness of the 2-stage architecture in Gen2OIE, we introduce another model called GenOIE, that outputs all extractions for a sentence as a single string, separated by an <*e*> token. The GenOIE model differs from the CopyAttention model, which outputs the various beams from a beam search as the multiple extractions for the input. CopyAttention only allows a fixed number of extractions for each sentence while GenOIE allows for variable number of extractions.

3.3.4 Implementation

We implement IMoJIE in the AllenNLP framework[6] (Gardner et al., 2018) using Pytorch 1.2. At the time of development of the IMoJIE model, pre-trained decoders were not yet well established. Hence, we chose BERT as the encoder and LSTM as the decoder for the IMoJIE model. We use "BERT-small" model as the encoder for faster training. In Section 3.4.5, we also experiment with using Seq2Seq pre-trained models as the IMoJIE backbone. Other hyper-parameters include learning rate for BERT, set to 2×10^{-5}, and learning rate, hidden dimension, and word embedding dimension of the decoder LSTM, set to $(10^{-3}, 256, 100)$, respectively. These hyperparameters are generally the standard values usually used for training BERT and LSTMs. Hyperparameter tuning resulted in no significant changes, and hence we stuck with the default choices.

Since the model or code of CopyAttention (Cui et al., 2018) were not available, we implemented it ourselves. Our implementation closely matches their reported scores of 47.3% AUC, achieving a (F1, AUC) of (56.4, 47.7)% on the OIE2016 benchmark.

We implement Gen2OIE using the T5 framework[7] using Tensorflow. We use "mT5-base" model for faster training. The multilingual version of T5 is used as the same model is also applied to other languages in Chapter 6. Other hyper-parameters include a learning rate of 0.001 with 24576 tokens per batch. As with IMoJIE, we use default hyperparameters recommended for T5, as intial hyperparameter tuning did not show any significant improvements.

To determine the speed of a system, we analyze the number of sentences it can process per second. We run all the systems on a common set of 3,200 sentences (Stanovsky et al., 2018), using a V100 GPU and four cores of Intel Xeon CPU (the non-neural systems use only the CPU).

3.4 Results and Analysis

In this section, we report the performance of IMoJIE and Gen2OIE and compare it with prior systems. Since IMoJIE was motivated as a model that solves the redundancy issue in CopyAttention, we also conducted experiments to determine if this has indeed been achieved.

3.4.1 Performance of IMoJIE

IMoJIE outperforms previously proposed neural and non-neural systems. It outperforms OpenIE-4, the best existing Open IE system, by 1.9% F1, 3.8% AUC. Qualitatively, we find that it makes fewer mistakes than OpenIE-4, probably because OpenIE-4 accumulates errors from upstream parsing modules (see Table 3.2).

IMoJIE outperforms CopyAttention by large margins — about 18% F1 and 13% AUC. Qualitatively, it outputs non-redundant extractions through the use of its iterative memory (see Table 3.1) and a variable number of extractions enabled by the *EndofExtractions* token. It also outperforms CopyAttention with BERT, which adds pretrained knowledge to the model and is thus a very strong baseline, by 1.9% F1 and 0.5% AUC.

RnnOIE performs much better than CopyAttention. However, it suffers due to its inability to generate auxilliary verbs and implied prepositions. E.g., it can only generate (Trump; President; US) instead of (Trump; is President of; US) from the sentence "US President Trump...". Moreover, it is trained only on limited number of pseudo-gold extractions generated by Michael et al. (2018).

[6] https://github.com/allenai/allennlp

[7] https://github.com/google-research/text-to-text-transfer-transformer

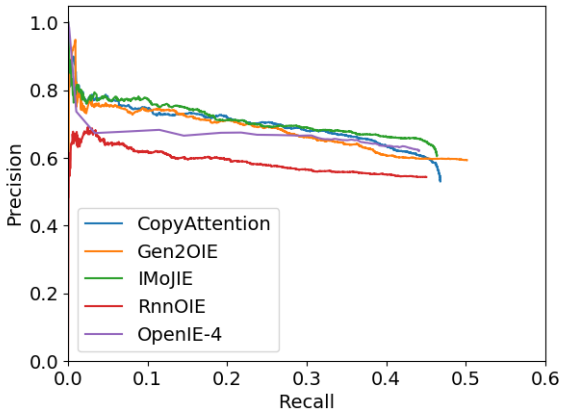

Figure 3.3: Precision-Recall curve of Open IE Systems.

CaRB evaluation of SpanOIE[8] results in (precision, recall, F1) of (58.9%, 40.3%, 47.9%). Hence, IMoJIE outperforms SpanOIE as well, both in precision and recall.

Figure 3.3 shows that the precision-recall curve of IMoJIE is consistently above that of existing Open IE systems, emphasizing that IMoJIE is consistently better than them across different confidence thresholds. We do find that CopyAttention+BERT outputs slightly higher recall at a significant loss of precision (due to its beam search with constant size), which gives it some benefit in the overall AUC.

3.4.2 Performance of Gen2OIE

We report the performance of Gen2OIE with and without using the RC heuristic. We find that the RC heuristic results in a significant increase of (2.5, 2.6)% in (F1, AUC). We find that using GenOIE results in (2.3, 2.0)% drop in F1, AUC compared to Gen2OIE, which leverages RC. We note that RC cannot be applied to GenOIE or IMoJIE as it involves the removal of words in the input sentence that may appear in other fields of the extraction.

Compared with IMoJIE, we see that Gen2OIE without RC performs worse by (1.3, 3.4)% in (F1, AUC). But with RC training, it beats IMoJIE by 1.2% in F1. These results indicate that although the architecture is less powerful than IMoJIE, the RC heuristic, applicable only to two-stage models, can handle noise in training data and improve the overall performance.

3.4.3 Redundancy

What is the extent of redundancy in IMoJIE output, when compared to earlier Open IE systems?

Apart from the models compared in Table 3.3, we additionally investigate various approaches to specifically reduce redundancy in CopyAttention, such as Logician's coverage attention (with both an LSTM and a BERT encoder) as well as diverse beam search. Table 3.4 shows that both approaches make significant improvements beyond CopyAttention. However, qualitative analysis of diverse beam search output reveals that the model gives out different words in different tuples in an effort to be diverse without considering their correctness. Moreover, since this model uses beam search, it still outputs a fixed number of tuples.

This analysis naturally suggested the IMoJIE (w/o BERT) model — an IMoJIE variation that uses an LSTM encoder instead of BERT. Unfortunately, IMoJIE (w/o BERT) is behind the

[8] github:zhanjunlang/Span_OIE

System	F1%	AUC%
CopyAttention	35.4	20.4
CoverageAttention	41.8	22.1
CoverageAttention+BERT	47.9	27.9
Diverse Beam Search	46.1	26.1
IMoJIE (w/o BERT)	37.9	19.1
IMoJIE	**53.2**	**33.1**

Table 3.4: Performance of models that attempt to address the redundancy issue prevalent in generative neural Open IE systems. All systems are bootstrapped on OpenIE-4.

CopyAttention baseline by 12.1% in AUC and 4.4% in F1. We hypothesize that this is because the LSTM encoder is unable to learn how to capture *inter-fact dependencies* adequately — the input sequences are too long for effectively training LSTMs.

This explains our use of Transformers (BERT) instead of the LSTM encoder to obtain the final form of IMoJIE. With a better encoder, IMoJIE is able to perform up to its potential, giving an improvement of **(17.8%, 12.7%)** in (F1, AUC) over existing Seq2Seq Open IE systems.

We further measure two quantifiable metrics of redundancy:

Mean Number of Occurrences (MNO): The average number of tuples every output word appears in.

Intersection Over Union (IOU): Cardinality of intersection over cardinality of union of words in the two tuples, averaged over all pairs of tuples.

These measures were calculated after removing stop words (from NLTK[9]) in the tuples. Higher values of these measures suggest higher redundancy among the extractions. IMoJIE is significantly better than CopyAttention+BERT, the strongest baseline, on both these measures (Table 3.5). Interestingly, IMoJIE has a lower redundancy than even the gold triples; this is due to imperfect recall. Gen2OIE also achieves lower redundancy compared to CopyAttention+BERT. We attribute this to the factorization of the problem into two stages, where the second stage predicts extractions corresponding to a distinct relation, thus reducing the chances of redundant extractions.

Extractions	MNO	IOU	#Tuples
CopyAttention+BERT	2.805	0.463	3159
IMOJIE	1.282	0.208	1598
Gen2OIE	1.310	0.283	1699
Gold	1.927	0.31	2650

Table 3.5: Measuring redundancy of extractions. MNO stands for Mean Number of Occurrences. IOU stands for Intersection over Union.

Attention is typically used to enable the model to focus on words considered important for the task. But the IMoJIE model successfully uses attention to *forget* certain words, those which are already covered. Consider the sentence "He served as the first prime minister of Australia and became a founding justice of the High Court of Australia". Given the previous extraction (He; served; as the first prime minister of Australia), the BERT's attention layers push the decoder to prioritize 'founding' and 'justice' as the words 'prime', and 'minister' have already been covered.

[9]https://www.nltk.org/

Base Architecture	Model	F1%	AUC%
GenOIE	BERT/LSTM	47.9	27.9
	T5	52.1	30.3
IMoJIE	BERT/LSTM	53.2	**33.1**
	T5	**53.5**	31.6

Table 3.6: Performance of IMoJIE and GenOIE architectures with BERT/LSTM and T5 base architectures. IMoJIE achieves similar performance with either of the architectures, but GenOIE achieves a significant increase. However, at the higher performance levels of IMoJIE, LSTM seems to better at confidence scoring compared to the transformer-based T5, resulting in a 1.5% drop in AUC from 33.1 to 31.6.

3.4.4 Performance with varying sentence lengths

In this experiment, we measure the performance of baseline and our models by testing on sentences of varying lengths. We partition the original CaRB test data into six parts with sentences of lengths 9-16 words, 17-24 words, 25-32 words, 33-40 words, 41-48 words and 49-62 words, respectively. Note that the minimum and maximum sentence lengths are 9 and 62, respectively. We measure the F1 score of both IMOJIE and Gen2OIE on these partitions as depicted in Figure 3.4. We observe that the performance deteriorates with increasing sentence length, which is expected. Also, for each of the partitions, Gen2OIE performs marginally better than or similar to IMoJIE.

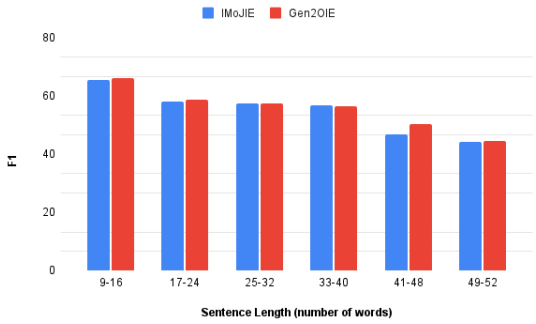

Figure 3.4: Measuring performance with varying input sentence lengths

3.4.5 Effectiveness of pre-trained decoders

Recent years have seen a rise in popularity of pre-trained Seq2Seq models such as mBART (Liu et al., 2020b) and mT5 (Xue et al., 2020). To test whether the iterative re-encoding strategy still provides value with the current generation of pre-trained models, we reimplement IMoJIE using the mT5 encoder-decoder model. The results are shown in Table 3.6. We also reimplement the single stage GenOIE model with BERT-LSTM, and it outputs all extractions for a sentence as a single string, separated by `<e>` tokens. GenOIE forms a strong baseline for IMoJIE as it also uses a similar autoregressive decoding strategy as IMoJIE without the iterative memory.

We find that while GenOIE (BERT/LSTM) lagged behind IMoJIE by a considerable margin of 5.3% F1 and 5.2% AUC, GenOIE (T5) fares much better, achieving only 1.1% F1 and 2.8% AUC lower than IMoJIE. It is also interesting to note that the IMoJIE (BERT/LSTM) can

achieve similar performance to IMoJIE (T5), indicating that the iterative re-encoding strategy can overcome the limitations of a weak decoder.

From a computational cost perspective, the BERT/LSTM model is much faster than the T5 model. For example, the inference speed of IMoJIE (T5) is 0.51 sentences per second, while that of IMoJIE (BERT/LSTM) is 2.6 sentences per second. Therefore, considering the faster inference time and a minor gap in F1 of 0.2%, we recommend using the IMoJIE with BERT/LSTM architecture as the default for future applications. In the rest of the dissertation, IMoJIE will indicate the model that uses BERT/LSTM as the encoder/decoder.

One natural-seeming way to combine IMoJIE and Gen2OIE would be to use iterative conditioning while generating the set of relations in Stage-1 of Gen2OIE. However, from the above results, we find that the iterative style of IMoJIE is more important when using weaker decoders such as LSTMs. With more powerful decoders, iterative conditioning doesn't seem to provide much additional value. Since Gen2OIE Stage-1 uses a pre-trained transformer decoder, changing it to iterative re-encoding doesn't seem promising.

3.4.6 Discussion on Order of Extractions

Open IE involves generating a set of tuples, a task that presents considerable challenges for neural models due to the unordered nature of the set along with the potential interdependencies among the set elements. Our experiments with IMoJIE and Gen2OIE indicate that the most critical interdependency is primarily the element's existence or presence in the set. This is needed to ensure that near-redundant extractions with only slightly differing subject/relation/object are not generated. The existential dependency implies that the exact order of extractions is not important for the final generation. The set of extractions should have nearly the same probability of generation irrespective of the order in which they have been generated.

As such, we do find that the order of extractions is not particularly vital to training the IMoJIE model. Due to the repeated encoding of generated extractions, the IMoJIE model can easily understand the presence/absence dependency among the extractions. However, in the Gen2OIE Stage-1 model, maintaining a deterministic order of relations (such as their order of appearance in sentence) does improve training stability. We attribute this to the fact that using random order of extractions for the same example across different batches results in the Gen2OIE Stage-1 model being trained with different outputs for the same input. This is not encountered with IMoJIE as random ordering also results in correspondingly different inputs, due to iterative concatenation strategy.

3.5 Conclusion

In this chapter, we have introduced two generative methods for the task of Open IE — IMoJIE and Gen2OIE. They are in the generative family of Open IE methods because they generate each word in the extraction, one after the other, in an auto-regressive fashion. IMoJIE also generates multiple extractions in an auto-regressive fashion, one extraction after another, with each extraction explicitly conditioned on all the previous extractions. However, Gen2OIE shortcuts this auto-regressive step at the extraction level by factorizing the extractions based on predicting relations and then generating all arguments corresponding to each predicted relation using a two-stage model.

Although they achieve superior accuracy compared to prior Open IE models, generative models are typically slower at inference time, as shown in Table 3.7. It is fundamentally due to

	Model	F1%	AUC%	Sentences/sec
Labeling	RnnOIE	49.0	26.0	**149.2**
	Multi^{2}OIE	52.5	31.6	41.3
Generative	IMoJIE	53.2	**33.1**	2.6
	GenOIE	52.1	30.3	1.4
	Gen2OIE	**54.4**	32.3	0.6

Table 3.7: Performance and Speed of labeling Open IE systems (RnnOIE, Multi^{2}OIE) and generative Open IE systems (IMoJIE, GenOIE, Gen2OIE) evaluated on the CaRB benchmark. Generative systems lead to better performance at the cost of slower inference speeds.

their autoregressive nature at the word level. Labeling models avoid this by generating labels for each word in the extraction in parallel. Therefore, in the next chapter, we develop novel labelling models that are much faster and, when trained with constraints, can effectively reduce the performance gap between labeling and generative models.

Chapter 4

Labeling Models for Open IE

As seen in Chapter 3, generative models output each word in the extraction in an auto-regressive manner. However, this comes at a high computational cost. An alternative paradigm is to label each input token as part of the subject, relation or object token of an extraction. In this chapter, we present a labeling-based system that achieves good performance for Open IE while processing inputs significantly faster.

Compared to generation systems such as IMoJIE (2.6 sentences per second), *labeling*-based systems like RnnOIE (Stanovsky et al., 2015) are much faster (149.2 sentences per second) but are relatively less accurate. The performance of RnnOIE is limited by the fact that each extraction is predicted independently, which does not model the inherent dependencies among the extractions.

We bridge this trade-off between speed and performance using a novel Open IE system that is both fast and accurate. It is based on a novel iterative labeling-based architecture — **I**terative **G**rid **L**abeling (**IGL**). Using this architecture, Open IE is modeled as a 2-D grid labeling problem of size (M, N) where M is the number of extractions and N is the sentence length, as shown in Figure 4.1. Each extraction corresponds to one row in the grid. Iterative assignment of labels in the grid helps IGL capture dependencies among extractions without the need for

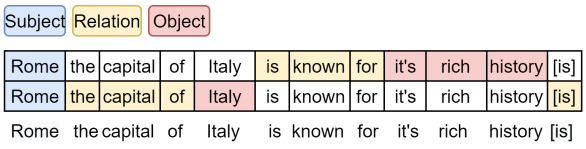

Figure 4.1: The extractions *(Rome; [is] the capital of; Italy)* and *(Rome; is known for; it's rich history)* can be seen as the output of grid labeling. We additionally introduce a synthetic token *[is]* to the input to facilitate more natural relation extractions.

Sentence	Other signs of lens subluxation include mild conjunctival redness, vitreous humour degeneration, and an increase or decrease of anterior chamber depth .
IGL	(Other signs of lens subluxation; include; mild conjunctival redness, vitreous humour degeneration)
IGL +Constraints	(Other signs of lens subluxation; include; mild conjunctival redness, vitreous humour degeneration, and an increase or decrease of anterior chamber depth)

Table 4.1: For the given sentence, IGL based Open IE extractor produces an incomplete extraction. Constraints improve the recall by covering the remaining words.

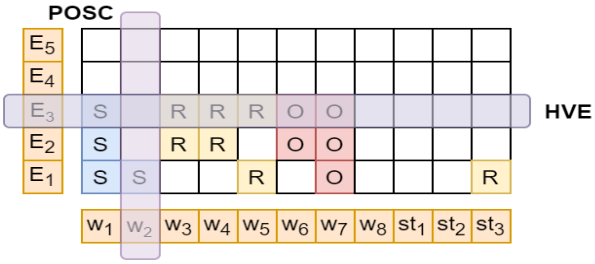

Figure 4.2: 2-D grid for Open IE with extraction as rows and words as columns. The values represent the labels (*S*)ubject, (*R*)elation, (*O*)bject. The empty cells represent *None*. Constraints can be applied across rows and columns.

re-encoding, thus making it much faster than generation-based approaches.

While IGL gives high precision, we can further improve recall by incorporating (soft) global coverage constraints on this 2-D grid. We use constrained training (Mehta et al., 2018; Nandwani et al., 2019) by adding a penalty term for all constraint violations. This encourages the model to satisfy these constraints during inference as well, leading to improved extraction quality, without affecting running time.[1]

4.1 Iterative Grid Labeling for Open IE

Recall that given a sentence S with N word tokens $[w_1, w_2, \ldots, w_N]$, the task of Open IE is to output a set of extractions, $E = \{E_1, E_2, \ldots, E_M\}$, where each extraction is of the form *(subject; relation; object)*. For a labeling-based system, each word is labeled as *S* (Subject), *R* (Relation), *O* (Object), or *None* for every extraction. We model this as a 2-D grid labeling problem of size (M, N), where the words represent the columns and the extractions represent the rows (Figure 4.2). The output at position (m, n) in the grid ($L_{m,n}$) represents the label assigned to the n^{th} word in the m^{th} extraction.

We propose a novel **I**terative **G**rid **L**abeling (IGL) approach to label this grid, filling up one row after another iteratively. Since each row corresponds to a distinct Open IE extraction, each extraction is conditioned on the previously generated extractions. We refer to the Open IE extractor trained using this approach as IGL-OIE. This iterative conditioning differentiates IGL-OIE from RnnOIE, which generates the extractions independently. The overall process is summarized in Algorithm 3 and shown schematically in Figure 4.3.

IGL-OIE is based on a BERT encoder, which computes contextualized embeddings for each word. The input to the BERT encoder is $[w_1, w_2, \ldots, w_N, \textit{[is]}, \textit{[of]}, \textit{[from]}]$. The last three tokens (referred as st_i) are appended because, sometimes, Open IE is required to predict tokens that are not present in the input sentence.[2] E.g., the sentence "*US president Donald Trump gave a speech on Wednesday.*" will have one of the extractions as *(Donald Trump; [is] president [of]; US)*. The appended tokens make such extractions possible in a labeling framework.

The contextualized embeddings for each word or appended token are iteratively passed through a 2-layer transformer to get their Iterative Layer (IL) embeddings at different levels, until a maximum level M, i.e. a word w_n has a different contextual embedding $IL_{m,n}$ for every row (level) m. At every level m, each $IL_{m,n}$ is passed through a fully-connected labeling layer to get the labels for words at that level (Figure 4.3). Embeddings of the predicted labels are added

[1] The code and trained models are available at https://github.com/dair-iitd/openie6

[2] 'is', 'of' and 'from' are the most frequent such tokens adopted by the OpenIE-4 system.

Algorithm 3 IGL-OIE Model

Input: Sentence $S = [w_1, w_2, \ldots, w_N]$, Encoder model Enc, Self-Attention Layers SA, Label Classifier $LCls$, Label Embedder $LEmb$

$IL_0 \leftarrow Enc(S)$
$LE_0 \leftarrow 0$
$E \leftarrow \{\}$
for $i = 1, 2, 3 \ldots,$ *M* **do**
  if E_{i-1} = *Empty* **then**
    break
  end if
  $IL_i \leftarrow SA(IL_{i-1} + LE_{i-1})$
  $E_i \leftarrow LCls(IL_i)$
  $E \leftarrow E \cup \{E_i\}$
  $LE_i \leftarrow LEmb(E_i)$
end for

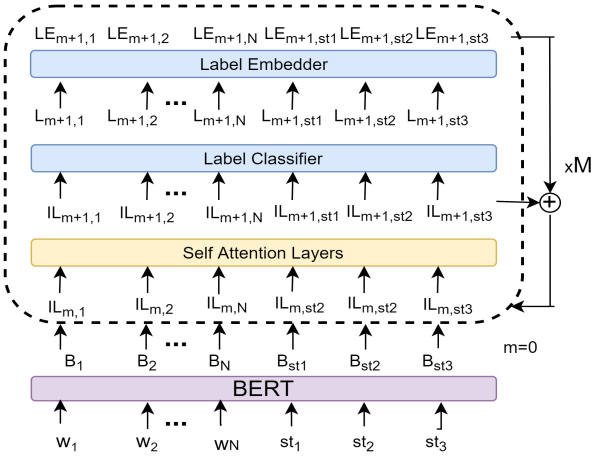

Figure 4.3: Architecture of IGL. BERT-embeddings of the words are iteratively passed through self-attention layers. st_1, st_2, st_3 refer to the appended tokens *[is]*, *[of]*, *[from]*, respectively. At every iteration, we get an extraction by labeling the words using a fully-connected layer. Embeddings of the generated labels are added to the iterative layer embeddings.

to the *IL embeddings* before passing them to the next iteration. This, in principle, maintains the information of the extractions output so far, and hence can capture dependencies among labels of different extractions. For words that were broken into word pieces by BERT, only the embedding of the first word piece is retained for label prediction. We find the model performance to be insensitive to this design choice as choosing the embedding of the last word-piece or an average of all word-piece embeddings gave similar results. We sum the cross-entropy loss between the predicted labels and the gold labels at every level to get the final loss, denoted by J_{CE}.

Open IE systems typically assign a confidence value to an extraction. In IGL, at every level, the respective extraction is assigned a confidence value by adding the log probabilities of the predicted labels (*S*, *R*, and *O*), and normalizing this by the extraction length.

4.2 Grid Constraints

Our preliminary experiments revealed that IGL-OIE has good precision, but misses important extractions. In particular, we observed that the set of output extractions may not capture all the information from the sentence (Table 4.1). We formulate constraints over the 2-D grid of extractions (as shown in Figure 4.2) which act as an additional form of supervision to improve the recall. We implement these as soft constraints, by imposing additional violation penalties in the loss function. This biases the model to learn to satisfy the constraints, without explicitly enforcing them at inference time.

To describe the constraints, we first define the notion of a *head verb* as all verbs except light verbs (*do, be, is, has,* etc). We run a POS tagger on the input sentence and find all head verbs in the sentence after removing all light verbs.[3] For example, for the sentence *"Obama gained popularity after Oprah endorsed him for the presidency"*, the head verbs are *gained* and *endorsed*. In order to cover all valid extractions like *(Obama; gained; popularity)* and *(Oprah; endorsed him for; the presidency)*, we design various coverage constraints. Note that these constraints are only encouraged. We introduce some notation that is used for formulating the constraints.

Notation: Let p_n be the POS tag of the word w_n. We define an importance indicator $x_n^{imp} = 1$ if $p_n \in \{$Noun, Verb, Adjective, Adverb$\}$, and 0 otherwise. Similarly, let $x_n^{hv} = 1$ denote that w_n is a head verb. At each extraction level m, the model computes $Y_{mn}(k)$, the probability of assigning the n^{th} word the label $k \in \{$S, R, O, *None*$\}$.

We now describe the constraints and the penalty terms corresponding to these constraints.

4.2.1 POS Coverage (POSC)

All words with POS tags as nouns, verbs, adjectives, and adverbs should be part of at least one extraction. E.g. the words *Obama*, *gained*, *popularity*, *Oprah*, *endorsed*, *presidency* should be covered in the set of extractions.

To ensure that the n^{th} word is covered, we compute its maximum probability ($posc_n$) of belonging to any extraction. We introduce a penalty if this value is low. This penalty is aggregated over words with important POS tags, $J_{posc} = \sum_{n=1}^{N} x_n^{imp} \cdot posc_n$, where

$$posc_n = 1 - \max_{m \in [1,M]} \left(\max_{k \in \{\text{S,R,O}\}} Y_{mn}(k) \right)$$

4.2.2 Head Verb Coverage (HVC)

Each head verb should be present in the relation span of some (but not too many) extractions. E.g. *(Obama; gained; popularity)*, *(Obama; gained; presidency)* is not a comprehensive set of extractions.

A penalty is imposed for the n^{th} word if it is not present in the relation span of any extraction or if it is present in the relation span of many extractions. This penalty is aggregated over head verbs, $J_{hvc} = \sum_{n=1}^{N} x_n^{hv} \cdot hvc_n$, where $hvc_n = \left| 1 - \sum_{m=1}^{M} Y_{mn}(R) \right|$.

We subtract the summation from one to penalize the model in case of the model does not generate the head verb in the relation of any of the extractions.

[3] We used the light verbs listed by Jain and Mausam (2016).

4.2.3 Head Verb Exclusivity (HVE)

The relation span of one extraction can contain at most one head verb. E.g. *gained popularity after Oprah endorsed* is not a good relation as it contains two head verbs.

A penalty is imposed if the relation span of an extraction contains more than one head verb. This penalty is summed over all extractions. I.e., $J_{hve} = \sum_{m=1}^{M} hve_m$, where

$$hve_m = \max\left(0, \left(\sum_{n=1}^{N} x_n^{hv} \cdot Y_{mn}(R)\right) - 1\right)$$

4.2.4 Extraction Count (EC)

The total number of extractions with head verbs in the relation span must be no fewer than the number of head verbs in the sentence. In the example, there must be at least two extractions containing head verbs, as the sentence itself has two head verbs.

ec_m denotes the score $\in [0, 1]$ of the m^{th} extraction containing a head verb, i.e. $ec_m = \max_{n \in [1,N]} \left(x_n^{hv} \cdot Y_{mn}(R)\right)$. A penalty is imposed if the sum of these scores is less than the actual number of head verbs in the sentence.

$$J_{ec} = \max\left(0, \sum_{n=1}^{N} x_n^{hv} - \sum_{m=1}^{M} ec_m\right)$$

Ideally, no constraint violations of **HVC** and **HVE** would imply that **EC** would also never get violated. However, as these are implemented as soft constraints, both constraints help in practice. We find that our model performs better and results in fewer constraint violations when trained with **POSC**, **HVC**, **HVE** and **EC** combined. The full loss function is given by:

$$J = J_{CE} + \lambda_{posc} J_{posc} + \lambda_{hvc} J_{hvc} + \lambda_{hve} J_{hve} + \lambda_{ec} J_{ec}$$

where $\lambda_\star$ are hyperparameters. We refer to the Open IE extractor trained using this constrained loss as Constrained Iterative Grid Labeling Open IE Extractor (CIGL-OIE).

The model is initially trained without constraints for a fixed *warmup* number of iterations, followed by constrained training till convergence.

4.3 Confidence Rescoring

The extractions output by an Open IE system is typically assigned a score based on the model's confidence for generating the particular extraction. However, the extractions generated by Open IE systems can be further rescored using a different set of models than the ones used to generate them. We find that this often leads to better calibration and an increase in the AUC of the system. We experiment with two types of rescoring models: labeling-based and generation-based models. The two approaches are briefly described below.

Labeling-based A sequence-labeling model is trained on extractions with ext-sentence (the *sentencized* form of the extractions after removing the tags) as input and S, R, and O labels over the ext-sentence as the output. The log probabilities given by the sequence-labeling model to the labels predicted by the Open IE system are summed up to get the new confidence scores of the extraction.

Generation-based A generative model consisting of a BERT encoder and an LSTM decoder is trained on (sentence, extraction) pairs. This trained model is used to compute the log-likelihood of the given extraction that has been generated by an Open IE system.

4.4 Experimental Setup

We train IGL and CIGL using the OpenIE-4 training dataset (from Section 3.3.1). We convert each extraction to a sequence of labels over the sentence. This is done by looking for an exact string match of the words in the extraction with the sentence. In case there are multiple string matches for one of the arguments of the extraction, we choose the string match closest to the other arguments. This simple heuristic covers almost 95% of the training data. We ignore the remaining extractions that have multiple string matches for more than one argument.

We implement our models using Pytorch Lightning (Falcon, 2019). We use pre-trained weights of "BERT-base-cased"[4] for Open IE extractor. We do not use BERT-large for Open IE extractor as we observe almost identical performance, but with a significant increase in computational costs. We set the maximum number of iterations, M=5 for Open IE. We use the SpaCy POS tagger[5] for enforcing constraints.

We follow the experimental setup mentioned in Section 3.3.

4.5 Experiments

System	CaRB		Speed
	F1%	AUC%	Sentences/sec.
MinIE	41.9	-	8.9
ClausIE	45.0	22.0	4.0
OpenIE-4	51.6	29.5	20.1
OpenIE-5	48.0	25.0	3.1
SenseOIE	28.2	-	-
SpanOIE	48.5	-	19.4
RnnOIE*	49.0	26.0	64
CopyAttention	51.6	32.8	11.5
IMoJIE	53.2	33.1	2.6
Gen2OIE	**54.4**	**34.2**	0.6
IGL-OIE	52.5	31.7	**142.0**
CIGL-OIE	54.0	33.6	**142.0**

Table 4.2: Evaluation of Open IE. Using constrained learning, CIGL-OIE gives better F1 than IMoJIE and reaches close to Gen2OIE. MinIE, SenseOIE, SpanOIE do not output confidence. The code of SenseOIE is not available to compute speed. *For RnnOIE, the reported speed is 149.2 sentences/sec, however, we have only been able to reproduce 64 sentences/sec with their latest implementation.

[4]https://github.com/huggingface/transformers
[5]https://spacy.io

4.5.1 Speed and Performance

How does IGL and CIGL compare in speed and performance?

Table 4.2 reports the speed and performance comparisons. We find that the base Open IE extractor — IGL-OIE — achieves a 60× speed-up compared to IMoJIE, while being lower in performance by 1.1% F1.

We find that training IGL-OIE along with constraints (CIGL-OIE), helps to improve the performance without affecting inference time. It beats IMoJIE by 0.8% F1 and narrows the gap with Gen2OIE to only 0.8% in F1. To improve the AUC we experiment with the two types of rescoring methods on the three neural architectures — IMoJIE, CIGL and Gen2OIE. The OpenIE-4 bootstrapped training data is used for training both the labeling-based and generation-based rescoring methods.

In Table 4.3, we notice that label-based rescoring performs most effectively in terms of AUC, outperforming both generation-based rescoring and the original model confidences themselves. However, analyzing the P-R plot corresponding to the label-rescoring methods, as shown in Figure 4.5, Figure 4.6 and Figure 4.7, we notice discontinuous curves. The discontinuity occurs as a consequence of the system outputting confidence values only within a limited range. This limits the recall values possible by filtering out low-confidence extractions. Such discontinuous curves are not properly evaluated using the trapezoidal-rule-based AUC that has been adopted by CaRB.

Therefore, we also measure the AUC of the interpolated P-R curve[6] (Manning et al., 2005), which is better suited for these cases. It considers the highest precision achieved by the model at or beyond a particular recall level. The interpolation thus considers the precision at every point to be the maximum precision value achieved at any correspondingly larger recall value. Thus, the empty region to the left of the curve is replaced by a flat line at the highest precision level. We refer to this metric as AUC_{int}.

Using this AUC_{int}, it is revealed that label-rescoring methods in fact perform worse than generation-rescoring. Generation rescoring improves over normal model confidences with both the AUC and AUC_{int} metrics. The AUC increases by as much as 0.6% in the case of CIGL and 1.3% in the case of Gen2OIE. The AUC_{int} increases by 1.3% and 1.2% in the case of CIGL and Gen2OIE, respectively. The optimal value of F1 remains nearly the same for all architectures with generation rescoring except for a decrease of 0.9% AUC for IMoJIE in the case of label rescoring.

We hypothesize that generation rescoring results in the better calibration of confidence scores as it can capture the "grammaticality" of the extraction – the confidence that the extraction forms a grammatical sentence on its own. This is possible due to the auto-regressive nature of generation rescoring which is absent in label rescoring.

In summary, CIGL achieves good extraction quality with a very high inference speed. If well-calibrated confidence scores are needed for each extraction, generative rescoring can help to achieve it. The final P-R curves of all generative-rescored P-R systems are shown in Figure 4.4.

4.5.2 Constraints Ablation

How are constraint violations related to model performance?

We divide the constraints into two groups: one which is dependent on head verb(s): {HVC, HVE and EC}, and the other which is not – POSC. We separately train IGL architecture-based Open IE extractor with these two groups of constraints and compare them with no constraints

[6]https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

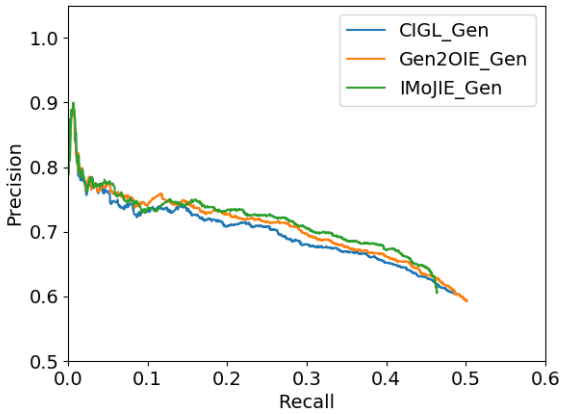

Figure 4.4: P-R curve of IMoJIE, Gen2OIE, CIGL and CIGL with generation rescoring.

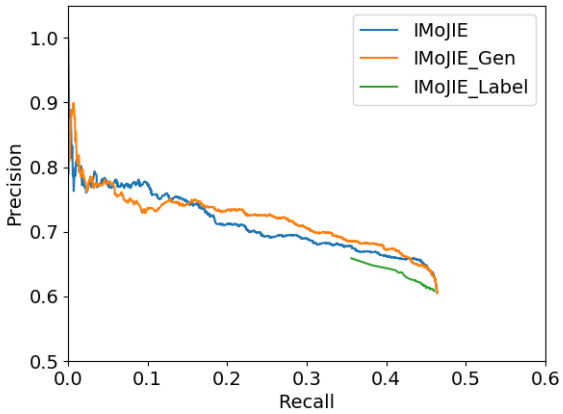

Figure 4.5: P-R curve of IMoJIE with no rescoring, label rescoring and generation rescoring.

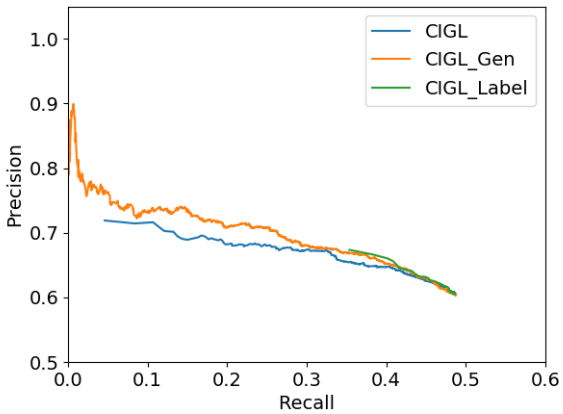

Figure 4.6: P-R curve of CIGL with no rescoring, label rescoring and generation rescoring.

Model	Model Confidence			Generation Rescore			Label Rescore		
	F1%	AUC%	AUC_{int}%	F1%	AUC%	AUC_{int}%	F1%	AUC%	AUC_{int}%
IMoJIE	53.3	33.1	33.3	53.3	33.5	33.6	52.4	36.2	30.1
CIGL	54.0	33.6	33.0	54.0	34.2	34.3	54.0	34.9	**33.4**
Gen2OIE	**54.4**	**34.2**	**34.4**	**54.5**	**35.5**	**35.6**	**54.5**	**38.9**	32.2

Table 4.3: The F1 and AUC scores of the three models – IMoJIE, CIGL and Gen2OIE using the original model confidence, generation rescoring and label rescoring.

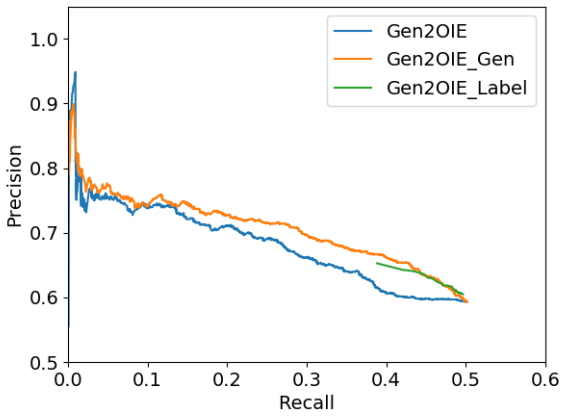

Figure 4.7: P-R curve of Gen2OIE with no rescoring, label rescoring and generation rescoring.

(IGL-OIE) and all constraints (CIGL-OIE). In Table 4.4, we report the CaRB metric, and also report the number of constraint violations in each scenario.

System	CaRB	Constraint Violations					Num. of Extractions
	F1%	POSC	HVC	HVE	EC	HVC+HVE+EC	
IGL-OIE	52.4	1494	375	**128**	284	787	1401
IGL-OIE (POSC)	49.6	**396**	303	200	**243**	746	**1577**
IGL-OIE (HVC,HVE,EC)	53.2	1170	295	144	246	**655**	1509
CIGL-OIE	**54.0**	766	**274**	157	237	668	1531
Gold	100	371	324	272	224	820	2714

Table 4.4: Performance and the number of constraint violations for training with different sets of constraints. CIGL-OIE represents training IGL architecture-based Open IE extractor with all the constraints: POSC, HVC, HVE and EC.

Training IGL architecture-based Open IE extractor with POSC constraint (IGL-OIE (POSC)), leads to a reduction in POSC violations. However, the number of violations of (HVC+HVE+EC) remains high. On the other hand, training only with head-verb constraints (HVC, HVE, EC) reduces their violations but the POSC violations remain high. Hence, we find that training with all the constraints achieves the best performance. Compared to IGL-OIE, it reduces the POSC violation from 1494 to 766 and (HVC+HVE+EC) violations from 787 to 668. The higher violations of Gold may be attributed to an overall larger number of extractions in the reference

set.

4.5.3 Performance using different metrics

We also evaluate the performance of the proposed models with OIE16 and Wire57 metrics evaluated on the same CaRB dataset. They are denoted as OIE16-C and Wire57-C, respectively. We report the results in Table 4.5. We find that CIGL-OIE is consistently able to reach the F1 performance of Gen2OIE while being significantly faster. Since label-rescoring is not involved in these experiments, we continue to use the normal AUC metric, which was observed to be within 0.5% of AUC_{int} metric for the Model Confidence and Generation Rescore methods in Table 4.3.

System	CaRB		OIE16-C		Wire57-C	Speed
	F1%	AUC%	F1%	AUC%	F1%	Sentences/sec.
IMoJIE	53.2	33.1	56.1	38.1	34.9	2.6
Gen2OIE	**54.4**	**34.2**	**60.5**	40.0	**37.1**	0.6
IGL-OIE	52.5	31.7	55.4	36.5	34.9	**142.0**
CIGL-OIE	54.0	33.6	59.6	**40.7**	36.8	**142.0**

Table 4.5: Evaluation of IMoJIE, Gen2OIE, IGL-OIE and CIGL-OIE using different metrics proposed for Open IE.

4.6 Conclusion

In this chapter, we introduce a novel way of modelling the task of Open IE using the Iterative Grid Labeling (IGL) architecture. We further improve the coverage of generated extractions using carefully designed constraints applied at the training time. The final Constrained Iterative Grid Labeling (CIGL) model achieves performance close to that of the best generation model, Gen2OIE (with a modest decrease of only 0.4% F1) while being significantly faster. The CIGL model can process sentences at a rate of 142 per second, which is 236× faster than Gen2OIE. IGL architecture has value beyond Open IE and can be helpful in tasks where a set of labelings for a sentence is desired, especially when labelings have dependencies amongst them. We showcase another application of IGL for the task of coordination analysis in Section 5.1.

We also demonstrate the utility of using separate models for rescoring the generated extractions. Experimentation with generative rescoring leads to an improvement in the AUC of the final set of extractions for the CIGL architecture.

In the next chapter, we look at improving the handling of specific linguistic phenomena that current Open IE systems struggle with, such as coordination and noun compounds.

Chapter 5

Handling of Linguistic Phenomena in Open IE

In Chapter 3 and Chapter 4, we focused on building better neural models for the task of Open IE, intending to build faster and more accurate systems. However, apart from the neural model used, the final system is also constrained by the data used for training the model. Since the training data only represents a sample of the entire distribution of possible examples, certain linguistic phenomena may be underrepresented or even completely absent. In this chapter, we explore two such linguistic phenomena which we have identified as sources of potential improvement for existing Open IE models:

1. Coordinations (such as "*and*", "*but*") are often used to compose simple statements to express a more complex idea. Open IE systems benefit from special handling of these structures to ensure that the generated extractions are as atomic as possible while maintaining faithfulness (Saha and Mausam, 2018). In Section 5.1, we build a new coordination analyzer to accurately extract the various parts of a coordination structure and use them to generate better Open IE extractions.

2. Noun Compounds (such as *Covid vaccine*) are a commonly occurring construct in the English language used to represent a more elaborate phrase (such as *a vaccine used to protect against Covid disease*) in a shortened manner. The longer phrase is also referred to as an interpretation of the noun compound. Adding these interpretations to Open IE extractions can lead to the discovery of implicit relations that are not mentioned explicitly in the sentence. In Section 5.2, we introduce a new task of interpreting proper noun compounds (where the first part of the compound is a proper noun). We collect a new dataset and build models for the task and develop an integrated mechanism to include the interpretations in Open IE systems.

Finally, in Section 5.3, we conclude this chapter by describing the construction of a new Open IE system that integrates all the advances introduced in Chapter 3, Chapter 4 and Chapter 5 into one common framework.

5.1 Coordinations

Coordinated conjunctions (CC) are conjunctions such as *"and"* and *"or"* that connect or coordinate words, phrases, or clauses (which are called the *conjuncts*). Sentences can have hierarchical

Sentence	Other signs of lens subluxation include mild conjunctival redness, vitreous humour degeneration, and an increase or decrease of anterior chamber depth .
CIGL-OIE	(Other signs of lens subluxation; include; mild conjunctival redness, vitreous humour degeneration, and an increase or decrease of anterior chamber depth)
CIGL-OIE +Coordination Analyzer	(Other signs of lens subluxation; include; mild conjunctival redness) (Other signs of lens subluxation; include; vitreous humour degeneration) (Other signs of lens subluxation; include; an increase of anterior chamber depth) (Other signs of lens subluxation; include; an decrease of anterior chamber depth)

Table 5.1: For the given sentence, IGL based Open IE extractor produces an incomplete extraction. Constraints improve recall by covering the remaining words. Coordination Analyzer handles hierarchical conjunctions.

coordinations, i.e., some coordination structures nested within the conjunct span of others (Saha and Mausam, 2018). The goal of coordination analysis is to detect coordination structures — the coordinating conjunctions along with their constituent conjuncts. We observe that existing neural Open IE models struggle in handling coordination structures and do not split conjunctive extractions properly. An example is shown in Table 5.1 where the Open IE model, CIGL-OIE, is unable to separate the various signs of the disease into different extractions. In response, we first design a new coordination analyzer. It is built with the IGL (Chapter 4) architecture by interpreting each row in the 2-D grid as a coordination structure. This leads to a new state of the art on this task, with a 12.3 pts improvement in F1 over the previous best-reported result (Teranishi et al., 2019), and a 1.8 pts gain in F1 over a strong BERT baseline. We then combine the output of our coordination analyzer with our Open IE extractor, resulting in a further increase in performance (Table 5.1).

5.1.1 Coordination Analyzer

We observe that coordination analysis can be posed as a hierarchical labeling problem. This is because coordination structures can be seen as labeling over the words in the sequence with their appropriate tags to indicate the particular parts of the coordination structure. Further, nested coordination structures can be treated as multiple levels of labeling. This is illustrated with an example in Figure 5.1, where the sentence "Jeff founded Amazon and Blue Origin and invested in Google" has three coordinate structures with the three "*and*'s" as the coordinating conjunctions (*CC*) and their respective conjunctions spans (*CONJ*). The "*and*" with "founded Amazon and Blue Origin' and "invested in Google, Grail and ZocDoc" as the conjunct spans has the remaining two nested conjunct structures within it. Hence, the first top-level coordination structure is extracted in the level 1 of the labeling (L1) while the remaining two coordination structures are extracted in the level 2 (L2).

Therefore, we formulate a 2-D grid labeling problem, where all coordination structures at the same hierarchical level are predicted in the same row. Specifically, we define a grid of size (M, N), where M is the maximum depth of hierarchy, and N is the number of words in the sentence. The value at position (m, n) in the grid represents the label assigned to the n^{th} word in the m^{th} hierarchical level, which can be *CC* (coordinating conjunction), *CONJ* (belonging to a conjunct span), or *N* (None). We can use the IGL architecture, introduced in Chapter 4 for labeling this grid. This gives us an end-to-end coordination analyzer that can detect multiple coordination structures with two or more conjuncts. We refer to this coordination analyzer as

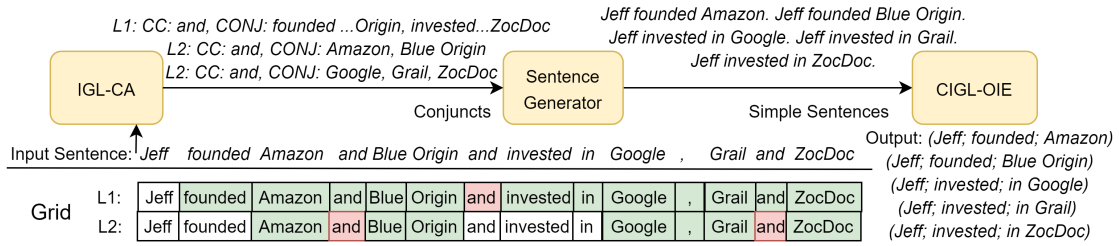

Figure 5.1: IGL-CA identifies conjunct boundaries by labeling a 2-D grid. This generates simple sentences, and CIGL-OIE emits the final extractions.

IGL-CA.[1]

5.1.1.1 Experimental Setup

We train and evaluate the quality of cooridnation analyzer on the coordination-annotated Penn Tree Bank (PTB) (Ficler and Goldberg, 2016b). We compute the precision, recall and F1 of the predicted conjunct spans. We compare our end-to-end IGL-CA model against previous state-of-the-art coordination analyzers. Teranishi et al. (2017) uses neural parsers trained on the PTB data using similarity and replaceability of conjuncts as features. Teranishi et al. (2019) independently detects coordinator, begin, and end of conjuncts and does joint inference using Cocke–Younger–Kasami (CYK) parsing over context-free grammar (CFG) rules.

5.1.1.2 Experiments

How does our coordination analyzer compare against other analyzers?

We evaluate two variants of our IGL architecture-based coordination analyzer (IGL-CA) – using BERT-Base and BERT-Large. In Table 5.2, we find that both BERT-Base and BERT-Large variants outperform the previous state-of-art (Teranishi et al., 2019) by 9.4 and 12.3 F1 points, respectively. For a fair comparison, we train a stronger variant of Teranishi et al. (2019), replacing the LSTM encoder with BERT-Base and BERT-Large. Even in these settings, IGL-CA performs better by 1.8 and 1.3 F1 points, respectively, highlighting the significance of our IGL architecture. Overall, IGL-CA establishes a new state of the art for the task of coordination analysis due to the use of an end-to-end neural network that avoids the pipeline errors that arise in the prior systems.

5.1.2 Coordination Analyzer in Open IE

Conjuncts in a coordinate structure exhibit *replaceability* – a sentence is still coherent and consistent if we replace a coordination structure with any of its conjuncts (Ficler and Goldberg, 2016c). Following CalmIE's Section 2.3.1.7 approach, we generate simple (non-conjunctive) sentences using IGL-CA. For example, the following simple sentences are generated from the example in Figure 5.1, "Jeff founded Amazon", "Jeff founded Blue Origin", "Jeff invested in Google", "Jeff invested in Grail" and "Jeff invested in ZocDoc". We also use some heuristics to determine cases where the sentence should not be split. For example, if the word "between" preceeds the coordination structure, then we do not split it. We then run CIGL-OIE on these

[1]The code and models are released at https://github.com/dair-iitd/openie6

System	Precision	Recall	F1
(Teranishi et al., 2017)	71.5	70.7	71.0
(Teranishi et al., 2019)	75.3	75.6	75.5
BERT-Base:			
(Teranishi et al., 2019)	83.1	83.2	83.1
IGL-CA	86.3	83.6	84.9
BERT-Large:			
(Teranishi et al., 2019)	86.4	86.6	86.5
IGL-CA	**88.1**	**87.4**	**87.8**

Table 5.2: P, R, F1 of the system evaluated on Penn Tree Bank for different systems. We use both BERT-Base and BERT-Large as the encoder

	System 1 (P, R, F1)	System 2 (P, R, F1)
Talks resumed between USA and China Gold: (Talks; resumed between; USA and China)	(Talks; resumed between; USA) (Talks; resumed between; China) CaRB: (50.0, 66.7, 57.1) CaRB (1-1): (50.0, 66.7, 57.1)	(Talks; resumed between; USA and China) CaRB: (100, 100, 100) CaRB (1-1): (100, 100, 100)
I ate an apple and orange Gold: (I; ate; an apple) (I; ate; an orange)	(I; ate; an apple) (I; ate; an orange) CaRB: (100, 100, 100) CaRB (1-1): (100, 100, 100)	(I; ate; an apple and an orange) CaRB: (57.1, 100, **72.7**) CaRB (1-1): (53.5, 50.0, **57.1**)

Table 5.3: Evaluation of CaRB and CaRB(1-1) on two sentences. CaRB under-penalizes Open IE systems for incorrect coordination split by giving a recall of 100% for the second example of System 2. On the other hand, CaRB(1-1) reports the recall as 50% in the second example for System 2.

simple sentences to generate the extractions. These extractions are de-duplicated and merged to yield the final extraction set for the original sentence. We call this combined system, OpenIE-6.

For a conjunctive sentence, CIGL-OIE's confidence values for extractions will be with respect to multiple simple sentences extracted from the original input, and may not be calibrated across them. Therefore, we use a separate confidence estimator (described in Section 4.3). It computes a log-likelihood for every extraction w.r.t. the *original* sentence — this serves as a better confidence measure for the generated extractions.

5.1.2.1 Evaluation

Apart from using CaRB metric for evaluating the generated Open IE extractions, we additionally introduce a new metric, CaRB(1-1), a variant of CaRB that retains CaRB's similarity computation but uses a one-to-one mapping for both precision and recall. We find experimentally that CaRB(1-1) is a better metric for evaluating conjunctive sentences.

CaRB on Conjunctive Sentences: We analyze the issues with using CaRB on conjunctive sentences and the motivating factors for developing CaRB(1-1).

Coordinate structure in conjunctive sentences are of two types (Shaw, 1998):

- *Combinatory*, where splitting the sentence by replacing the coordinate structure with one

of the conjuncts can lead to incoherent extractions. E.g. splitting "*Talks resumed between USA and China*" will give *(Talks; resumed; between USA)*.

- *Segregatory*, where splitting on the coordinate structure can lead to shorter and more coherent extractions. E.g. splitting "*I ate an apple and orange.*" gives *(I; ate; an apple)* and *(I; ate; an orange)*.

Combinatory coordinate structures are hard to detect (in some cases, even for humans). Some systems (ClausIE, CalmIE and ours) use some heuristics, such as not splitting if the coordinate structure is preceded by "*between*". In all other cases, the coordinate structure is treated as segregatory and is split.

The human-annotated gold labels of the CaRB dataset correctly handle conjunctive sentences in most cases. However, we find that compared to the scoring function of OIE2016 and Wire57 (Section 2.2), CaRB under-penalizes systems for incorrectly splitting combinatory coordinate structures. We trace this issue to the difference in mapping used for recall computation (one-to-one vs many-to-one).

Consider two systems – System 1, which splits into all conjunctive sentences (without any heuristics), and System 2, which does not. For the sentence "*I ate an apple and orange*", the set of gold extractions are {*(I; ate; an apple)*, *(I; ate; orange)*}. System 2, which (incorrectly) does not split on the coordinate structure, gets a perfect recall score of 1.0, similar to System 1, which correctly splits the extractions (Table 5.3).

Due to this phenomenon, we find that CaRB does not sufficiently penalize current systems for not splitting on seggregatory coordinations while penalizing the systems for incorrectly splitting on combinatory coordinations. This leads to gains obtained from our system being overshadowed. To re-affirm this, we evaluate all the systems on **CaRB(1-1)**, a variant of CaRB which retains all the properties of CaRB, except that it uses one-to-one mapping for computing recall. This ensures that System 2 gets a recall of only 50 pts as the generated extraction can match only one of the two gold extractions for computing recall.

We notice that our CIGL-OIE+IGL-CA shows improvements in CaRB(1-1) and other metrics which use one-to-one mapping (OIE16, Wire57) (Table 4.2). However, it shows a decrease in the CaRB score. This demonstrates that the primary reason for the decrease in performance is the many-to-one mapping in CaRB.

However, we also observe that CaRB(1-1) is also not the best strategy for evaluation as it assigns an equal score to both the cases — splitting a combinatory coordinate structure and not splitting a segregatory coordinate structure (Table 5.3). This is also not desirable as a long extraction which is not split is better than two incorrectly split extractions. Hence, we consider that one-to-one mapping for computing recall over-penalizes splitting a combinatory coordinate structure.

Determining the right penalty, in this case, is an open-ended problem. We leave it to further research to design an optimal metric for evaluating conjunctive sentences for Open IE. In this experiment, we use CaRB(1-1) as the metric for better evaluating the quality of Open IE extractions in conjunctive sentences.

5.1.2.2 Experiments

How much does the coordination analyzer benefit Open IE systems?

In Table 5.6, we find that adding the coordination analyzer module (IGL-CA) to any of the three Open IE models — IMoJIE, Gen2OIE, and CIGL — leads to improvements in CaRB(1-1) score, while leading to a decrease in the original CaRB metric.

Coordination Analyzer	IMoJIE	CIGL-OIE
None	36.0	36.8
CalmIE	37.7	38.0
(Teranishi et al., 2019)	36.1	36.5
IGL-CA	**39.5**	**40.0**

Table 5.4: Wire57 F1 scores of IMoJIE and CIGL-OIE with addition of different coordination analyzers. IGL-CA improves both of the Open IE extractors.

System	Precision	Yield	Total Extrs
CIGL-OIE	77.9	131	174
CIGL-OIE + IGL-CA	**78.8**	**222**	**291**

Table 5.5: Manual comparison of Precision and Yield on 100 random conjunctive sentences from CaRB Gold.

As discussed, we notice that the current scoring function used in CaRB does not handle conjunctions properly. CaRB under-penalizes Open IE systems which do not split seggregatory coordinations by assigning them a higher recall while still penalizing Open IE systems that split on combinatory coordination splits. This is also evidenced in the lower CaRB scores for both OpenIE-5[2] (vs. OpenIE-4) and OpenIE-6 (vs. CIGL-OIE) — the two systems that focus on conjunctive sentences. We trace this issue to the difference in mapping used for recall computation (one-to-one vs many-to-one).

To resolve this variation in different scoring functions, we undertake a manual evaluation (discussed in detail in Section 5.1.2.3). Two annotators (authors of the paper), blind to the underlying systems (CIGL-OIE and OpenIE-6), independently label each extraction as correct or incorrect for a subset of 100 conjunctive sentences. Their inter-annotator agreement is 93.46%. After resolving the extractions where they differ, we report the precision and yield in Table 5.5. Here, yield is the number of correct extractions generated by a system. It is a surrogate for recall since its denominator, the number of all correct extractions, is hard to determine for Open IE.

We find that OpenIE-6 significantly increases the yield (1.7×) compared to CIGL-OIE, along with a marginal increase in precision. This result underscores the importance of splitting coordination structures for Open IE.

To affirm that the gains of better coordination analysis help the downstream Open IE task, we experiment by using different coordination analyzers with CIGL-OIE and IMoJIE. From Table 5.4, we see a considerable improvement in the downstream Open IE task using IGL-CA for both IMoJIE and CIGL-OIE, which we attribute to better conjunct-boundary detection capabilities of the model. For CIGL-OIE, this gives a 3.5 pts increase in Wire57-C F1, compared to using the CA from Teranishi et al. (2019). In Table 5.6, we find that the IGL-CA benefits the Gen2OIE system with an increase of 1.8 pts in F1 and 2.4 pts in AUC with the CaRB(1-1) metric.

5.1.2.3 Manual Comparison

The set of extractions from both the systems, CIGL-OIE and OpenIE-6 were considered for a random 100 conjunctive sentences from the validation set. We identify a conjunctive sentence

[2]OpenIE-5 uses CalmIE for conjunctive sentences.

System	CaRB		CaRB(1-1)	
	F1	AUC	F1	AUC
IMoJIE	52.5	36.2	41.6	24.8
IMoJIE + IGL-CA	50.9	34.9	44.4	27.5
Gen2OIE	**54.5**	**38.9**	43.8	26.7
Gen2OIE + IGL-CA	51.5	36.4	**45.6**	**29.1**
CIGL-OIE	54.0	35.7	42.8	24.6
CIGL-OIE + IGL-CA	50.6	35.1	45.0	28.4

Table 5.6: Adding a coordination analyzer, IGL-CA, to IMoJIE, Gen2OIE and CIGL, improves the score consistently in the CaRB(1-1) metric that is suitable for evaluating conjunctive sentences. Label rescoring is consistently used in all the experiments.

based on the predicted conjuncts of the coordination analyzer. The annotators are instructed to check if the extraction has well-formed arguments and is implied by the sentence.

A screenshot of the process is shown in Figure 5.2.

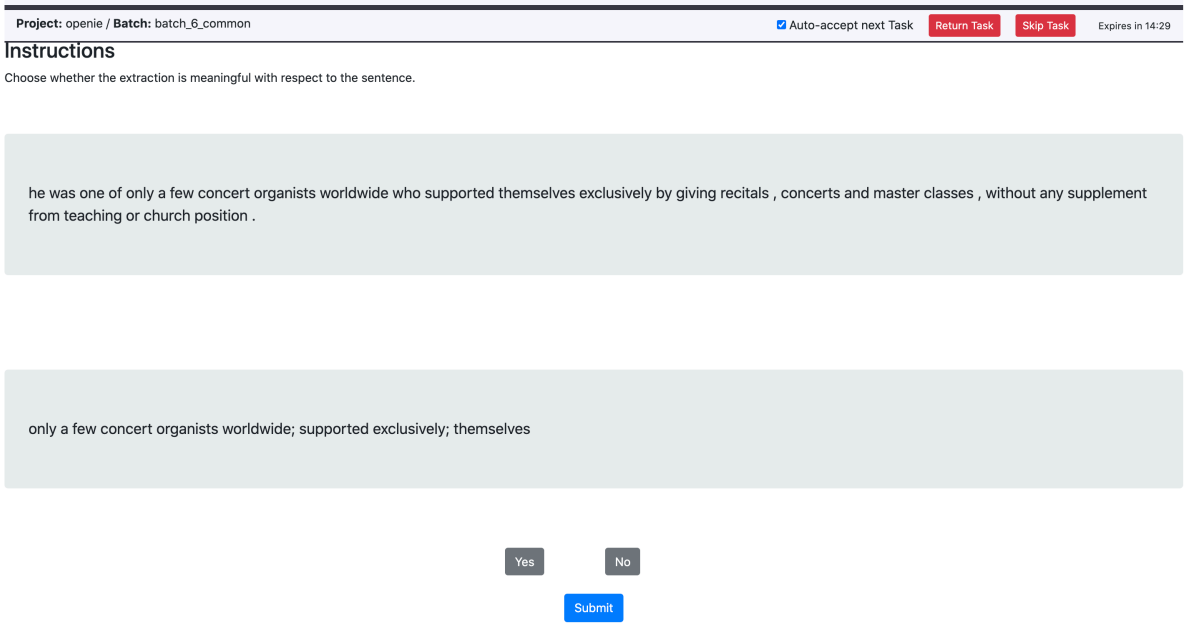

Figure 5.2: Process for manual comparison. Each extraction from both systems is presented to the annotator in a randomized order. The annotator checks if the extraction can be inferred from the original sentence and marks it accordingly.

5.1.3 Discussion

Breaking complex sentences into simpler sentences enables us to generate more accurate Open IE extractions, but in the process, we may lose the context of the other parts of the sentence. Therefore, using models from document-level Open IE (Yong et al., 2023) which are capable of reincorporating such context when generating extractions may prove advantageous in certain cases that involve complex relations that may span over multiple simple sentences or involve complex reasoning between them. For example, in the sentence 'Maria stored her valuables in a safe, and her son knew only about the key.', the extraction with an implicit relation (Her son;

Type	Example	Semantic Interpretations
Proper NC (*Proper-Common*)	*Shakespeare biography*	is a biography about Shakespeare
	London theatre	is a theatre in London *;* is a theatre located in London
	Concorde airplane	[NON-CMP] (Non-Compositional)
	Notre Dame cathedral	[NON-CMP] (Non-Compositional)
Common NC (*Common-Common*)	*nursing job*	is a job in nursing field *;* is a job involving nursing
	oil price	is price paid for the oil

Table 5.7: Examples of common and proper noun compounds along with their semantic interpretations ("*;*" separates multiple interpretations). [NON-CMP] indicates the absence of implicit relation between the constituent nouns.

did not know about; her valuables) would be missed by using simple sentences alone, even after co-reference resolution is used.

5.2 Proper Noun Compound Interpretation

In the previous section, we introduced techniques to handle coordinations within the framework of Open IE. In this section, similarly deal with another common linguistic structure, which are called proper noun compounds. *Proper noun compounds* (PNCs) (Breban et al., 2019)[3] are grammatical constructions where a proper noun is followed by a common noun, for example *Covid vaccines* or *Buddhist monks*. These often serve as a compact way to convey information about an already known entity, omitting predicates that are interpreted by the reader using surrounding context, common sense, and world knowledge. For example, a reader is likely to interpret that "Buddhist monks" are "*religious people who are* Buddhists". In other cases, PNCs are used to identify specific entities and do not provide additional information. For example, *Watergate scandal* and *Kawasaki disease* do not have any implicit relation between the proper and common nouns as they refer to a specific instance of a scandal and a disease. Table 5.7 provides additional examples.

Thanks to their brevity, PNCs are commonly used to shorten descriptions in space-constrained domains, such as news articles headlines (Breban et al., 2019). However, we find that prior work on compound noun interpretations only considered cases where the constituents are common nouns (e.g. *baby oil*), thus missing all of the information conveyed in proper noun compounds (Shwartz et al., 2018; Hendrickx et al., 2019).

To address this limitation in current systems, we design a two-stage method for PNC interpretation (Section 5.2.2). The first stage requires identifying whether a given PNC is compositional or not, while the second stage is the generation of an interpretation, where applicable.

In Section 5.2.3, we present PRONCI, a crowd-sourced dataset over Wikipedia containing 22.5K proper noun compounds and their annotated semantic interpretations. Candidates' PNCs are found using syntactic parsing and are then presented to crowd workers who are asked to interpret them. Our annotation interface marks whether workers needed to read the full sentence, thus identifying PNCs whose interpretation relies on context.

In Section 5.2.4, we develop two approaches for PNC interpretation: (1) a multi-task neural model that performs classification and sequence generation in two distinct stages and (2) a text-to-text approach, using a sequence-to-sequence model for both classification and gen-

[3]also referred to as proper noun modified compounds.

eration. We further experiment with different methods for injecting various sources of world knowledge, which seems crucial for the task, using external resources like Wikipedia and WordNet (Fellbaum, 2010), that give relevant information or definitions about the PNCs, that help in improving performance.

For evaluating the generated interpretations, we propose a combination of classification-based metrics and generation metrics to properly handle the interpretable and non-interpretable cases, respectively (Section 5.2.5). Since multiple correct interpretations are possible for a PNC, we use learned metrics such as BLEURT (Sellam et al., 2020), which is finetuned on human-annotated preferences.

In Section 5.2.7 we show that training on ProNCI yields models that can readily benefit extrinsic downstream application in the task of Open IE, thus widely extending its coverage. Our approach first automatically extracts PNC interpretations using our models, then introduces it explicitly back into an Open IE extraction using a sequence-to-sequence model, thus giving an interpretation-integrated extraction. We then apply a high precision rule to generate new relations which leads to a 7.5% increase in yield at an estimated precision of 85% on the added extractions, when compared to extractions generated from the original sentences themselves. A major advantage of this approach is that it is agnostic to the Open IE system being used.[4] Summarizing, the main contributions described in this section are:

1. We introduce the ProNCI dataset, containing interpretation for 22.5K proper noun compounds and their semantic interpretations.

2. We develop multi-task and generation-based neural baselines that can leverage external knowledge for achieving higher performance.

3. We propose metrics for evaluating the quality of generated semantic interpretations.

4. We demonstrate the extrinsic usefulness of our model in a downstream application by using the interpretations to augment the expressivity of Open Information Extraction systems.

5.2.1 Related Work

Noun compounds are commonly used in the English language, constituting 3.9% of the tokens in the Reuters corpus (Baldwin and Tanaka, 2004). They can be arbitrary length phrases, such as *split air conditioner*, but most prior work on interpreting noun compounds has primarily looked at two-word noun compounds of the type *noun-noun*, where both are common nouns. To the best of our knowledge challenges in interpretation where the first word is a proper noun (i.e., *proper noun compounds*) have not been addressed, although their functional analysis and prevalence in certain domains have been studied in linguistics (Rosenbach, 2007; Alexiadou, 2019; Breban et al., 2019). We briefly summarise the various types of noun-compound interpretations in literature and discuss their uses in applications.

Types of Interpretation: Various types of interpretations for noun compounds have been explored, covering classification, ranking and generation. Prior literature has frequently posed the interpretation as a **classification** task, where the classes can belong to abstract labels (Fares, 2016), semantic frame elements (Ponkiya et al., 2018) or prepositions (Lauer, 1995). However, none of these schemes can cover all range of possible noun compounds, thus limiting their

[4]The dataset and code are available at https://github.com/dair-iitd/pronci

expressivity and coverage. SemEval 2010 Task 9 (Butnariu et al., 2009) annotates human preferences for a set of 25-30 templatized paraphrases for each of the 250 training and 300 testing noun compounds. The task is framed as producing an accurate score for each paraphrase that **ranks** them in the correct order. SemEval 2013 Task 4 (Hendrickx et al., 2019) released a dataset of noun compounds and annotated free paraphrases for each compound. Participating models were evaluated by matching and scoring the **generated** predictions with the gold set.

Non-compositionality of common noun compounds has been defined in Yazdani et al. (2015); Reddy et al. (2011) as compounds whose meanings don't follow from their constituents (e.g., *sitting duck*, *acid duck*). However, for the case of semantic interpretation of proper noun compounds, we define it as the absence of an implicit relation between the constituents.

Ponkiya et al. (2020) is the current state of the art which poses the problem of generation of masked tokens using a pretrained T5 model (Raffel et al., 2020) to get free paraphrase interpretations in a completely unsupervised manner. This leads to better performance than techniques that use the available training data. However, with the PROPNCI dataset, we find that supervised models do outperform zero-shot models due to the scale of the dataset.

Applications: Noun compound interpretations have been helpful in the translation of noun compounds by either using a one-to-one mapping of interpreted prepositions (Paul et al., 2010) or using recursive translation patterns (Balyan and Chatterjee, 2015). In Question Answering systems, they have been used for disambiguating different types of noun-noun compounds in passage analysis (Ahn et al., 2005). They have also been useful for normalizing text that can help textual entailment (Nakov, 2013) and as auxiliary semantic annotation modules to improve parsing (Tratz, 2011). We also show their use in the task of Open IE.

5.2.2 Problem Definition

Interpretations of noun compounds are meant to expose the implicit relation. Free-form paraphrases as interpretations provide flexibility for expressing relations implied in noun compounds, overcoming the limitations associated with choosing from a fixed set of classes or templates at the cost of a possibly non-consolidated representation, i.e., where similar-meaning noun compounds are represented differently. Hence, we define the semantic interpretation of a PNC as a free-form paraphrase that exposes the implicit relation between the constituent nouns, if any relation exists, else identify it as non-compositional ([NON-CMP]).

$$\text{SemInt}(pnc) = \begin{cases} \text{Paraphrase}, & \text{if } \textit{reln.} \text{ exists} \\ \text{[NON-CMP]}, & \text{if } \textit{reln.} \text{ absent} \end{cases}$$

5.2.3 PROPNCI Dataset

To facilitate research on semantic understanding of proper noun compounds, we collect and release a supervised dataset called PROPNCI. It contains 22,500 PNCs and their semantic interpretations which were written by human workers hired from Amazon Mechanical Turk (AMT). Here we describe how PROPNCI was prepared.

The scale of the dataset is orders of magnitude greater than previously published free paraphrase (common) noun compound datasets like SemEval 2013 Task 4 (Hendrickx et al., 2019) that have only considered 355 noun compounds. For handling the evaluation of generated interpretations where multiple correct answers are possible, prior datasets choose to annotate multiple interpretations for each noun compound (varying from 30-50). On the other hand, PROPNCI

Task Instructions
1. Your goal is to describe the relation between the two words by filling in the blanks. 2. You can write up to five words (or less!) 3. The resulting relation should form a valid English sentence (see below for an example). 4. You can consult an example sentence as additional context, but the relation you write should be inferred only from the two words, and not using additional information. 5. If the compound is a name, entity, or location or if you can't describe the relation between the words, please leave the relation blank.
Examples
1. Coke Spokesman *is a worker of* Coke. 2. Leake government *is located in* Leake. 3. Capitol Hill
Pitfalls
1. Coke Spokesman *employment* Coke. The relation should form a valid sentence. 2. Leake government *has a failed* goverment. The relation should be inferred by the words themselves and not by additional context.

Table 5.8: Instructions for the task along with examples and common pitfalls that are provided to the human workers from AMT for constructing PRONCI dataset.

dataset only contains one interpretation per noun compound, because we choose to invest our annotation budget in breadth rather than depth. We rely on recent advances in semantic text similarity (e.g. BLEURT (Sellam et al., 2020)) to help evaluate the generated interpretations.

Moreover, prior datasets consider noun compounds out of context, while PRONCI also contains the sentence in which the proper noun compound is used. Providing this additional context helps to limit the ambiguity associated with multiple possible interpretations of the noun compound. For example, *U.S. sanctions* can mean either sanction imposed by U.S. or sanctions imposed on the U.S. The exact case can be determined based on the context in which it is used. "*U.S. sanctions* on Iran have crippled the country", implies the former and "*U.S. sanctions* by Iran..." implies the latter.

To prepare the PRONCI dataset, we randomly sample sentences from Wikipedia and retain sentences which contain two-word proper noun compounds as identified by the SpaCy dependency parser (Honnibal et al., 2019). For every word, SpaCy identifies the root word along with the dependency tag. The "compound" dependency tag is used if the word and its root are part of a compound word. Then the parts of speech of the first and second word of the compound are checked. If they are proper nouns ("PROPN") and common nouns ("NOUN") respectively, we identify them as a proper noun compound and include it. If any word pairs have been identified incorrectly as proper noun compounds, they are marked by annotators to indicate the absence of any relation.

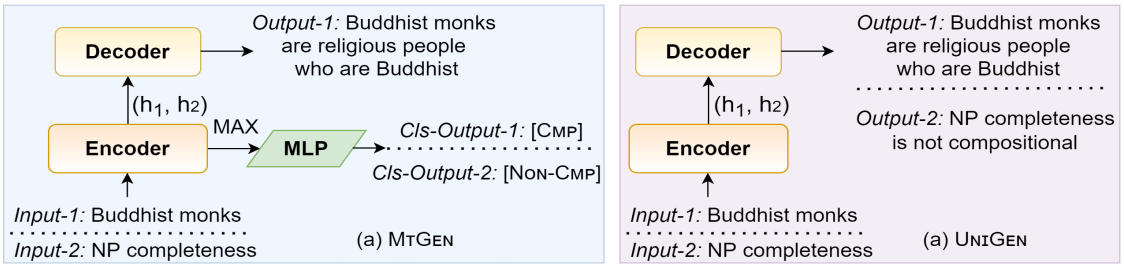

Figure 5.3: MTGEN (multi-task Seq2Seq model) classifies the example into (non) compositional classes and generates the interpretation where valid, while UNIGEN (unified generation model), uses a Seq2Seq model to generate interpretations or identify non-compositional examples using a specific string "is not compositional".

After the collection of proper noun compounds and corresponding sentences in which they appear, we posted Human Intelligence Tasks (HITs) on the AMT platform for the identification of the relation between two words. The HITs were accompanied by task instructions, summarized in Table 5.8. The workers were paid 9 USD per hour on average, based on initial annotation experiments which indicated an average annotation time of 20 seconds on each compound.

We split the collected 22,500 examples into training, validation and testing examples (see Section 5.2.3). To check the quality of the annotation, we randomly sample 100 examples from the validation set and get the annotations verified independently by three NLP experts. At least two of the experts agree with the annotated interpretation 93% of the time, which represents an acceptable level, considering the difficulty of understanding certain compounds that need technical knowledge (*AES key*) or cultural background (*Abner characters*), as well as the subjectiveness in determining non-compositionality.

Knowledge	**Example**
None	Buddhist monks
Sentence	Recent visitors to the campus include Buddhist monks who installed an environmental artwork at Lower Pond. [*SEP*] Buddhist monks
WordNet-NN	Buddhist meaning: Buddhism is a widespread Asian religion based on a series of original teachings attributed to Gautama Buddha. [*SEP*] Buddhist monks
Wiki-NNP	monks meaning: a religious male living in a cloister and devoting himself to contemplation and prayer and work [*SEP*] Buddhist monks
NER-NNP	Buddhist belongs to nationalities or religious groups [*SEP*] Buddhist monks

Table 5.9: Examples demonstrating the addition of different sources of knowledge for the compound, "Buddhist monks", in form of prompts that are concatenated with [*SEP*] token. NNP and NN correspond for information about proper and common nouns respectively, which can be from WordNet, Named Entity tags or Wikipedia.

5.2.4 Models

The task of semantic interpretation of proper noun compounds involves generating valid paraphrases that explicate the relation in compositional cases. So a model designed for this task needs to first identify if the given noun compound is compositional ([CMP]) or not ([NON-CMP]), and generate a paraphrase accordingly. We experiment with (1) supervised neural models, (2) adding external information and (3) zero/few-shot prompting models.

Supervised neural models: We use two types of supervised neural models: (1) a multi-task and (2) a unified generative model. Both models are depicted in Figure 5.3. The multi-task neural model uses a single model to perform both the tasks of classification as well as generation. For classification, the model uses the max-pooled representations of encoder hidden states that are passed to an MLP (Maini et al., 2020) to get the corresponding class probabilities of [CMP] and [NON-CMP]. In case the example is classified as compositional, a decoder is used for generating the paraphrase. We refer to this model as MTGEN.

In the unified generation model, we follow the recent advances in NLP where multiple tasks are posed in a common text-to-text format and are handled by a single Seq2Seq model like T5 (Raffel et al., 2020). For this purpose, we pose the task as a simple string generation problem that outputs either the paraphrase itself in cases where it is interpretable or generates the string "*proper noun compound* is non-compositional" in the remaining cases. We refer to this model as UNIGEN.

External Information: Since the task of interpretation requires knowledge of the noun compound, we also experiment with adding different types of knowledge to the model that help it in generating accurate interpretations. Various methods have been proposed to incorporate external knowledge into pre-trained language models (Wang et al., 2020; Liu et al., 2022b; Verga et al., 2021). We use a simple strategy of concatenating the knowledge along with the proper noun compound before passing it to the model. A [*SEP*] token is added to demarcate the added knowledge.

We use four sources of knowledge that provide further information about the noun compound. They include information on the proper noun, from (1) the first paragraph of the Wikipedia page associated with the entity linked to the mention represented in the text by an entity linking system (Wiki-NNP), (2) tags assigned to it by the Named Entity Recognition system (NER-NNP), or include information about the common noun using (3) the corresponding synset definitions provided in Wordnet (WordNet-NN), or information about the entire compound based on (4) the sentence in which it is used. An example of each type of knowledge is shown in Table 5.9.

Zero/Few-Shot Prompting: Prior techniques for noun compound interpretation such as (Ponkiya et al., 2020) have proposed zero-shot generation using pre-trained language models to achieve state-of-art performance on SemEval 2013 Task 4 (Hendrickx et al., 2019) and SemEval 2010 Task 9 (Butnariu et al., 2009). We, therefore, evaluate the performance of such techniques along with some extensions using few-shot learning on the PRONCI dataset. We find that they lead to a significant decrease in performance compared to finetuning on the supervised dataset, demonstrating the importance of having a large-scale dataset for the task of PNC interpretation.

5.2.5 Experimental Setup

Data Splits: The 22,500 examples of PRONCI are split into the train, validation and test such that all compounds with the same common noun occur exclusively in a single set. Such splitting ensures that there is no intersecting common noun in either the train or evaluation splits. This results in a more challenging setting than splitting the examples randomly, for which results are shown in Section 5.2.6.5. Further, we also consider subsets that contain only compositional examples (CMP) or only non-compositional examples (Non-CMP). The number of examples in each case is shown in Table 5.10.

Type	#Train	#Validation	#Test	#Total
CMP	9,722	1,416	2,497	14,389
Non-CMP	5,568	934	1,609	8,111
All	15,290	2,350	4,106	22,500

Table 5.10: The number of training, validation and testing examples in the PRONCI dataset. CMP indicates the subset that contains only compositional examples and constitutes 63.9% of the examples. Non-CMP indicates the complementary subset that contains only non-compositional examples and constitutes the remaining 36.1% of the examples.

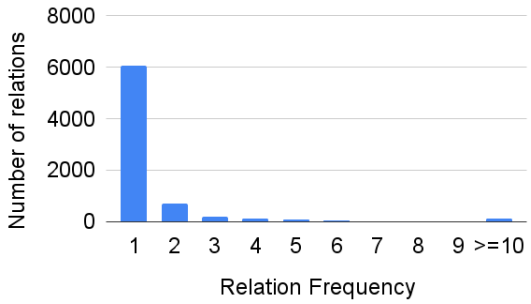

Figure 5.4: The plot of relation distribution in the PRONCI dataset. It shows the number of relations that have a frequency of 1 to 9 and >=10.

The dataset has 7,383 unique relations, with every relation occurring in an average of 1.84 examples. It contains 6,061 relations that occur only once in the dataset, as shown in Figure 5.4. The top 5 most commonly occurring relations along with their frequency (indicated in brackets) are *is located in* (560), *is based in* (389), *are relatives of* (245), *is an area of* (215) and *are located in* (125).

Evaluation metrics: Since the task involves a combination of classification and generation, the evaluation metric uses either an exact match or semantic similarity depending on the type of instance. If an instance has either the model prediction (**p**) or the gold annotation (**g**) as non-compositional, then an exact match (EX-MATCH) between the prediction and gold gives a binary score of 0 or 1. In examples where both the gold annotation and model prediction are compositional, a semantic matching algorithm (SEM-MATCH) is used to give a score between 0 and 1 which indicates the extent of their similarity.

$$\text{Score}(\mathbf{g}, \mathbf{p}) = \begin{cases} \text{SEM-MATCH}(\mathbf{g}, \mathbf{p}), \text{if gold label is CMP} \\ \text{EX-MATCH}(\mathbf{g}, \mathbf{p}), \text{if gold label is Non-CMP.} \end{cases}$$

In particular, we compare two alternatives for SEM-MATCH: (1) the popular BLEU score (Papineni et al., 2002), which relies on *n-gram* overlap and is often used in machine translations; and (2) BLEURT (Sellam et al., 2020), which is a finetuned soft-matching function that builds upon a pretrained language model. BLEURT represents a recent trend in trained evaluation metrics for text generation tasks.

For both alternatives, we use the entire paraphrase to evaluate the semantic score. Evaluating only the relations does not suit metrics such as BLEURT, which expects a well-formed sentence to infer the semantic meaning. Evaluation of the quality of BLUERT for the similarity between predicted and gold paraphrases using 1K human-annotated judgements indicates a 0.57 Pearson

and 0.56 Kendall correlation. We follow standard protocols in evaluating metric quality, as used in WMT Metrics shared tasks, and ask human annotators to rate the compositional model predictions as good, average and bad and see how these judgement scores correlate with the BLEU and BLEURT scores. Further details are provided in Section 5.2.6.4.

We denote the final evaluation metric as SEM/EX-MATCH. When using BLEU or BLEURT as the semantic matcher, the metric is also referred to as BLEU/EX or BLEURT/EX, respectively. To understand the effect of each type of match, we also report the EX-MATCH classification accuracy over all the examples, where the compositional type is assigned the positive class, and the non-compositional type is assigned as the negative class. Along with binary accuracy, we compute the precision and recall as well. Since the SEM-MATCH cannot be computed over all examples, we report the scores averaged over only the cases where both gold and prediction are compositional.

Pre-trained models: For all our experiments, we use the T5-base (Raffel et al., 2020) as the default initialization, unless explicitly mentioned otherwise. It contains 220M parameters. For checking the statistical consistency, every model is trained five times with different seeds and their mean and standard deviation are reported.

Hyper-parameters and computational resources: We run all our experiments using a V100 GPU. We use the standard hyper-parameters recommended in T5 for all the experiments, using a batch size of 16, the initial learning rate of 2e-5. The final model is chosen using early stopping on the validation set after training for 10 epochs. Each round of training and evaluation takes around one hour.

5.2.6 Experimental Results

In this section, we address three primary questions:

1. How do UNIGEN and MTGEN compare with each other and what benefit does adding external knowledge provide to these models?
2. What is the performance difference between few-shot learning and supervised training?
3. How do individual components of the noun compound influence the model predictions?

We also study the quality of evaluation metrics used, the performance on a random split of PRONCI, the effect of pre-training, effect of adding multiple sources of knowledge and analyze the mistakes made by the current model.

5.2.6.1 Performance of Supervised Models

In Table 5.11, we show the results of both, the multi-task model, MTGEN and the unified generation model, UNIGEN (Section 5.2.4).

We find that the UNIGEN model outperforms the MTGEN model in overall performance but leads to a modest drop in the compositionality classification performance. For example, in the case where no additional knowledge is used, UNIGEN leads to a higher SEM/EX-MATCH score with both BLEU and BLEURT leading to an increase of (2.4, 1.1) pts. But UNIGEN achieves a lower classification score with the EX-MATCH accuracy reducing by 0.8%. We attribute this observation to the fact that MTGEN uses a separate module that enables it to be tuned better for

Model	Knowledge	Ex-Match			Sem-Match		Sem/Ex-Match	
		Precision	Recall	Accuracy	BLEU	BLEURT	BLEU	BLEURT
MtGen	None	**79.1** ± 1.37	67.1 ± 1.84	**79.5** ± 0.58	32.7 ± 1.61	57.9 ± 0.42	44.3 ± 1.05	57.5 ± 0.66
	Sentence	78.1 ± 1.51	68.4 ± 2.50	79.4 ± 0.25	34.7 ± 0.36	58.3 ± 0.76	45.7 ± 0.60	57.8 ± 0.75
	WordNet-NN	74.2 ± 3.71	76.4 ± 5.68	79.4 ± 1.08	33.2 ± 1.08	57.6 ± 0.51	47.1 ± 0.92	58.9 ± 0.76
	Wiki-NNP	52.8 ± 2.43	**90.6** ± 3.02	63.2 ± 2.96	24.0 ± 0.36	32.9 ± 2.38	43.0 ± 0.50	45.4 ± 0.98
	NER-NNP	**79.1** ± 0.63	67.7 ± 1.63	79.7 ± 0.55	34.5 ± 0.23	59.2 ± 0.37	45.4 ± 0.51	58.3 ± 0.68
UniGen	None	73.5 ± 2.99	74.4 ± 2.26	78.7 ± 1.40	34.1 ± 1.99	58.6 ± 0.78	46.7 ± 1.12	58.6 ± 0.94
	Sentence	73.0 ± 1.57	77.6 ± 1.83	79.3 ± 0.55	34.4 ± 0.81	58.8 ± 0.68	**47.9** ± 0.41	**59.5** ± 0.57
	WordNet-NN	65.3 ± 5.76	82.9 ± 5.05	74.5 ± 3.74	33.7 ± 0.88	56.5 ± 0.65	47.4 ± 0.45	56.7 ± 1.52
	Wiki-NNP	65.3 ± 3.05	66.3 ± 5.50	71.8 ± 1.32	25.7 ± 0.59	37.8 ± 2.13	38.4 ± 1.55	43.9 ± 1.09
	NER-NNP	75.7 ± 0.95	72.3 ± 1.52	79.4 ± 0.21	**35.2** ± 0.23	**59.4** ± 0.40	46.9 ± 0.45	59.0 ± 0.42

Table 5.11: Performance of MtGen and UniGen on the ProNCI dataset trained under five different knowledge settings. All the models are evaluated using the three types of matching. 'None' corresponds to using no external knowledge. Adding external knowledge improves the performance of the models in three out of four cases.

the classification task. However, UniGen performs better in overall performance as both the encoder and decoder can benefit from positive transfer between the tasks.

By adding knowledge to the model, using the prompting described in Table 5.9, at both training and testing time, we see gains in performance in three out of four types of knowledge added. Using information of the proper noun from Wikipedia often reduces the performance due to incorrect entity linking. Among the remaining three sources of knowledge, we find that WordNet-NN leads to the maximum increase in performance in three of the four settings. We find that the predicted interpretations are often biased to re-use words that occur in the knowledge prompts and this leads to higher scores in the case of less frequently occurring compounds. For instance, the prediction changes from "Kirati community is a group of Kirati" to "Kirati community are people of Kirati", when added with the knowledge, "Major groups of Kirati community follows Buddhism". Using student paired t-test we find that improvements are statistically significant with p-value of $3.78e^{-10}$ of BLEURT scores averaged over all 5 seeds. We do not find additional improvements when multiple knowledge sources are added simultaneously (Section 5.2.6.7).

Predictions of UniGen trained with sentence knowledge are rated to be 72% correct when checked manually on a sample of 100 sentences. This indicates a significant scope for improvement, when compared to the upper bound of 93% data quality (Section 5.2.3).

We conduct two further experiments on the trained UniGen model to understand the strength of semantic matching used and the effectiveness of the model on the related task of common noun compound interpretation.

Template scoring: To test the effect of template words (i.e, 'is', 'of', 'noun-compound' and 'common-noun'), on BLEU and BLEURT scores, we use an output which contains a dummy relation: i.e., the prediction for every non-compositional example is forced to be 'noun-compound is none of common-noun'. This ensures that only template words match, but the semantic meaning is wrong. On re-computing the Sem-Match scores of UniGen, this reduces the BLEU score from 34.1 to 22.9 and the BLEURT score from 46.7 to -3. This follows the expected trend as BLEU gives partial scores to template matches, but BLEURT focuses on the overall semantic meaning.

Model	Ex-Match			Sem-Match		Sem/Ex-Match	
	Precision	Recall	Accuracy	BLEU	BLEURT	BLEU	BLEURT
Ponkiya et al. (2020)	0.0	0.0	60.8	23.1	44.9	13.8	26.8
Rand Few-Shot (5)	37.3	11.0	55.3	27.7	40.2	18.5	25.1
Rand Few-Shot (10)	62.1	21.4	58.2	27.6	39.3	22.3	28.2
KNN Few-Shot (5)	68.7	43.6	69.1	**29.9**	46.1	33.1	41.4
KNN Few-Shot (10)	**67.1**	**50.5**	**69.9**	**29.9**	**46.9**	**35.2**	**43.7**

Table 5.12: Performance of T5 model without any finetuning. Ponkiya et al. (2020) corresponds to the zero-shot setting adapted from the corresponding paper. Few-shot techniques use either five or ten example demonstrations. In 'Rand' the few-shot examples are chosen randomly while in 'KNN' the nearest neighbours of the query are chosen as the few-shot examples. Availability of annotated examples from ProNCI helps to substantially improve the performance of the model. Overall performance remains inferior to the finetuned models.

SemEval evaluation: When UniGen is evaluated on the free noun compound paraphrasing task of SemEval 2013 Task 4 (Hendrickx et al., 2019), it achieves a result of 72.8 compared to 80.1 on the isomorphic scoring used by Ponkiya et al. (2018). We attribute this to different interpretation styles with ProNCI focusing on detailed relations (average length of 6.9 words) compared to SemEval (average length of 5.1 words), leading to slightly lower scores with word match heuristics adopted by the task.

5.2.6.2 Performance of few-shot learning

State-of-the-art models for free paraphrasal interpretations of *common* noun compounds (Ponkiya et al., 2020) use the zero-shot generation capabilities of T5. They find that the performance exceeds that of supervised models. To check if the same holds for the ProNCI dataset, we also experiment with zero-shot generation. Similar to Ponkiya et al. (2020), we use the masked template, "$w_1 w_2$ is a <*extra_id_0*> the w_1", where T5 fills in the missing words in place of <*extra_id_0*>.

We further experiment with few-shot learning, where *K* training examples are chosen as part of the prompt which the model can use to perform in-context learning and generate the prediction for the given input. No additional knowledge is used in this set of experiments. These *K* examples can either be chosen randomly or the nearest neighbours to the input query can be chosen, where the cosine distance between the input and a training example is computed after embedding them with a pre-trained T5-Encoder (Liu et al., 2022a). We experiment with K = 5 or 10. The limitations of context size in the pretrained models prevent us from testing with higher values of *K*.

In Table 5.12, we find that the zero-shot performance trails behind the best few-shot model with a decrease of 21.4 and 41 pts in BLEU/Ex and BLEURT/Ex, respectively. This is partly because of the variety of examples in the ProNCI dataset, which cannot be fit into specific templates and the inability of the zero-shot model to handle non-compositional examples. Ponkiya et al. (2020) cannot detect non-compositional cases and hence achieves a score of zero in the Ex-Match metrics. In few-shot learning, expanding the prompt size and dynamically choosing the prompt examples helps achieve higher performance but the performance remains lower than the fully supervised UniGen model which is still 11.2, 15.3 pts higher in BLEU/Ex, BLEURT/Ex.

Shuffle	Ex-Match	Sem/Ex-Match	
	Accuracy	BLEU	BLEURT
None	**78.7** ± 1.40	**46.7** ± 1.12	**58.6** ± 0.94
NNP	62.4 ± 0.97	43.6 ± 1.01	50.6 ± 0.44
NN	43.7 ± 1.02	40.9 ± 0.15	41.0 ± 0.16

Table 5.13: UniGen evaluated after random shuffling of characters in the proper (NNP) or common (NN) noun.

5.2.6.3 Proper noun vs. Common noun

The interpretation of a proper noun compound depends on both the proper noun and common noun present in it. To study how each of the two nouns influences the prediction, we randomly shuffle their characters in both input and gold annotation. For example, to study the effect of proper noun, the characters of "Buddhist" in "*Buddhist monks*" are randomly shuffled to give the new proper noun compound, "*Dudhsitb monks*" whose interpretation is generated.

In Table 5.13, we find that common noun has a larger effect on the model performance as shuffling its characters leads to a significant drop performance of (5.8, 17.6, 35) pts in (BLEU/Ex, BLEURT/Ex, Ex-Match Accuracy%). Comparatively, the proper noun results in a much smaller drop of (3.1, 8, 16.3) pts in the three evaluation metrics. This shows that the common noun has a more prominent role to play in the generation of semantic interpretations, compared to the proper noun in the PNC.

5.2.6.4 Quality Assessment of Evaluation Metrics

For evaluating the quality of the metrics that are used for evaluating the model predictions, in particular, the semantic matching component (Section 5.2.5), we manually annotate the quality of model predictions with respect to gold using a 3-index scale. The scale indicates whether the quality of the prediction is bad, average or good. This is done only for the cases where the gold annotation indicates that the compound is compositional and the prediction of the model is also a paraphrase, as semantic matching is applicable only in these cases. A total of 1500 examples are annotated out of which 500 are used for finetuning the learned metrics such as BLEURT. On the remaining 1K examples, we compute the Pearson and Kendall correlation between the scores assigned by the evaluation metric and the human-annotated scores. We report the results in Table 5.14 for five evaluation metrics which include BLEU, and BLEURT with and without finetuning on both the base and large variants. We find that the fine-tuned BLEURT outperforms both BLEU and the un-trained BLEURT. It specifically outperforms BLEU by a significant margin from 0.28 to 0.57 in Pearson correlation and 0.23 to 0.46 in Kendall correlation. We find that the performance of both the base and large variants of BLEURT perform similarly after being finetuned and a minor difference exists in their untuned variants. Therefore, we use the base variant of BLEURT in the rest of the experiments.

We note that the correlation of 0.57 is on par with the current state of NLG metrics. For example, Chen et al. (2020), reports a correlation of 0.45-0.60 for standard metrics such as BLEU, BERTScore (Zhang* et al., 2020) on short-text evaluation. To further encourage research in building better generation metrics, we release the human judgements of the interpretations.

Metric	Pearson $\lvert\rho\rvert$	Kendall τ
BLEU	0.28	0.23
BLEURT-base	0.43	0.37
BLEURT-large	0.49	0.4
BLEURT-base *(tuned)*	0.56	**0.46**
BLEURT-large *(tuned)*	**0.57**	**0.46**

Table 5.14: Quality of metrics evaluated using Pearson and Kendall rank correlation. (tuned) indicates models that are fine-tuned on 500 manually evaluated comparisons.

Model	Knowledge	Ex-Match			Sem-Match		Sem/Ex-Match	
		Precision	Recall	Accuracy	BLEU	BLEURT	BLEU	BLEURT
MtGen	None	**78.2** ± 1.14	74.5 ± 1.48	82.8 ± 0.25	40.5 ± 0.58	63.8 ± 0.36	50.0 ± 0.39	62.9 ± 0.29
	Sentence	76.9 ± 1.70	78.6 ± 1.38	83.3 ± 0.69	40.4 ± 0.43	63.2 ± 0.30	51.1 ± 0.25	63.4 ± 0.50
	WordNet-NN	76.4 ± 1.30	80.5 ± 1.95	**83.5** ± 0.36	40.8 ± 0.63	63.3 ± 0.40	51.8 ± 0.55	63.8 ± 0.47
	Wiki-NNP	51.7 ± 1.04	94.7 ± 0.82	65.2 ± 1.41	25.9 ± 1.26	36.0 ± 3.80	42.9 ± 0.44	46.0 ± 1.14
	NER-NNP	75.4 ± 2.19	80.5 ± 3.06	82.9 ± 0.45	40.5 ± 0.79	63.4 ± 0.62	51.4 ± 0.29	63.5 ± 0.26
UniGen	None	71.7 ± 0.68	83.4 ± 1.07	81.6 ± 0.21	41.5 ± 0.16	63.7 ± 0.17	52.0 ± 0.24	63.2 ± 0.15
	Sentence	72.1 ± 0.32	83.6 ± 0.44	81.9 ± 0.19	41.3 ± 0.19	63.4 ± 0.45	52.0 ± 0.12	63.3 ± 0.17
	WordNet-NN	71.0 ± 1.71	**86.2** ± 1.23	81.7 ± 0.88	**42.0** ± 0.40	64.0 ± 0.39	**52.9** ± 0.34	**63.8** ± 0.42
	Wiki-NNP	68.6 ± 2.08	68.2 ± 1.93	76.5 ± 0.82	26.1 ± 0.78	39.0 ± 2.18	38.7 ± 0.42	45.3 ± 1.25
	NER-NNP	71.9 ± 0.98	81.8 ± 1.70	81.3 ± 0.25	41.6 ± 0.34	**64.2** ± 0.68	51.6 ± 0.42	63.1 ± 0.49

Table 5.15: Performance of the two models, MtGen and UniGen on the randomly split ProNCI dataset trained under five different knowledge settings.

Model	Ex-Match			Sem-Match		Sem/Ex-Match	
	Precision	Recall	Accuracy	BLEU	BLEURT	BLEU	BLEURT
Ponkiya et al. (2020)	0.0	0.0	62.8	22.9	44.1	14.4	27.7
Rand Few-Shot (5)	53.7	1.2	63.0	27.6	41.2	17.7	26.2
Rand Few-Shot (10)	37.7	33.6	54.4	28.8	42.2	24.7	29.7
KNN Few-Shot (5)	**70.5**	53.0	74.3	34.8	51.7	38.7	48.0
KNN Few-Shot (10)	68.4	**60.1**	**74.8**	**35.4**	**53.2**	**41.0**	**50.3**

Table 5.16: Performance of T5 model without any finetuning on the random split of ProNCI dataset.

5.2.6.5 Random Split of ProNCI

In this section, we evaluate the results of UniGen and MtGen on a random split of the ProNCI dataset, where the 22,500 examples are randomly split into 17,500 training, 2,500 validation and 2,500 testing examples. The results are reported in Table 5.15 and Table 5.16. We find that the performance is higher compared to when split according to common nouns. This can be attributed to the lack of intersecting common nouns between the training and evaluation sets that could have provided additional clues. This leads to a drop in (BLEU/Ex, BLEURT/Ex, Ex Acc%) scores of (5.7, 5.4, 3.3) pts in MtGen and (5.3, 4.6, 2.9) pts in UniGen.

Init	Ex-Match	Sem/Ex-Match	
	Accuracy	BLEU	BLEURT
Random	63.9 ± 1.98	33.9 ± 1.25	30.5 ± 1.27
T5-base	78.7 ± 1.40	46.7 ± 1.12	58.6 ± 0.94
T5-large	**79.4** ± 0.11	**47.7** ± 0.29	**58.7** ± 0.35

Table 5.17: Performance of the UniGen model on the ProNCI dataset trained using different initializations of the Seq2Seq model. Random initialization leads to a huge drop in performance.

Knowledge	Ex-Match	Sem/Ex-Match	
	Accuracy	BLEU	BLEURT
Sentence	79.3 ± 0.55	**47.9** ± 0.41	**59.5** ± 0.57
+WNet-NN	77.4 ± 2.14	46.5 ± 1.48	57.4 ± 1.63
+Wiki-NNP	74.0 ± 1.62	38.9 ± 4.21	46.1 ± 6.38
+NER-NNP	**79.4** ± 0.23	47.0 ± 0.52	58.9 ± 0.40

Table 5.18: Performance of the UniGen model on ProNCI dataset trained with additional sources of knowledge added over Sentence knowledge. The additional sources do not provide further benefits.

5.2.6.6 Effect of Pretraining

To understand the effect pretraining has on the effect of model performance for the task of semantic interpretation of proper noun compounds, we re-train the UniGen on the NOUN split starting from random initialization, instead of using T5-base, the default in all of our experiments. We also experiment with using T5-large. We report the results in Table 5.17. We find that Random initialization is considerably worse, where the scores reduces from 46.7 to 33.9 in BLEU/EM and 58.6 to 30.5 in BLEURT/EM. This indicates that pretrained initialization plays a significant role in the final performance on the task. Moreover, on experimenting with the larger model, T5-large, we find a slight increase in scores from (46.7, 58.6, 78.7) to (47.7, 58.7, 79.4) in (BLEU/EM, BLEURT/EM, CMP). Thus the task can benefit from the scaling of the language models as they typically gain more information about the common and proper nouns.

5.2.6.7 Adding multiple sources of knowledge

In Table 5.11 and Table 5.15, we observed statistically significant benefits to model performance after adding information about the noun compound from various sources of knowledge. We also experiment with adding information from multiple sources of knowledge to see if it can further augment the model performance. On taking the best-performing sentence knowledge in the UniGen model on NOUN split, we add the remaining three sources of knowledge and report their performance in Table 5.18. We find that it results in a slight decrease in performance in the case of WNet-NN and NER-NNP and in the case of Wiki-NNP the decrease is much greater because of the reduced quality of Wikipedia entities. We attribute this to possible confusion arising from disparate sources of knowledge that highlight different parts of the noun compound.

5.2.6.8 Error Analysis

We analyze the mistakes made by the UNIGEN model trained with Sentence Knowledge to find potential scopes for improvement. We divide them into the following categories -

1. Lack of word sense disambiguation: We notice mistakes in the model predictions in cases when some words have multiple meanings. The model defaults to choosing the one with the most frequent usage and not disambiguating properly based on the context. For example, the interpretation, "*Sunday strip* is a comic printed on a Sunday" is mistaken as "*Sunday strip* is a show on Sunday", even when the sentence contains sufficient clues for the same. The given sentence is "In a few cases, the topper introduced characters later developed into a successful Sunday strip."

2. Non-Informative predictions: Although predictions are not wrong they are often not very informative. For example, the model produces the following interpretation, "*EU economies* are based in EU" compared to the more detailed gold "*EU economies* are the financial condition of EU members".

3. Errors in evaluation and mistakes in Gold: In some cases, the evaluation metric is unable to capture semantic similarity. For example, the model prediction "Baltimore hospitals are located in Baltimore" and the gold, "Baltimore hospitals are medical institutions in Baltimore", has a BLEURT score of only -0.11.

5.2.7 Application to Open IE

To demonstrate the downstream value of the noun compound interpretations, we add them to a state-of-art Open IE system, Gen2OIE (Section 3.2), and generate new extractions that capture implicit relations. We apply this on a corpus of 21,228 COVID-19 news headlines that contain proper noun compounds like *COVID-19 outbreak*, *Rohingya refugee*, etc (Aslam et al., 2020).
Integration: To achieve this, we train a Seq2Seq model that takes as input the sentence concatenated with the interpretation of the PNC present in it and outputs an interpretation-augmented sentence. For example, the sentence, "Workers sound alarm on *Covid-19 outbreak*" and the interpretation, "*Covid-19 outbreak* is an outbreak of Covid-19" are integrated to get the following output, "Workers sound alarm on outbreak of Covid-19". Considering the simplicity of the task, we annotate a small set of 200 examples of this kind and use it to train a Seq2Seq model. Since this style of integration converts the implicit relation in the noun compound to an explicit form, it allows for the Open IE system to add new relations that were missing earlier.
Processing: We experiment with a high-precision rule that post-processes an extraction to generate a new one, whenever the extraction contains a PNC at the start of its object. For example, if the original extraction is (Workers; sound alarm on; COVID-19 outbreak), and the corresponding integrated extraction is (Workers; sound alarm on; outbreak of COVID-19), then the rule generates a new extraction by moving words till the proper noun back into the relation. In this case, we get the extraction, (Workers; sound alarm on outbreak of; COVID-19) – thus exposing a direct relationship between workers and COVID-19, which was not present earlier. The overall pipeline is shown in Figure 5.5.

We find that extractions generated using this pipeline lead to an increase in yield of 7.5% where the added extractions have a precision of 85%, compared to a precision of 82.2% of the original extractions, as determined on a random sample of 500 extractions. We note that the method can use any Open IE system without any additional finetuning to produce the noun-compound extractions.

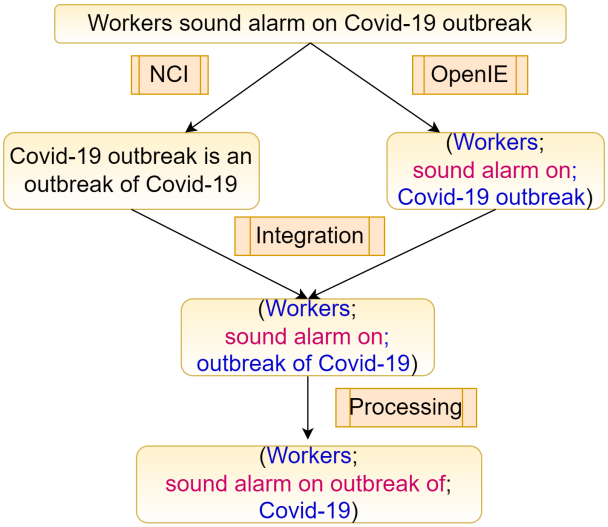

Figure 5.5: Open IE Pipeline. Postprocessing of the extraction integrated with noun compound interpretation generates the new extraction.

5.3 Open IE Systems: Open IE 6.2

Concluding the contributions discussed in Chapter 3, Chapter 4 and Chapter 5 – we have contributed three new Open IE architectures, handled two specific linguistic phenomena and used two rescoring mechanisms. We release a new Open IE system, OpenIE-6.2, that combines all these features and releases them as a software package for the community to use.[5]

In this subsection, we describe the overall flow of the proposed OpenIE-6.2 system, summarized in Figure 5.6. The input sentences are passed through a coordination analysis module that uses the IGL-CA model. The model detects the coordination structures present in the input and splits a possibly complex sentence into multiple simple sentences (Section 5.1). Each sentence is passed to one of the three available Open IE systems – IMoJIE, Gen2OIE or CIGL-OIE to generate extractions of each of the outputs of the simple sentence by the coordination analysis module. The simple extractions that are generated by the chosen Open IE system are merged to get the coordination-analyzed set of extractions for each of the original input sentences. The noun compound interpretation module takes the merged extractions and integrates the interpretations of any proper noun compounds that may be present in the input (Section 5.2). The interpreted extractions are then passed into either the labeling or generative rescoring model (Section 4.3) to give the final set of output extractions.

In summary, we build a new state-of-the-art model for Open IE in Chapter 3, Chapter 4 and Chapter 5. However, the proposed OpenIE-6.2 system is still limited to English and is not designed to handle other languages. To address this, in the next chapter, we focus on building Open IE systems in other languages apart from English.

[5]https://github.com/dair-iitd/openie6

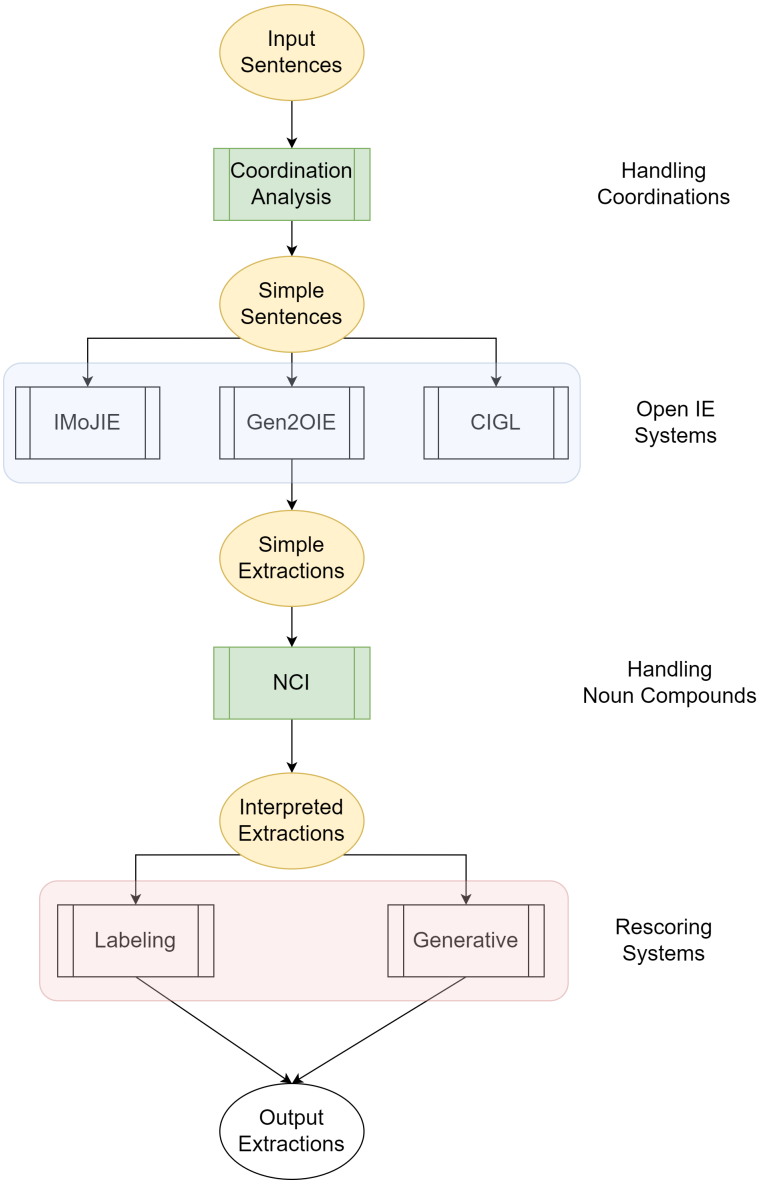

Figure 5.6: Flowchart of the OpenIE-6.2 system. It allows flexibility of choosing from three Open IE systems (IMoJIE, Gen2OIE, CIGL), adding two linguistic features (Coordination Structures, Noun Compounds) and rescoring using two models (Labeling, Generative)

Chapter 6

Interlingual Transfer of Open IE Training Data

Progress with supervised Open IE has been largely limited to English, owing to the scarcity of training data in other languages. In this chapter, we explore techniques to automatically convert English text for training Open IE systems to other languages. We introduce a model to translate English sentences and their corresponding extractions *consistently* with each other — with no changes to the vocabulary or the semantic meaning. We call this the Alignment-Augmented Consistent Translation (AACTRANS) model. Using the data generated with AACTRANS, we train the generative Open IE model, Gen2OIE, introduced in Section 3.2. Gen2OIE increases relation coverage using the RC heuristic that is generalizable to multiple languages, in contrast to CIGL which uses an English-specific training loss. Evaluations on 5 languages — Spanish, Portuguese, Chinese, Hindi and Telugu — show that the Gen2OIE with AACTRANS data outperforms prior systems by a margin of 6-25% F1.[1]

6.1 Alignment Augmented Consistent Translation

Both neural and non-neural types of Open IE systems have been limited to only a few languages – earlier non-neural systems required language-specific Open IE insights, and current neural systems require annotated training corpus that pose a barrier, particularly for low-resource languages.

Related tasks such as Semantic Role Labeling face similar challenges in extending to multiple languages. X-SRL (Daza and Frank, 2020) addresses this by automatic translation of English sentences to the target language followed by projecting the labels from the source sentence to the target sentence. This allows us to infer the semantic role labels in the translated sentence. However, translating the sentence alone may be insufficient for Open IE because the generated tuples can include additional words that are absent in the sentence, or require some changes to the word morphology used in the sentence. Although less prevalent in English, these characteristics need to be addressed in many other languages.

X-SRL approach may be extended such that each extraction can also be automatically translated and subject, relation, and object labels projected from English extractions. However, *independent* translation of sentence and extraction may introduce unwanted *lexical* (e.g. synonyms) or *semantic* (e.g., change in gender) variations between the translations, as shown in Table 6.1. Such translation inconsistencies in the training data lead to suboptimal Open IE examples.

[1]Code and models are released at https://github.com:dair-iitd/moie

Lexical Inconsistency	
English Sentence	*The shield of Athena Parthenos, sculpted by Phideas, depicts a **fallen** Amazon*
English Extraction	*<s> The shield of Athena Parthenos </s> <r> depicts </r> <o> a **fallen** Amazon </o>*
Spanish Sentence	El escudo de Atena Parthenos, sculptado por Phideas, representa un Amazonas **fallecido**
Spanish Ext (Indp)	<s> El escudo de Atena Parthenos </s> <r> representa </r> <o> un Amazonas **caído** </o>
Spanish Ext (Const)	<s> El escudo de $\cdots$ <r> representa </r> <o> un Amazonas **fallecido** </o>
Semantic Inconsistency	
English Sentence	*The discovery was remarkable as the skeleton was almost identical to **a modern Kuvasz***
English Extraction	*<s> skeleton </s> <r> was </r> <o> almost identical to **a modern Kuvasz** </o>*
Spanish Sentence	Un descubrimiento notable porque fósil era casi idéntica a **un Kuvasz moderno**
Spanish Ext (Indp)	<s> skeleto </s> <r> era </r> <o> casi idéntica a **una Kuvasz moderna** </o>
Spanish Ext (Const)	<s> fósil </s> <r> era </r> <o> casi idéntica a **un Kuvasz moderno** </o>

Table 6.1: Open IE examples transferred from English to Spanish, using both Independent (Indp) and Consistent (Const) translations. Independent translation results in inconsistencies which may have the same meaning (by using synonyms, fallecido vs. caído) or may change the meaning (changing gender from male to female, moderno to moderna). Consistent translation avoids these issues, resulting in better-quality training data.

To maintain consistency between translations of a sentence and its extractions, both translations must use the same words and their morphological variants as much as possible. Hence, we propose Alignment-Augmented Consistent Translation (AACTrans), a seq2seq model that translates the given input text in a way that is consistent with a *reference translation* by biasing the translation to use words similar to those available in the reference. To ensure that translations of sentences and extractions are consistent with each other, we use the AACTrans model to translate each of them with the same reference. In Section 6.2.1, we describe the reference used in training and inference.

As shown in Chapter 3 and Chapter 4, both generation-based and labeling-based architectures have shown competitive performance on English Open IE. However, labeling-based models cannot naturally introduce new words or change the morphology of sentence words required in some languages. We also use the training heuristic specific to two-stage models that increase relation coverage across multiple languages.

Our major contributions in this chapter are that:

1. we introduce a novel technique for transferring data from English to other languages using the AACTrans model and label projection,

2. we release Open IE evaluation datasets for two Indian languages, Hindi and Telugu, and

3. the Gen2OIE trained with AACTrans outperforms prior systems by 6-25% in F1 across five languages – Hindi, Telugu, Spanish, Portuguese and Chinese.

6.2 AACTrans: Crosslingual Data Transfer

In this section, we introduce a new translation methodology, which is developed for converting Open IE training data from source language[2] L_{src} to a target language L_{tgt}. More formally, the

[2]In the current work, we always use English as the source.

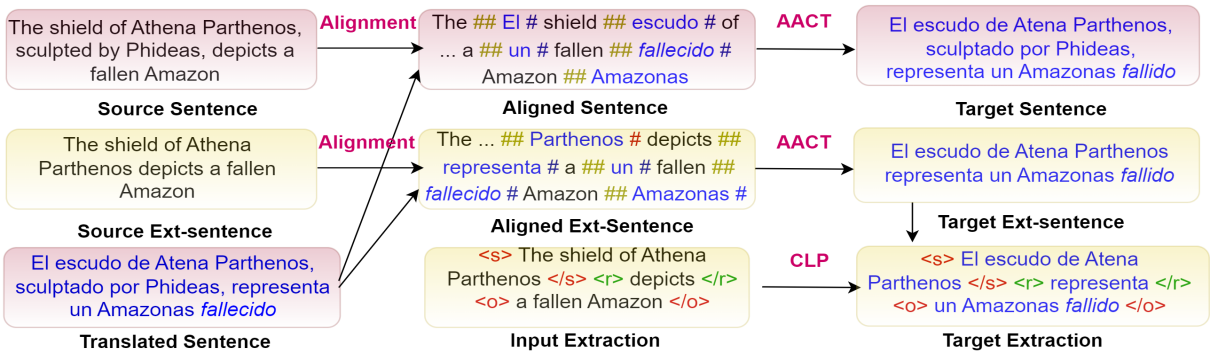

Figure 6.1: Crosslingual Data Transfer pipeline from English to Spanish. Firstly, The sentence and ext-sentences in English are aligned with a translation of the sentence (**Source Sentence** + **Translated Sentence** → **Aligned Sentence** and **Source Ext-sentence** + **Translated Sentence** → **Aligned Ext-sentence**). Secondly, the AACTRANS model uses the aligned text to generate the final consistent translations (**Aligned Sentence** → **Target Sentence** and **Aligned Ext-Sentence** → **Target Ext-Sentence**). Finally, Cross Lingual Projection (CLP) introduces **S**, **R**, **O** tags in the extraction (**Target Ext-Sentence** + **Input Extraction** → **Target Extraction**).

source sentence, $S^{L_{src}}$, and all of its extractions, $E^{L_{src}}$, are translated to language L_{tgt}, to get the translated sentence $S^{L_{tgt}}$ and its corresponding extractions $E_i^{L_{tgt}}$.

The data transfer is achieved with the help of two components, a novel type of translation model called AACTrans that ensures *consistency* in translations and a label projection model called CLP that projects word-level labels from one language to another. The overall flow is described in Algorithm 4 which includes 1) the training of the AACTrans model (Section 6.2.1), 2) applying the AACTrans model on Open IE examples in the source language (Section 6.2.2), 3) using the CLP algorithm to get extractions of the target language (Section 6.2.3).

Algorithm 4 Crosslingual Data Transfer

Step 1: AACTRANS model is trained for translation between L_{src} and L_{tgt} languages.
Step 2: The trained AACTRANS model is used to translate the Open IE examples in L_{src}.
Step 3: The CLP label projection is used to obtain the final labelled extractions in L_{tgt}.

Figure 6.1 further illustrates the application of the AACTrans+CLP pipeline to Open IE with the help of an example.

6.2.1 Consistent Translation

The pursuit of high-quality, accurate machine translation is a prominent challenge in the field of natural language processing. However, most existing models tend to focus on translation accuracy at the sentence level, sometimes leading to inconsistent translations across multiple related texts. The Alignment-Augmented Consistent Translation (AACTRANS) model seeks to address this gap by ensuring consistent translations of similar phrases across different sentences. This is needed to ensure that the extraction is translated similarly to the original sentence.

We propose a novel Seq2Seq-based translation model, denoted as Alignment-Augmented Consistent Translation (AACTRANS), that ensures consistent translations from the source language, L_{src}, to the target language, L_{tgt}. A translation is deemed consistent if similar phrases

across multiple texts maintain the same grammatical structure and vocabulary, allowing for minimal adjustments necessary to maintain fluency.

To understand how consistency among multiple translated texts applies to Open IE, we can look at it from the angle that there is a 'parent text' (the original sentence) and a 'child text' (the linearized version of the extraction, removing the field delimiters). Let P_{src} and P_{tgt} represent the parent text in the source and target languages, respectively, and C_{src} and C_{tgt} represent the child text in the source and target languages respectively. A translation $T : L_{src} \rightarrow L_{tgt}$ is deemed consistent if for all similar phrases p present in multiple texts $x \in P_{src}, C_{src}$, we have:

$$T(p_{x_{src}}) = p_{x_{tgt}} \tag{6.1}$$

for minimal adjustments necessary to maintain fluency, where $p_{x_{src}}$ is the phrase in the source language text x_{src} and $p_{x_{tgt}}$ is the translated phrase in the target language text x_{tgt}.

To aid this process of consistent translation, we employ a reference text in the target language, R_{tgt}, that guides all translations. By individually maintaining consistency with the reference text, the translations of both parent and child texts are intrinsically consistent with each other. For instance, the phrase 'a fallen Amazon' might appear in both the parent and child texts, but we desire a consistent translation of this phrase in both contexts. Therefore, a reference translation like 'un Amazonas fallecido' is provided, biasing the translation system to opt for 'fallecido' (or its appropriate morphological variants) as the translation of 'fallen' over synonymous words such as 'caído'.

Let A_{s_i} denote the set of aligned words in R_{tgt} for each word s_i in P_{src} or C_{src}, as determined by a word alignment model, which identifies pairs of semantically equivalent words in two parallel sentences in different languages (Dou and Neubig, 2021). Then an aligned text $S'_{x_{src}}$ for each x_{src} in P_{src}, C_{src} is constructed by concatenating each of the words s_i with their aligned words A_{s_i}, using ## as a separator. Using our example, we detect 'Amazon' is aligned with 'Amazonas' and 'fallen' with 'fallecido'. So 'fallen Amazon' will appear as 'fallen ## fallecido # Amazon ## Amazonas' in S'.

Next, we train the AACTRANS model using parallel sentences from languages L_{src} and L_{tgt} available in existing translation corpora. For each parallel sentence pair, **s** and **t**, we utilize **t** as the reference **r**. We then form the input **s**′ using the alignments between the words of **s** and **t**. The AACTRANS model is trained with **s**′ as the input and **t** as the output using the Seq2Seq architecture. Since **s**′ has words from **t**, the model learns to use them during training and applies the same at inference time. As a result of this process, the AACTRANS model learns to generate translations of the child text that match the source language's semantic content and maintain grammatical consistency with the parent text in the target language.

6.2.2 Consistent Translation for Crosslingual Data Transfer

To aid in the translation of Open IE extractions, we create a sub-sentence from each extraction by concatenating the phrases in all the fields of the extraction. The order of concatenation is such that the constructed sub-sentence is grammatically valid. We refer to this sub-sentence as an **ext-sentence** and represent it as es^L, where the superscript *L* represents the language. For most English extractions, the ext-sentence corresponds to concatenating the fields in the order of subject, relation and object. However, other languages may follow a different order or allow for multiple orders. We rely on the output of the system that translates the English ext-sentence to determine the ext-sentence in other languages. Moreover, each extraction can be seen as a labeling over the words of ext-sentence with either the **S**ubject, **R**elation or **O**bject tags. Tags for each word in the ext-sentence can also be regarded as the extraction.

We need to consistently translate English sentence $S^{L_{src}}$ and each of its ext-sentences $es_i^{L_{src}}$. We use an off-the-shelf translation system (*ofst*) to translate $S^{L_{src}}$ to language L_{tgt}, represented as *ofst*-$S^{L_{tgt}}$. For example, the sentence could be 'The shield of Athena Parthenos sculpted by Phideas, depicts a fallen Amazon' and the ext-sentence could be 'The shield of Athena Parthenos depicts a fallen Amazon', with the reference translation as 'El escudo de Athena Parthenos, sculptado por Phideas, representa un Amazonas fallecido'. This *ofst*-$S^{L_{tgt}}$ is used as the common reference **r** for constructing aligned sentence al-S and aligned ext-sentence al-es_i from sentence $S^{L_{src}}$ and ext-sentence $es_i^{L_{src}}$, respectively. Due to using alignments, al-S and al-es_i contain words from both the languages, L_{src} and L_{tgt}.

In the above example, the aligned sentence would be 'The ## El # shield ## escudo # of ## de # Athena ## Athena # Parthenos ## Parthenos # sculpted ## sculptado # by ## por # Phideas, ## Phideas, # depicts ## representa # a ## un # fallen ## fallecido # Amazon ## Amazonas' and the aligned ext-sentence would be 'The ## El # shield ## escudo # of ## de # Athena ## Athena # Parthenos ## Parthenos # depicts ## representa # a ## un # fallen ## fallecido # Amazon ## Amazonas'. We then apply the trained AACTRANS model on al-S and al-es_i to generate target sentence $aact$-$S^{L_{tgt}}$ and target ext-sentence $aact$-$es_i^{L_{tgt}}$ respectively. This leads to generation of the target sentence, 'El escudo de Atena Parthenos, sculptado por Phideas, representa un Amazonas fallido' and the target ext-sentence, 'El escudo de Atena Parthenos representa un Amazonas fallido' that corresponds to the triple (El escudo de Atena Parthenos; representa; un Amazonas fallido). To get the triple from the ext-sentence, we rely on Crosslingual Label Projection which is explained next.

6.2.3 Crosslingual Label Projection (CLP)

Each word in the target ext-sentence, $aact$-$es_i^{L_{tgt}}$, must be labeled with either the S, R, or O tag to form the completed extraction in language L_{tgt}. The tags from the corresponding $E_i^{L_{src}}$ are projected onto $aact$-$es_i^{L_{tgt}}$ using the Crosslingual Projection algorithm (described in Section 2.4.5.1), which uses word alignments between $es_i^{L_{src}}$ and $aact$-$es_i^{L_{tgt}}$ and produces as output, the tags over $aact$-$es_i^{L_{tgt}}$, giving extraction $aact$-$E_i^{L_{tgt}}$. For example, the labels from input extraction, (The shield of Athena Parthenos; depicts; a fallen Amazon) are projected onto the target ext-sentence 'El escudo de Atena Parthenos representa un Amazonas fallido' to give the extraction, (El escudo de Atena Parthenos; representa; un Amazonas fallido). The final set of <sentence, extractions> pairs constitute the data for training the Open IE system in language L_{tgt}.

6.3 Experimental Setting

We train Open IE systems in 5 languages, Spanish (ES), Portuguese (PT), Chinese (ZH), Hindi (HI) and Telugu (TE), by using the training data transferred from English to the respective language. For training the Seq2Seq models used in the data generation pipeline and the Open IE systems based on the Gen2OIE architecture, we choose either the mBART (Liu et al., 2020b) or mT5 (Xue et al., 2020) model depending on the particular language. Both of them are pre-trained multilingual Seq2Seq models that are trained with a span denoising objective on a large corpus of text containing many languages. mBART is pre-trained on CC25 and mT5 is pre-trained on mC4 corpus which contains text in 25 and 101 languages, respectively. Since mBART does not support Portuguese and Telugu, we use mT5 for these two languages and mBART for the remaining 3 languages. We use the default hyperparameters recommended for these models.

Training Datasets for AACTrans: We make use of parallel (English, language L_t) sentences available in standard translation corpora using the method described in Section 6.2. For Spanish, we use parallel sentences from EuroParl corpus (Koehn et al., 2005), and for Portuguese, we use a subset of the ParaCrawl corpus (Bañón et al., 2019), as chosen by Lopes et al. (2020). For Hindi, we use the IIT-B corpus (Kunchukuttan et al., 2018), and for Telugu, we use the Samanantar corpus (Ramesh et al., 2021). For Chinese, we use the data released for WMT19 (Barrault et al., 2019). We list the BLEU scores of the various systems in Section 6.4.4.
Training Dataset for Open IE: We use the same OIE4 training corpus from Chapters 3 and 4 and transfer it to the other languages.
Evaluation Datasets and Metrics: For evaluating translation systems we use the test sets available in the respective corpora and use SacreBLEU (Post, 2018) as the metric.[3] As in Chapter 3 and Chapter 4, for evaluating different Open IE systems we use the Optimal F1 and Area Under Curve (AUC) computed by the CaRB (Bhardwaj et al., 2019) scoring function. For Spanish, and Portuguese Open IE we use test sets provided in Ro et al. (2020). For Chinese Open IE, we randomly choose 10% of the SAOKE dataset (Sun et al., 2018b).

To evaluate our method on medium and low-resource languages, we release new Open IE test sets in Hindi and Telugu. Human annotators who are fluent in both languages and are knowledgeable about the Open IE task translated about 300 randomly chosen sentences and their corresponding extractions from the CaRB test set. They were paid $2.5 per sentence.[4]

Table 6.2 lists the number of examples in different languages used for training and evaluating translation and Open IE systems.

	EN	ES	PT	ZH	HI	TE
Translation						
Train	-	1.9M	5M	1M	1.6M	4.8M
Test	-	38473	99,087	2001	2507	2390
Open IE						
Train	91K	91K	91K	91K	91K	91K
Test	641	594	594	3833	298	302

Table 6.2: Data statistics for Open IE examples and (English, language *F*) parallel sentences.

6.4 Experiments

We perform experiments to answer the questions:

1. What is the quality of data generated with the AACTrans+CLP pipeline, assessed both by the final performance of systems trained using it and with metrics defined for evaluating consistency?

2. What are the incremental contributions of different components in the performance of Gen2OIE with AACTrans+CLP data?

[3] BLEU+case.mixed+numrefs.1+smooth.none+tok.intl+version.1.5.1

[4] Shubham Mittal helped with parts of the evaluation dataset collection and ablation analysis, which has been included as part of his BTech thesis.

Model	Training Data	ES		PT		ZH		HI		TE	
		F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
(Faruqui, 2015)	English	45.5	28.6	48.5	31.5	13.7	3.3	30.4	12.5	36.7	16.2
Multi^{2}OIE	English	60.0	41.5	60.2	41.1	23.7	8.1	28.8	10.9	16.5	4.1
Multi^{2}OIE	SentTrans+CLP	62.0	42.8	60.9	41.3	21.2	6.5	48.1	27.6	33.4	15.4
OpenIE6	SentTrans+CLP	56.8	37.4	58.7	39.4	18.2	4.8	46.3	28.0	39.0	**18.3**
IMoJIE	AACTrans+CLP	61.6	43.1	59.7	39.9	15.4	4.0	47.5	26.3	33.9	15.5
GenOIE	SentTrans+CLP	60.4	40.6	63.5	43.7	20.9	4.9	51.5	28.5	**41.7**	16.3
	SentExtTrans+CLP	58.3	39.7	57.3	36.5	20.8	5.6	51.6	28.1	36.6	13.9
	AACTrans+CLP	60.8	41.3	63.9	44.8	23.1	5.9	51.6	28.6	39.3	15.1
Gen2OIE	SentTrans+CLP	64.2	44.6	65.6	**50.0**	29.0	8.9	52.3	30.8	40.3	15.6
	SentExtTrans+CLP	64.7	46.1	63.7	45.5	29.3	10.2	52.5	31.0	39.8	15.6
	AACTrans+CLP	65.9	47.2	**66.4**	49.2	29.8	10.3	52.8	**32.0**	41.5	16.6
Gen2OIE-mT5	AACTrans+CLP	**67.9**	**48.5**	**66.4**	49.2	**33.3**	**12.7**	**53.6**	30.9	41.5	16.6

Table 6.3: F1 and AUC performance of Open IE systems in Spanish (ES), Portuguese (PT), Chinese (ZH), Hindi (HI) and Telugu (TE). Training with AACTrans+CLP data shows strong performance with both GenOIE and Gen2OIE models. We also report the results of training Gen2OIE model with mT5 on all languages.

6.4.1 Quality of AACTrans+CLP data

To test the quality of the Open IE examples generated using the AACTrans+CLP pipeline, we train both the GenOIE and Gen2OIE models over the data generated for different languages. In Table 6.3, we compare it with examples generated from two other methods, SentTrans and SentExtTrans.

SentTrans+CLP represents an adaptation of X-SRL (Daza and Frank, 2020) for Open IE where only the sentence is translated, and each extraction, which is expressed as labeling over the words in the sentence, is projected onto the translated sentence using the CLP algorithm described in Section 6.2.3. The projected extraction is now a labeling over the translated sentence; hence, it uses the same morphology as the sentence and cannot add new words. SentExtTrans+CLP uses an independent translation of English sentences and ext-sentences followed by CLP algorithm between the English and translated ext-sentences to transfer the labels. Although this allows for adding new words and changing morphology, it can result in a lack of consistency between the translations.

We find that both GenOIE and Gen2OIE show consistent gains with AACTrans+CLP data across various languages when compared with SentExtTrans+CLP and SentTrans+CLP data.

We experiment with two versions of Multi^{2}OIE (Section 2.4.5.5): 1) trained only on English Open IE data and applied to other languages in a zero-shot manner and 2) using language-specific training data generated from SentTrans+CLP. We specifically choose SentTrans+CLP data as all the extractions can be expressed as labels over the sentence, which is a requirement for training Multi^{2}OIE which is itself a labeling model. We find that Multi^{2}OIE model trained with SentTrans+CLP data improves over the zero-shot setting in all languages other than Chinese (discussed below). However, it performs significantly worse than Gen2OIE by (5.2, 3.3)% in (F1, AUC) on average, even on training with the same SentTrans+CLP data. This can be attributed to Multi^{2}OIE's lack of capability to handle: 1) overlapping relations, 2) multiple extractions per relation, 3) adding auxiliary words or 4) changing inflectional forms.

Model (Data)	ES		ZH		HI	
	F1	AUC	F1	AUC	F1	AUC
Gen2OIE (AACTRANS+CLP)	**65.9**	**47.2**	**29.8**	**10.3**	**52.8**	32.0
Gen2OIE (AACTRANS w/o Sentence Consistency+CLP)	64.0	44.3	29.6	**10.3**	51.9	30.8
Gen2OIE w/o Relation Ordering (AACTRANS+CLP)	65.2	45.6	29.6	9.8	52.5	31.8
Gen2OIE w/o Relation Coverage (AACTRANS+CLP)	60.6	40.3	23.9	6.6	**52.8**	**32.3**

Table 6.4: Ablations of Gen2OIE model trained with AACTRANS+CLP data on ES, ZH and HI. We analyze the effect of removing three components and re-training the model: 1. Sentence Consistency used in AACTRANS data generation, and 2. Relation Ordering is used, and 3. Relation Coverage used in Stage-1 model training.

	ES	PT	ZH	HI	TE
SenExtTrans+CLP	12.2	9.5	24.5	13.3	19.6
AACTrans+CLP	5.4	3.9	5.7	6.9	10.3

Table 6.5: Evaluating inconsistency between translated extractions and corresponding sentences.

We train IMoJIE and OpenIE6 (initialized with mBERT) on AACTRANS+CLP and SentTrans+CLP data. We find that they underperform Gen2OIE and Multi^{2}OIE. Compared to the two-stage models, both IMoJIE and OpenIE6 generate all the extractions autoregressively, which makes them more susceptible to noise in the automatically generated training data.

We additionally compare with Faruqui (2015), where the test sentence is translated into English, extractions are generated using OpenIE6 and they are projected back onto the test sentence. We find that the system results in poor performance due to a lack of language-specific training.

We observe that all systems have low performance in Chinese. We attribute this to the various artifacts present in the SAOKE test set, which include special relations such *DESC*, *TIME*, *ISA*, etc. Since these extractions cannot be generated in our pipeline, we observe the performance of only 33.2% F1 and 15.8% AUC with our best model, when compared to training Gen2OIE with SAOKE training data, which gives 52.5% F1 and 32% AUC.

We additionally train the Gen2OIE model using mT5 on AACTRANS data for all five languages (Gen2OIE-mT5 in Table 6.3) and find improvements of (2.1%, 3.5%, 0.8%) F1 over the mBART models used for ES, ZH and HI.

6.4.2 Evaluating Consistency

In order to measure the inconsistency of the generated extractions with respect to the sentence, we compute the fraction of words that occur in the extraction but are absent in the sentence. In Table 6.5, we find that across languages, the fraction is lower for training examples generated through the consistent translation methodology (AACTRANS+CLP) when compared against independent translations (SentExtTrans+CLP). This indicates that AACTRANS+CLP indeed achieve better consistency.

To analyze the reasons for the improvement in CaRB performance, we compute the fraction of words that are present in model predictions but absent in the gold extractions of the test set (denoted by AG - Absent in Gold). In Table 6.6, we see that Gen2OIE trained on AAC-

Data	ES		PT		ZH		HI		TE	
	AG↓	F1↑	AG↓	F1↑	AG↓	F1↑	AG↓	F1↑	AG↓	F1↑
SentExtTrans+CLP	2.74	64.7	3.51	63.7	10.55	29.3	1.78	52.5	2.36	39.8
AACTRANS+CLP	2.31	65.9	2.22	66.4	9.67	29.8	1.6	52.8	2.09	41.5

Table 6.6: Evaluating CaRB F1 and AG of Gen2OIE predictions trained on SentExtTrans+CLP and AACTrans+CLP data. We find a decreasing trend of AG with increasing F1.

TRANS+CLP achieves lower values than the same model trained on SentExtTrans+CLP data and this correlates with the increased CaRB performance. This shows that the model generates words closer to gold extractions (and hence closer to the input sentence), which contributes to higher performance.

6.4.3 Ablation Study

We choose three representative languages to conduct the ablation study — Spanish, Chinese, and Hindi. Portuguese and Telugu belong to the same language family as Spanish and Hindi, respectively. In Table 6.4, we show the results of individually removing components from the Gen2OIE trained on AACTRANS+CLP data.

In AACTRANS w/o Sentence Consistency, we use regular translation of sentences while using the consistent translation of extraction. This leads to a drop of (1.9, 0.2, 0.9)% in F1 for the three languages, and shows the importance of using consistent translation on both the sentence and extraction.

In Gen2OIE w/o Relation Ordering, we train Stage-1 Gen2OIE with randomly shuffled relations. This reduces the performance as our model uses auto-regressive training which benefits from following a fixed order, which we choose as the order of occurrence of the relations in the sentence.

In Gen2OIE w/o Relation Coverage, we find that performance decreases in Spanish and Chinese by 5.3% and 5.9% in F1, respectively, but remains the same in Hindi.

6.4.4 BLEU scores

Table 6.7 contains the BLEU scores of both the normal as well as consistent translations. We find that the performance remains nearly the same, indicating that the improved Open IE performance stems from the consistency in the translations.

BLEU	ES	PT	ZH	HI	TE
Translation	45.2	48.4	26.8	20.5	7.0
AACTranslation	43.7	47.8	28.2	20.1	7.5

Table 6.7: BLEU scores of translation and AAC-translation are similar showing that the performance improvement is because of the added consistency.

6.4.5 Effect of word alignments quality

To understand the effect of alignment quality, we replace the language-specific trained aligners (TA), with a standard pre-trained mBERT model (MA). We first note that in Table 6.8 that MA has a much higher alignment perplexity (used as a measure of unsupervised alignment quality in (Dou and Neubig, 2021)). We now experiment to replace TA with MA in our methodology. Aligners are used at two places in our setup - 1. Alignment-Constrained Translation and 2. Crosslingual Label Projection. We replace each of them with an mBERT aligner (MA) and show the results in Table 6.9. We find that there is some performance drop by using MA, but it is quite less compared to the drop in alignment perplexity. This suggests that our model is relatively robust to the quality of alignment.

Language	MA	TA
ES	0.38	0.19
HI	0.49	0.20

Table 6.8: Unsupervised alignment perplexity for mBERT (MA) and Trained (TA) aligners

(AACTRANS,CLP)	HI		ES	
	F1	AUC	F1	AUC
(TA, TA)	62.1	38.8	65.9	47.2
(TA, MA)	58.7	34.4	64.7	46.2
(MA, TA)	59.4	37.9	65.6	46.7

Table 6.9: F1 and AUC of Gen2OIE trained with examples generated using TA and MA alignment strategies. (1, 2) corresponds to aligner 1 being used in AACTRANS and aligner 2 being used in CLP.

6.5 Conclusion

In this chapter, we develop a novel AACTRANS+CLP pipeline for consistently transferring English Open IE examples to other languages. We show improvements over the existing baseline of Multi^{2}OIE, with an average improvement of 7.2% in F1 and 16.1% in AUC. It is tested in five languages, the largest number of languages covered by a single Open IE technique known to us. To encourage research in medium and low-resource languages, we additionally release new Open IE evaluation examples in Hindi and Telugu.

Chapter 7

Application of Open IE to Knowledge Bases

So far, we have focused on building Open IE systems that can generate a better quality of extractions, focusing on computational efficiency, handling special linguistic structures such as coordination, and noun compounds, and extending to other languages. In this chapter, we explore applications of the generated Open IE extractions – particularly, in relation to a structured source of knowledge, such as Knowledge Bases (KBs).

KBs are a large and useful source of information about the world and are composed of curated facts regarding entities. Each fact asserts a relation that exists between two entities, which can be expressed as *(subject; relation; object)*. KBs have proven to be helpful for NLP tasks like Question Answering (Saxena et al., 2020) and are used in a variety of industry applications (Dong, 2017; Fensel et al., 2020). KBs can be classified into two types: *canonical KBs*, like WikiData[1], DBPedia (Auer et al., 2007) and *Open KBs*, like OLPBench (Broscheit et al., 2020), ReVerb Open KB (Vashishth et al., 2018). Creation of KBs such as WikiData requires a lot of manual supervision, with KB facts comprising entities and relations that come from a pre-defined set with distinct unambiguous IDs. On the other hand, Open KBs are automatically constructed, making use of Open IE tuples themselves as facts, where all the fields contain unrestricted text. This allows Open KBs to achieve wider coverage of knowledge, at the cost of additional noise that may be introduced due to the reliance on existing Open IE systems.

In this chapter, we explore applications that deal with both kinds of KBs. First, in Section 7.1, we look at how natural language text or Open IE tuples can be linked to facts from existing canonical KBs across languages when the source text and target canonical tuple may be in different languages. Second, in Section 7.2, we tackle the problem of inferring new facts from existing Open KBs constructed from Open IE tuples.

7.1 Knowledge Base Fact Linking

In general, external sources of knowledge are helpful for NLP tasks such as question answering and fact verification. For example, KILT (Petroni et al., 2020) uses a retrieval+seq2seq model with Wikipedia documents as the knowledge source to solve many knowledge-intensive tasks and FAE (Verga et al., 2021) uses WikiData KG to inject knowledge into pre-trained language models in a modular fashion. Such knowledge-intensive NLP tasks can further benefit from linking natural language text to tuples from a canonical KB. While linking KB facts to text

[1] https://www.wikidata.org/

has received some attention in the literature (Elsahar et al., 2018), most of that work has been restricted to English facts and text. There is a growing need to connect facts and their mentions in text, especially when the language of the fact and text are different. For example, we would like to link a fact in English with a text in Hindi or Telugu. Therefore, we define the task of *multilingual fact linking* (MFL) as the task of predicting all the KB facts that are implied by a sentence, even when the sentence and KB fact are expressed in different languages.

KBs such as Wikidata often contain entity and property surface forms in multiple languages, although with severe skew (Kaffee et al., 2017). For example, *Argovia* and *Argau* are the English and Spanish *entity surface forms* for entity Q11972 in Wikidata. Similarly, *country* and *país* are the English and Spanish *property surface forms* for property P17. Combining these labels, we define the notion of *fact surface forms*, which are language-specific textual representations of an otherwise language-agnostic fact. For example, *(Aargau; country; Switzerland)* and *(Argovia; país; Suiza)* are the English and Spanish fact surface forms for the canonical fact (Q11972; P17; Q39) in Wikidata.

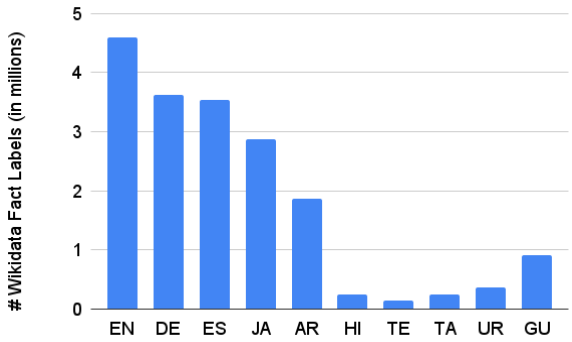

Figure 7.1: Distribution of languages of fact surface forms (in millions) on a subset of Wikidata. Compared to English and a few other languages, fact surface forms in Indian languages (the last five: HI, TE, TA, UR, GU) are extremely sparsely represented.

The problem of multilingual fact linking is made more difficult due to language skew among the fact surface forms in a KB. In Figure 7.1, we present the language distribution of fact surface forms (across ten languages) obtained from a subset of Wikidata containing popular entities. We find that the fact surface forms are heavily skewed towards higher-resource languages (first five), compared to the remaining five Indian languages, viz., Hindi (HI), Telugu (TE), Tamil (TA), Urdu (UR), and Gujarati (GU). These languages are spoken by hundreds of millions of speakers, although they are severely understudied in the NLP community. We focus on the problem of MFL, particularly Indian languages.

Multilingual fact linking involves selecting a small subset of facts relevant to the input sentence from the complete list of all KB facts, which may run into the scale of millions. To handle this, we make use of the dual encoder-cross encoder paradigm, where a dual encoder (Reimers and Gurevych, 2019) is used to retrieve the potential top-k facts quickly. These top-k facts can be re-ranked using classification-based cross-encoder architectures. However, due to the pipelined nature of the system, the performance of the re-ranking model is limited by that of the retrieval model. Therefore, we propose a novel model, **Re**trieval based **F**act-**Co**nstrained **G**eneration (**ReFCoG**), which replaces *re-ranking* cross encoders with a Seq2Seq cross encoder that is constrained to *generate* facts from the KB. This allows the generation of facts even when they are absent in the re-ranked set of facts, but are present in the KB. We show that such generative models outperform classification-based re-ranking, achieving a 10.7 pts improvement in

Task	Input/Output
Entity Linking (Botha et al., 2020)	**Input**: Table Jura stretches across the Swiss cantons of {*Basel-Landschaft*} and Aargau. **Output**: Q12146; English Surface Form = *Landschaft*
Relation Classification (Ormándi et al., 2021)	**Input**: Table Jura stretches across the OBJ{*Swiss*} cantons of SUBJ{*Basel-Landschaft*} and Aargau. **Output**: P17; English Surface Form = *country*
Canonical Fact Extraction (Elsahar et al., 2018)	**Input**: {*Table Jura*} stretches across the {*Swiss*} cantons of Basel-Landschaft and Aargau. **Output**: (Q356545; P17; Q39) English Surface Form = (*Table Jura*; *country*; *Switzerland*)
Multilingual Fact Linking	**Input**: टेबल जुरा बेसल-लैंडशाफ्ट और आरगौ के स्विस कैंटन में फैला हुआ है। (*tebal jura baasel-laindshaaft aur aaragau ke svis kaintan mein phaila hua hai*) **Output**: F_{23} = (Q12146; P17; Q39) English Surface Form = (*Landschaft*; *country*; *Switzerland*) F_{52} = (Q11972; P17; Q39) English Surface Form = (*Aargau*; *country*; *Switzerland*)

Table 7.1: KB linking task examples. Multilingual fact linking involves discovering the subset of KB facts expressed in a sentence, even when fact labels are available in a different language, requiring cross-lingual inference (Hindi-English in the above example). Fact-linking systems only output facts already present in the KB. Canonical fact extraction aims to discover new canonical facts not present in the KB while using the entities and relations defined in the existing KB schema. In contrast, Open IE extracts open-ended facts that may or may not correspond to entities, relations, or facts defined in the KB. *Q* and *P* represent the entity and property identifiers in Wikidata. The fact identifiers (e.g., F_{23}) are assigned and are not part of Wikidata.

precision and a 15.2 pts improvement in recall.

To facilitate research on the problem of multilingual fact linking, we curate a new evaluation dataset, **INDICLINK**, containing parallel sentences in English and six Indian languages tagged with the corresponding Wikidata facts expressed in them. The English sentences and facts are from the relation extraction dataset, WebRED (Ormándi et al., 2021). The test examples are manually translated into different languages, and automatic translations of sentences are used for training. We use KB facts from Wikidata and explore different strategies to use their fact surface forms in English and other languages, wherever available.

In summary, the main contributions of this chapter are as follows:

1. We introduce the task of multilingual fact linking (MFL) to link KB facts with their mentions in text, especially when there is a mismatch between the languages of the fact surface form and text.
2. We present INDICLINK, an evaluation dataset for MFL in English and six widely used Indian languages that are rarely studied in the NLP community. To the best of our knowledge, this is the first dataset of its kind for these languages.
3. We propose REFCOG, a novel retrieval+constrained-generation model for the task of MFL. The proposed method significantly outperforms standard retrieval+re-ranking models.

7.1.1 Related Work

In contrast to Open Information Extraction that represents semi-structured information from the text in an ontology-independent manner, Closed/Ontological Information Extraction often involves extracting structured information from the text by linking it to different parts of a KB with a pre-defined ontology. Linking text to KBs has been traditionally explored in various settings, such as entity linking, relation classification and extraction, fact extraction, and fact linking. We list various KB-related tasks and their corresponding inputs and outputs in Table 7.1 and are briefly described below. The entity linking and relation classification tasks are already discussed in Section 2.6.1.

Fact Extraction: The task involves joint entity and relation extraction (Zhong and Chen, 2021; Sui et al., 2020) focusing on discovering new facts that are not present in the KB. Whereas fact linking deals with connecting existing KB facts with text. Therefore, fact-linking models use KB facts (which may be millions), whereas fact extraction systems do not.

Fact Linking: Existing fact linking/alignment systems such as T-REx (Elsahar et al., 2018) align English DBPedia abstracts with Wikidata triples and provide a corpus of 11 million high-quality alignments. They use ad-hoc pipelines of entity linking, coreference resolution, and string matching-based predicate linkers. Our experiments show that our end-to-end linker, REFCOG outperforms such pipeline systems. In another line of work, multilingual fact retrieval (Jiang et al., 2020) is used to judge the factual knowledge captured within LM parameters by predicting masked entities in facts. However, we are only concerned with retrieving the facts present in the input text.

7.1.2 Multilingual Fact Linking: Problem Overview

A knowledge graph (KG) contains a list of entities $\mathcal{E}$, relations $\mathcal{R}$ and fact-triples $\mathcal{T}$ (also referred to as facts), where the *i*th fact links two entities ($s_i, o_i \in \mathcal{E}$) with a relation ($r_i \in R$) and is defined as, $F_i = (s_i; r_i; o_i)$. Multilingual KGs like Wikidata also contain textual forms of entities and relations in multiple languages. Given the set L of languages in the KG, for each language l, we construct the fact surface forms by concatenating the textual form of the entities and relation, if available. The surface form of F_i in language $l \in L$, F_i^l, exists if the entity and relation surface forms are available in language l, i.e., if s_i^l, r_i^l and o_i^l exist in KG, then, F_i^l=(s_i^l;r_i^l;o_i^l).[2] For example, in Table 7.1, (Q12146; P17; Q39) and (Landschaft; country; Switzerland) represent the fact and its English surface form.

Let T_m be a text in language m. Then, MFL aims to discover the set of linked KG facts, $LinkedFacts(T_m) \subseteq \mathcal{T}$, that are explicitly expressed in the text. The output set can be formally defined as,

$$\text{LinkedFacts}(T_m) = \{F_i : F_i = (s_i; r_i; o_i) \ \wedge F_i \in F \ \wedge \ T_m \Rightarrow F_i\}, \tag{7.1}$$

where $T_m \Rightarrow F_i$ implies that the fact F_i is expressed by the text T_m. The fact surface forms available in different languages are used to predict this set.

In Section 7.1.3, we describe the INDICLINK dataset curated for the MFL task, and in Section 7.1.4, we discuss the baselines and our proposed methods for the task.

IndicLink	English (EN)	Hindi (HI)	Telugu (TE)	Tamil (TA)	Urdu (UR)	Gujarati (GU)	Assamese (AS)	Total
#Test Examples	1002	889	888	881	1001	881	887	6429
#KG Facts	4.6M	230K	145K	248K	361K	91K	257	4.6M

Table 7.2: The new IndicLink dataset (Section 7.1.3) contains examples in English and corresponding manually translated test examples in six Indian languages. KG fact surface forms are always available in English but are only sparsely available in other languages.

7.1.3 IndicLink: A New Dataset for Fact Linking in Indian Languages

For curating a fact-linking dataset, we need a collection of sentences and the KG facts expressed in them. We use a subset of Wikidata facts as the oracle KG. Specifically, we consider all facts that exist between the most popular 1 million Wikipedia entities. Existing entity-linking datasets provide only the mentioned entities, but we also need the relations between entities. So we repurpose a relation classification dataset for fact linking by collecting the entity pair that expresses the particular relation.

We use WebRED (Ormándi et al., 2021) as it is the largest relation classification dataset covering over 100 relations. Multiple relations associated with a sentence are kept as separate examples in WebRED. However, not all examples have associated Wikidata entities. In some cases, the relation may be expressed between entity mentions that are literals such as dates/numbers, or refer to entities that do not exist in Wikidata. Therefore, for each sentence in WebRED, we collect facts if the relation involves valid Wikidata entities. We associate a special Null fact for sentences that have no expressed relations.

WebRED contains sentences only in English. We extend it to multiple languages by translating the test sentences using professional translators. To ensure the high quality of translations, we use three layers of quality checks: initial automatic translation, review, and proofreading.

To encourage research in Indian languages, which have historically lacked knowledge-linked resources, we consider six Indian languages – Hindi, Telugu, Tamil, Urdu, Gujarati, and Assamese – for our multilingual fact linking dataset, IndicLink. Table 7.2 contains the number of test examples and KG facts considered. For 6,429 sentences we end up with 11,293 facts implying an average of 1.7 linked facts per sentence.

We explore automated techniques for getting training examples as it is expected that high-quality language-specific training data will be unavailable for the vast majority of languages. We translate 31K WebRED training sentences into the respective Indian languages using Google Translate. The one exception is Assamese which does not have a translation system available and hence does not have any training data.

7.1.4 ReFCoG: Proposed Method for MFL

The dual encoder - cross encoder architecture is commonly used for tasks such as semantic search (Reimers and Gurevych, 2019), and more recently for tasks like Entity Linking (Botha et al., 2020) that require classification over a large target space. It typically involves a pipeline of dual encoder-based retrieval models and a cross-encoder re-ranking model to get the most relevant targets for the given input text. The retrieval model returns the top-k targets from the entire set. Then the re-ranking model scores the top-k targets using a slower model that would have been intractable to apply on the complete set. Apart from re-ranking cross encoders, we

[2]For the right to left languages like Urdu, we use $(s_i^l;r_i^l;o_i^l)$ as the fact description.

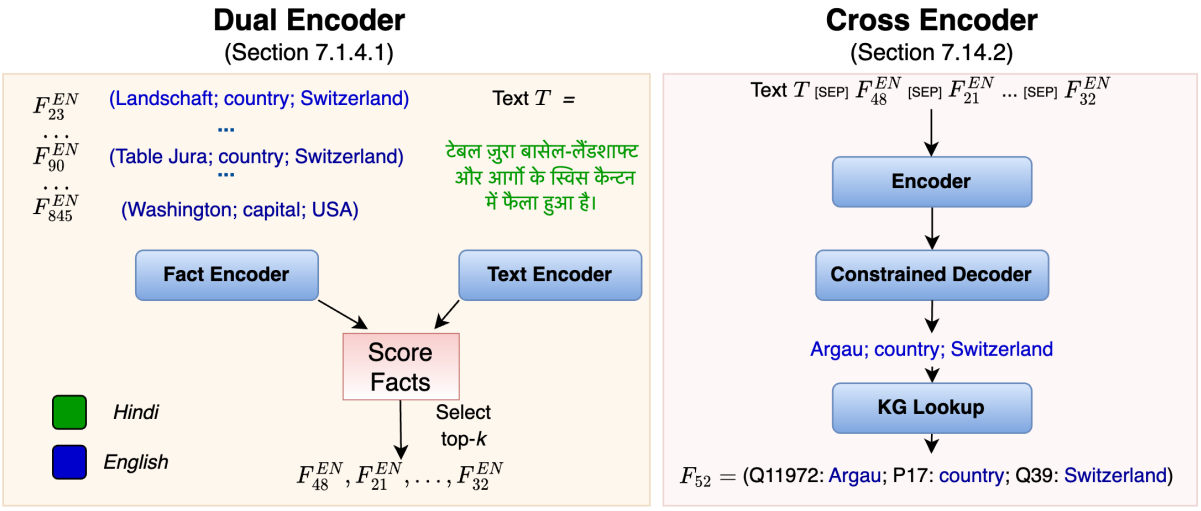

Figure 7.2: REFCOG architecture for linking Hindi sentences with KG facts (using their English surface forms). Fact-Text Dual Encoder scores the text, T, with all the KG facts, F_i, and outputs the top-k facts. A generative Seq2Seq model encodes the text T concatenated with top-k retrieved facts. A constrained decoder is then used to generate the correct fact.

experimented with a Seq2Seq cross encoder. We explain the dual encoder and cross encoders used below.

7.1.4.1 Fact-Text Dual Encoder for Retrieval

Dual encoders (DE) are generally used for neural retrieval. They independently encode the input and target text and use the cosine similarity between the embeddings to assess their relevance. This strategy can be scaled to millions of KG facts as all the facts can be encoded beforehand, independent of the input text. In the case of MFL, as shown in Figure 7.2, the input text T is used to retrieve the closest set of facts defined in the KG. The input text and facts are encoded using the text and fact encoders. We build an approximate nearest neighbors index of the fact embeddings using FAISS (Johnson et al., 2019), which can be used to retrieve the top-k facts efficiently for a given text embedding. We initialize both the encoder parameters with LaBSE (Feng et al., 2020) weights as it is pre-trained for cross-lingual text retrieval over 109 languages. In Section 7.1.6.3, we explore various choices for choosing the language of fact surface form, F_i^{Ret}, used for retrieval.

7.1.4.2 Cross Encoders for Re-ranking

Since dual encoders encode the source and target independently, they fail to capture fine-grained interactions between them. Therefore, cross encoders take the input text T and the closest facts returned by the dual encoder as an input and rescore them to get the final ranked list of facts present in T. In prior work (Botha et al., 2020), classification-based cross encoders are often used to re-score the retrieved results. The re-scoring is done by concatenating input text with each retrieved result, allowing for inter-attention between the input text and fact surface form. We explore two classification-based based cross encoder architectures (shown in Figure 7.3) for re-ranking the top-k facts returned by a retrieval model ($F_{T1}, F_{T2}..F_{Tk}$). We further introduce a novel generation-based cross encoder. The three types of cross encoders are explained below.
Independent Classification (INDCLS): The input text T is concatenated with the textual descriptions of the retrieved facts, $[F_{Ti}]^{Rnk}$ (the language chosen for ranking may be different

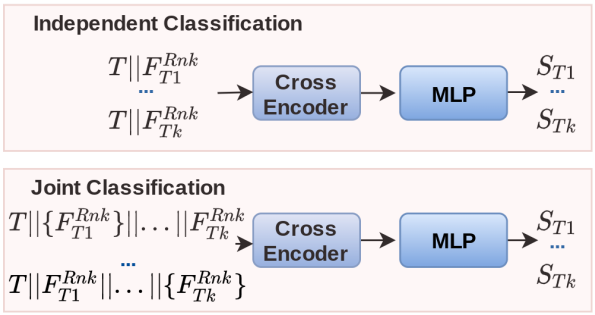

Figure 7.3: Independent and Joint Classification for re-ranking the facts output by the retrieval model.

from that used for retrieval, $[F_{Ti}]^{Ret}$) and passed through the cross encoder. An MLP operates on the pooled embedding to produce a score between 0 and 1, indicating its confidence in whether the fact is expressed in the text. The scoring function can be expressed as,

$$\textsc{IndCls}(F_{Ti}) = \text{MLP}(\text{CE}(T||F_{Ti}^{Rnk})) \tag{7.2}$$

In this case, each fact is considered independently and does not use the dependencies that may exist among the facts.

Joint Classification (JntCls): To score the facts jointly, the score of the *i*th fact, F_{Ti} is computed by concatenating the text T along with all the fact surface forms. The *i*th fact is specially marked with a beginning { and ending }, so the model is aware of the fact that must be scored. This concatenated string is embedded using the cross-encoder. Like the Independent Classification case, the MLP scores the pooled embeddings. Since the embeddings are pooled from all the facts, the model can also capture the dependencies among facts.

$$\textsc{JntCls}(F_{Ti}) = \text{MLP}(\text{CE}(T, F_{T1}^{Rnk}||\ldots||\{F_{Ti}^{Rnk}\}||\ldots||F_{Tk}^{Rnk})_{F_{Ti}}) \tag{7.3}$$

Constrained Generation Cross Encoder: Seq2Seq models can be used as cross encoders that take as input the concatenated text and retrieved facts and use a decoder to generate the most confident fact. This allows the model to produce facts not returned by DE as well, overcoming the issue of error propagation in re-ranking systems whose performance is limited by the facts returned by the dual encoder. To ensure that only valid KG facts are generated, following mGENRE (Cao et al., 2021), we constrain the beam search of the decoder to generate tokens that follow a trie constructed from the English surface forms of KG facts. All facts considered have an English surface form available (Table 7.2). To achieve this, at every decoding step, the vocabulary is restricted to the tokens the trie allows to follow the prefix generated so far. We note that the current formulation is optimized for generating a single best fact, and we leave it to future work to extend it to produce a set of facts. For now, we consider the top-k facts resulting from the beam search as the system output. The architecture is described in the below equation:

$$\textsc{CoGen}(T) = \textsc{Trie-Dec}(\textsc{Enc}(T||F_{T1}^{Rnk}\ldots||F_{Tk}^{Rnk})) \tag{7.4}$$

CoGen stands for constrained generation, Trie-Dec stands for Trie-based decoding and Enc stands for the encoder. We initialize the encoder and decoder weights from the trained mGENRE model. For a fair comparison, we also use the same initialization for the cross-encoders in the classification models.

KG Lookup: We maintain a dictionary of the fact id and corresponding fact surface form (constructed from the entity and relation surface forms). We take the predicted fact surface and look

up this dictionary to get the corresponding fact id. In case constraints are not applied, then the decoder may generate invalid fact surface forms not present in the dictionary.

We refer to DE+CoGen method as Retrieval based Fact-Constrained Generation (REFCOG). The method's schematic is displayed in Figure 7.2, also illustrating the benefits of using the CoGen model, which can do joint rescoring of facts while understanding and encoding the dependencies between them. As a coarse-grained embedding model, the dual-encoder may sometimes retrieve irrelevant facts. For example, besides relevant facts such as (Landschraft; country; Switzerland) and (Table Jura; country; Switzerland), it may also incorrectly attribute a high score to (Washington; capital; USA). This may have happened due to incorrect entity linking between the text T in Hindi and the fact in English. However, when provided with all these facts, the cross-encoder can discern that the fact (Washington; capital; USA) is out of context and accordingly assign it a lower score.

7.1.5 Experimental Setting

Dataset: As described in Section 7.1.4, we use the auto-translated examples of WebRED (total 186K examples with 31K in EN, HI, TE, TA, UR, GU and 0 for AS) to train the model and the manually translated test examples of INDICLINK for evaluation. We randomly choose 5% of the training examples to be used as validation set for early stopping during model training.
Evaluation Metrics: We compare the facts predicted by the model with the gold set of facts and report the value of Precision@1 (P@1) and Recall@5 (R@5). P@1 is the fraction of examples where the most confident fact is contained in the gold set.[3] R@5 is the fraction of gold facts that are present in the top-5 predicted facts. We also compute **macroP@1**, the macro-average counterpart to P@1, where the gold facts are divided into relation-specific classes, performance computed independently in each class and then averaged across all classes.
Implementation: We implement all models in Pytorch framework. We use Sentence Transformers (Reimers and Gurevych, 2019) library for training dual encoder models and use GENRE codebase[4] and fairseq (Ott et al., 2019) library for implementing the various cross encoders. We train the dual encoder and generation models for five epochs each. Total training time is 6 hrs on A100 GPU.

7.1.6 Experiments

We conduct experiments to address the following three questions:

- How well does REFCOG, a retrieve+generation architecture, work for Multilingual Fact Linking, especially when compared to retrieve+reranking models for the task? (Section 7.1.6.1)
- What is the effect of different components in generative decoding? (Section 7.1.6.2)
- How to effectively utilize multilingual fact surface forms during retrieval as well as generative decoding stages of REFCOG? (Section 7.1.6.3)

We note that translation systems may not be available at inference time for certain languages. For example, Assamese, one of the languages we consider in our experiments, is currently not supported by Google Translate. Hence, in our experiments, we aim to evaluate multilingual

[3] NULL is also considered as a separate fact for measuring performance.
[4] https://github.com/facebookresearch/GENRE

fact linking by relying on the cross-lingual ability of the trained model rather than test-time translation.

7.1.6.1 Effectiveness of ReFCoG

In Table 7.3, we compare the results of various models trained on IndicLink. All the cross encoder models use English fact surface forms and $DE_{ALL\text{-}Sum}$ for retrieval (which is explained in detail in Section 7.1.6.3). We notice that the DE models particularly struggle with retrieving Null fact, as its surface form ("None") does not have any word overlap with the sentence. Therefore, in IndCls-N and JntCls-N, we deterministically add Null to the input, along with the remaining DE outputs. This is not required for the ReFCoG model as it is free to generate facts even if they are not returned by the retrieval module. ReFCoG model outperforms classification based re-ranking by 10.7, 15.2 pts in P@1, R@5, respectively. Within the re-ranking models, IndCls achieves better performance compared to JntCls, indicating that providing the other facts tend to confuse the model.

Model	EN	HI	TE	TA	UR	GU	AS	Average	
	P@1	P@1	P@1	P@1	P@1	P@1	P@1	P@1	R@5
$DE_{ALL\text{-}Sum}$	37.5	29.7	32.8	27.8	28.7	29.9	13.8	28.6	31.9
+IndCls	26.3	20.9	23.5	20.3	20.1	22.7	9.6	20.5	31.9
+IndCls-N	46.2	41.8	44.3	42.5	43.8	40.9	29.5	41.4	36.3
+JntCls	13.8	13.8	12.7	13.1	11.9	14.9	5.9	12.3	31.9
+JntCls-N	38.5	38.7	39.9	38.4	38.4	38.6	34.0	38.1	36.3
$ReFCoG_{ALL\text{-}Sum,\ EL}$	**56.4**	**52.4**	**53.4**	**53.2**	**53.6**	**52.5**	**43.1**	**52.1**	**51.5**
-Constraints	55.9	52.3	**53.4**	52.8	53.4	52.4	42.3	51.9	49.5
-SRO links	42.0	38.8	38.9	35.8	38.2	36.9	32.4	37.6	21.4
-DE	50.2	48.7	49.1	47.4	47.5	47.7	39.6	47.2	45.6

Table 7.3: Comparison of different models on the IndicLink dataset. ReFCoG with All-Sum dual encoder and El cross encoder, outperforms independent (IndCls) and joint (JntCls) classification based re-ranking on top of $DE_{ALL\text{-}Sum}$. Ablations indicate the importance of DE and joint prediction of S, R and O for the ReFCoG model. Constraints reduce the P@1, R@5 metrics but ensure production of only valid facts. Please see Section 7.1.6.1 and Section 7.1.6.2 for further details.

7.1.6.2 ReFCoG ablations

In Table 7.3, we consider four variants of the ReFCoG model: (1) ReFCoG w/o Constraints: after removing the constraints on the decoder beam search, (2) ReFCoG w/o SRO links: predicting the subject (S), relation (R) and object (O) of the fact independent of one another, (3) ReFCoG w/o DE: removing DE facts from Cross Encoder input, and (4) ReFCoG w/o DE, Constraints: removing DE facts and constraints.

Removing constraints leads to generation of incorrect facts in 865/6253 examples and reduces performance by 0.2 pts in P@1 and 2 pts R@5, respectively. Instead of predicting the entire fact jointly, we perform an ablation in which we predict each of the components independently. This leads to reduction in performance of 18.1 pts in P@1, showing the importance

of jointly predicting all the components. Unlike the re-ranking models, REFCOG can generate facts even in the absence of Dual Encoder retrieved facts. This allows us to evaluate the model performance without the Dual Encoder. The results indicate that DE facts are responsible for (4.9, 5.9) pts improvement in P@1, R@5.

7.1.6.3 Effect of Multilingual Fact Surface Forms

We consider fact surface forms in English (EL), the language of input text T — Text Language (TL), English and TL combined (ETL), and all the languages in IndicLink (ALL).

For the dual encoder, we either form a single embedding for the fact by concatenating the surface forms in various languages or embed the different language surface forms separately. In either case, we only consider languages for which the fact has a surface form available. The score for a fact D is computed as the cosine similarity between the text embedding and fact embedding. If multiple embeddings are associated with a fact, we aggregate their individual scores through a sum/max operation. The various types of fact surface forms and scoring operations can be summarized as follows:

- EL (or TL): Using the English (or Text language) surface form of the fact.
- ETL-Concat (or ALL-Concat): Using the concatenated English and Text language surface form of the fact (or concatenation of all language surface forms available).
- ETL-Max (or ALL-Max): Embed the surface forms in each language in ETL (or ALL) separately and consider the max of their cosine similarity as the score for the fact.
- ETL-Sum (or ALL-Sum): Use the sum of scores of surface forms in each language in ETL-Sum (or ALL-Sum).

We find that embedding the fact surface forms separately and adding their individual scores leads to the best performance across all languages in P@1, R@5.

However, we don't find similar improvements in concatenating the various language surface forms for cross encoders. To study this further, we also compute the macroP@1 in Table 7.5. The results on the Complete test set indicate that using language fact surface forms other than English does not help in P@1 but results in a modest increase of 3.7 pts in macroP@1 for ETL-Concat. This indicates that other language fact surface forms can help in facts that involve less-frequently occurring relations. Also, language fact surface forms are not consistently available in all languages. So we construct a modified test set that uses only KG facts where fact surface forms are available in all languages and the test examples that can be answered with these KG facts. On this subset, ETL-Concat shows an increase of 19.6 pts in macroP@1 compared to using only English fact surface forms. This shows that robustly handling the sparse availability of fact surface forms can improve performance. ALL-Concat does not seem to improve performance over ETL-Concat, which may be due to larger sequence lengths when all language surface forms are concatenated.

7.1.6.4 REFCOG Error Analysis

We analyze the errors made by REFCOG and classify them into the following three types:

- **Rare relations**: The model struggles with facts that contain rare relations, which can be traced back to WebRED data where the top-10 relations are responsible for 80% of the examples.

Model	Fact Surface Form	EN	HI	TE	TA	UR	GU	AS	Average	
		P@1	P@1	P@1	P@1	P@1	P@1	P@1	P@1	R@5
DE	EL	27.2	19.8	20.8	16	19.1	19.6	8.3	18.9	26
	TL	24.3	13.3	14.0	10.2	12.2	12.1	5.1	13.2	20
	ETL-Concat	26.6	21.3	25.6	17.5	23.6	22.7	6.9	20.8	28.4
	ALL-Concat	28.2	19.9	21.1	19.1	21.2	19	7.5	19.6	27
	ETL-Max	27.8	24.1	24.9	20.2	21.9	23.0	5.4	21.2	28.9
	ETL-Sum	27.8	25	26.4	20.7	21.2	23.5	6.2	21.6	29.3
	ALL-Max	31.1	25.8	25.5	22.2	24.1	24.9	10.1	23.4	26.1
	ALL-Sum	37.5	29.7	32.8	27.8	28.7	29.9	13.8	28.6	31.9
REFCOG	EL	56.4	**52.4**	53.3	**53.2**	**53.6**	51.9	**43.1**	**52.1**	**51.5**
	TL	55.2	47.2	48.8	47	47.6	47.8	33.3	46.9	43
	ETL-Concat	57	49.7	**53.5**	49.9	51.8	50.2	41	50.6	50
	ALL-Concat	**57.1**	51.5	53.3	50.4	51.1	**52.1**	40.1	50.9	50

Table 7.4: Multilingual fact surface forms in Retrieval and Generation models (Section 7.1.6.3). EL, TL, ETL and ALL correspond to descriptions in English, language of input text *T*, EL+TL and all languages, respectively. Concat, Max and Sum refer to concatenation, max and sum scoring operations. For REFCOG, we use ALL-Sum facts for retrieval and experiment with different fact surface forms for cross-encoder.

Fact Surface Form	Complete		Subset	
	P@1	macroP@1	P@1	macroP@1
EL	**52.1**	12	65.2	35.4
TL	43	6.1	63.7	43
ETL-Concat	50.6	**15.7**	**67.5**	**55**
ALL-Concat	50.9	14.7	64.2	54.8

Table 7.5: P@1, macroP@1 of REFCOG with fact surface forms in various languages at cross encoder stage. The macroP@1 is evaluated for the Complete test set as well as the Subset where descriptions are available in all languages. Improvement in macroP@1, indicates stronger performance on facts with less-frequently occurring relations.

- **NULL fact**: In 33% of the examples, the model mistakenly predicts a fact even when the gold fact is NULL, demonstrating the difficulty in detecting the absence of any facts.
- **Issues with Gold**: A few examples contain relations that are not explicitly implied by the sentence but require background world knowledge.

7.1.7 Effectiveness of REFCOG for linking Open IE tuples

In the previous sections, we studied the effectiveness of the REFCOG model for linking sentences to facts in WikiData using the INDICLINK dataset. In this section, we aim to test the effectiveness of REFCOG for linking Open IE tuples of the format (subject phrase; relation phrase; object phrase) to facts in WikiData. Considering that we have trained Gen2OIE models (Section 3.2) for English and two Indic languages, Hindi and Telugu (in Chapter 6), we choose these three languages for testing the KG linking capabilities.

We generate extractions from all the sentences of IndicLink in these three languages using the Gen2OIE system. The generated extractions are converted into the form of a sentence by concatenating the subject, relation, and object phrases to form an *ext-sentence* (Section 6.2.2). These ext-sentences are passed through the trained ReFCoG system to obtain the associated KG facts for each ext-sentence.

As there is no ground truth available for the KG facts associated with each extraction, evaluating the performance of the task is a challenge. Therefore, we resort to evaluation at the sentence level, which gives an upper bound of the performance on the actual extraction-level task. We take the set of facts linked for each of the extractions associated with the original sentence. Each of the facts is also associated with a score that indicates the confidence of the fact being linked to the ext-sentence. We aggregate the facts from multiple ext-sentences by summing up their associated scores when the same fact appears multiple times.

This gives us a ranked set of facts for every sentence. Thus, we now use the standard IndicLink evaluation mechanisms to evaluate the quality of the set of linked facts. By evaluating the accuracy at the sentence level, instead of directly at the extraction level, we obtain an upper bound for the expected performance at the extraction level. The results are reported in Table 7.6.

We observe that the overall performance drops when we used extractions for linking, compared to directly using the sentences, with as much as a 4.1 pts drop in R@5 on the average of three languages. This is possibly because the Open IE extractions may be missing important contextual information when viewed independently from the sentence from which it is extracted. For example, a pronoun may be used in the extraction while the object of the pronoun may not be present in the extraction but is present only in the sentence. Due to the lack of this contextual information, the extraction cannot be linked correctly to the KG. Therefore, there is significant scope for improvement in developing systems that can link Open IE tuples with KG facts and also make Open IE extractions contextually independent.

ReFCoG$_{\text{ALL-Sum, EL}}$	EN		HI		TE		Average	
	P@1	R@5	P@1	R@5	P@1	R@5	P@1	R@5
Sentence	**56.4**	**69.4**	**52.4**	**53.4**	53.3	**55.8**	**54.0**	**59.5**
Extraction	56.3	60.8	52.2	50.1	**53.4**	55.3	53.9	55.4

Table 7.6: Evaluation of KG facts linked to Open IE extractions.

7.2 Open Knowledge Base Completion

In the previous section, we have looked at how to link natural language text or Open IE triples to existing canonical KBs. This section looks at Open Knowledge Bases (Open KBs), particularly completing the Open KBs. Open KBs represent one application of Open IE whose extractions are used to generate these Open KBs. However, like canonical KBs, the Open KBs are also often found to be incomplete, either because of reliance on the output of imperfect Open IE systems or the lack of complete information in the base text that is used to generate the Open IE extractions. Therefore, Open Knowledge Base Completion (OKBC) systems aim to discover new links between nodes of an Open Knowledge Base constructed using Open IE systems. However, current OKBC systems score each entity pair independently. This results in missed information that could be captured by considering interactions among different entities. Therefore, we develop a novel Cross-Entity Aware Reranker (CEAR) model that jointly scores the top-k entities

obtained from embedding-based KBC models, using cross-entity attention in BERT.[5]

7.2.1 Related Work

Many Knowledge Graph Embedding (KGE) methods have been proposed for the task of KBC in the closed KB setting (Bordes et al., 2013; Kazemi and Poole, 2018; Lacroix et al., 2018; Jain et al., 2018), which use various scoring functions to evaluate the plausibility of triples. RotatE (Sun et al., 2019) defines relations as rotations from the source entity to the target entity in a closed vector space. ComplEx (Trouillon et al., 2016) defines embeddings in a complex vector space and uses a Hermetian dot product to calculate a scalar score for the triple.

With the rise of pre-trained language models in NLP, prior works have also explored the use of BERT for Knowledge Base Completion (Yu et al., 2020; Kim et al., 2020; Shah et al., 2020). KG-BERT (Yao et al., 2019) uses BERT to score all possible triples, formed by the concatenating the input query , either subject/relation or relation/object, with each entity in the KB. Since each answer entity is scored independently, KG-BERT does not benefit from cross-entity attention. Pre-train KGE (Zhang et al., 2020a) uses BERT to initialize entity/relation embeddings used by TransE (Bordes et al., 2013).

Since Open KBs use fact triples generated using Open IE systems, the un-normalized surface forms of entities and relations make link prediction challenging. Open Link Prediction (Broscheit et al., 2020) provides a benchmark for this task, using OPIEC KB (Gashteovski et al., 2019), and we perform experiments on the same.

7.2.2 CEAR: Cross-Entity Aware Reranker

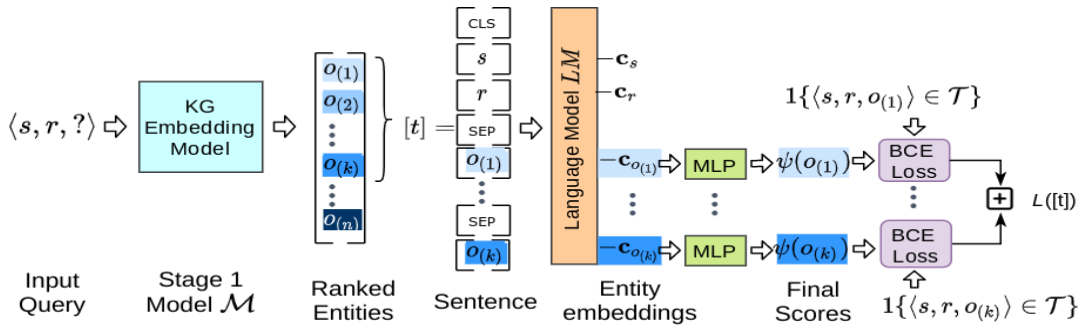

Figure 7.4: The two stage architecture. Stage 1 model outputs top-k entities that the Stage 2 model uses to generate contextual entity embeddings. The embeddings are passed through an MLP to get the final score for each entity.

Notation and Task Description: Recall that, we assume a KB with relations $\mathcal{R}$, entities $\mathcal{E}$ and an incomplete set of fact-triples $\mathcal{T}$ (also referred to as just *facts*). Each fact in $\mathcal{T}$ is represented as a triple $(s; r; o)$, where the subject entity $s \in \mathcal{E}$, is related to the object entity $o \in \mathcal{E}$ via the relation $r \in \mathcal{R}$. Given a query $(s; r; ?)$ (or $(?; r; o)$), the task of link prediction (or KBC) is to find new facts $(s; r; o) \notin \mathcal{T}$. A KB can be represented as a graph, $\mathcal{G}$, where for each fact $(s; r; o) \in \mathcal{T}$, the nodes representing the entities s and o are connected by a labeled edge r. We represent an embedding-based model for KBC as $\mathcal{M}$ and its scoring function as $\mathcal{S}^{\mathcal{M}}$. $\mathcal{M}$ learns embeddings of all the entities and relations such that $\mathcal{S}^{\mathcal{M}}(\boldsymbol{e_s}, \boldsymbol{w_r}, \boldsymbol{e_o})$ is high whenever $(s; r;$

[5]Mayank Chauhan helped with the ideation and implementation of the CEAR architecture and included it as part of his MTech thesis.

$o) \in \mathcal{T}$. Here $\boldsymbol{e_s}$, $\boldsymbol{e_o}$ and $\boldsymbol{w_r}$ represent the learnt embeddings of the entities s, o and relation r respectively.

Further, we assume that we are given the surface form (or description) of the entities and relations in some natural language l, along with a pre-trained language model LM, trained on some external corpus of text in the language l. As defined in Section 7.1.2, s^l and r^l represent the entity and relation surface forms in language l. Let $[s]$ (or $[r]$) represent the sequence of words in the English surface form of an entity $s \in \mathcal{E}$ (or relation $r \in \mathcal{R}$).[6] The language model LM takes a sentence, $[t]$, as input and generates a contextual embedding for each token in the sentence $[t]$.

In the context of an Open KB, we adhere to a similar setting where every unique surface form is treated as a distinct node or edge in the graph representation of our knowledge base, irrespective of the fact that multiple surface forms may represent the same real-world entity. Consider an entity 'Barack Obama' with surface forms TP_{BO} = { 'Barack Obama', 'B. Obama', $\cdots$ }. In our formulation, even though these surface forms refer to the same real-world entity, each form in TP_{BO} is considered a distinct node in the graph. That is the nodes for 'Barack Obama' and 'B. Obama' are distinct even though they refer to the same entity.

Motivation: Embedding-based models score all the entities independently for a given query (s; r; ?), *i.e.*, $\forall i \neq j$, the scores $\mathcal{S}^{\mathcal{M}}(\boldsymbol{e_s}, \boldsymbol{w_r}, \boldsymbol{e_{o_i}})$ and $\mathcal{S}^{\mathcal{M}}(\boldsymbol{e_s}, \boldsymbol{w_r}, \boldsymbol{e_{o_j}})$ of the entities o_i and o_j are computed independent of each other. Note that such a model exploits only the structural information present in the corresponding graph. Further, these models treat the entities and relations as atomic objects, making their score oblivious to the information present in the surface forms.

In contrast, using BERT, models such as KG-BERT (Yao et al., 2019) exploit the information in the surface forms of entities and relations. But, similar to embedding based models, they also score each entity independently, ignoring the relationship between the entities that may help in finding the correct entity.

CEAR not only exploits the benefits of both approaches, but it also overcomes their common shortcoming of scoring entities independently. It follows a two-stage approach, similar to the Dual Encoder and Cross Encoder pipeline used in Section 7.1.4. It is described below.[7]

Stage-1: Score using an Embedding model $\mathcal{M}$

In the first stage, CEAR exploits the structural information present in the graph, $\mathcal{G}$. It trains an embedding based model $\mathcal{M}$, to rank all the entities $o_j \in \mathcal{E}$ based on their score $\mathcal{S}^{\mathcal{M}}(\boldsymbol{e_s}, \boldsymbol{w_r}, \boldsymbol{e_{o_j}})$ for a given query (s; r; ?). Once such a model is trained, we pick the top-k answers $o_{(j)}$ for any query and pass them to Stage-2 (described below) for re-ranking them based on their surface forms and world knowledge in the pre-trained language model. The value of k depends on the capacity of the pre-trained language model LM, which usually restricts the maximum number of tokens in a sentence that it can process.[8]

For closed KBs, we employ either the ComplEx or RotatE models as the embedding model $\mathcal{M}$, whereas, for Open KBs, we use their variants that have an LSTM to encode the surface form of the entity/relation (Broscheit et al., 2020). This is particularly important in Open KBs to gain information from the surface forms, as uncanonicalized Open KBs have sparse connectivity for individual nodes. Using the surface forms in the model helps alleviate this issue by allowing it to learn from other nodes with semantically similar surface forms and possibly represent the same entity. We refrain from directly depending on entity canonicalization (Vashishth et al., 2018) due to the current systems' lack of reliability. Any potential errors stemming from these

[6] We consider only English for the current task and therefore drop the language notation.

[7] The code is released at https://github.com/dair-iitd/CEAR/

[8] BERT has a 512 word-pieces limit

systems could lead to a cascade of additional errors throughout other parts of the system.

Stage-2: Re-rank using Language Model LM

For a query $(s; r; ?)$[9], the surface form ($[o_{(j)}], j = 1 \ldots k$) of the top-$k$ entities retrieved from stage 1 are used along with the surface form ($[s]$ and $[r]$) of the head entity and relation to create a sentence in the natural language $\mathcal{L}$. Specifically, $[t]$ = [CLS]$[s]$[SPC]$[r]$[SEP]$[o_{(1)}]$[SEP] $\ldots$ [SEP]$[o_{(k)}]$, represents such a sentence, where [CLS] is a special token to mark the beginning of a sentence; [SPC] is a special token used to separate the subject from the relation; and [SEP] is a special token separating different answers from each other as well as from the query. Note that the description of an entity or relation may contain multiple tokens.

The sentence $[t]$ is fed as input to the language model LM, which generates a context-aware embedding for each token in the sentence. The embeddings of the tokens belonging to a candidate entity $o_{(j)}$ are mean pooled to create its final embedding, $\boldsymbol{c_{o_{(j)}}}$. Such an embedding is aware of not only the query entity and relation but also the other plausible answers. Such *cross-entity aware* embeddings make use of additional context that is helpful to answer the query $(s; r; ?)$.

Finally, $\boldsymbol{c_{o_{(j)}}}$ is passed through an MLP M to generate its final score, $\psi(o_{(j)})$, *i.e.*, $\psi(o_{(j)}) = M(\boldsymbol{c_{o_{(j)}}})$. Thus, the top-$k$ entities from Stage-1 are re-ranked based on their final scores. The LM is trained by minimizing the standard Binary Cross Entropy (BCE) loss, $L([t])$ for the sentence $[t]$, computed using the final scores $\psi(o_{(j)}), \forall j = 1 \ldots k$,

$$L([t]) = -\sum_{j=1}^{k}(L_p^j + L_n^j) \tag{7.5}$$

where $L_p^j = Ind\{\langle s, r, o_{(j)}\rangle \in \mathcal{T}\} \log(\sigma(\psi(o_{(j)})))$ and $L_n^j = Ind\{\langle s, r, o_{(j)}\rangle \notin \mathcal{T}\} \log(1 - \sigma(\psi(o_{(j)})))$, Ind is the indicator function and σ is the standard sigmoid function.

7.2.3 Experimental Setting

Datasets: We consider the open link prediction dataset, OLPBENCH. OLPBENCH proposes multiple train sets based on test set leakage removal. We use the most difficult train data set called *thorough train dataset*, which contains the harshest test evidence removal. Table 7.7 contains the various statistics associated with the dataset.

Dataset	Entities	Relations	Train	Valid	Test
OLPBENCH	2.47M	961K	30.6M	10K	10K

Table 7.7: Statistics of the dataset used.

Evaluation: Link prediction performance is the average of head entity and tail entity prediction. Evaluation is done under filtered settings where the model is not penalized for ranking entities appearing with query in train and val sets higher than the gold entity. We report MRR, HITS@N metrics.

Baselines: For OLPBENCH, we compare with the state-of-the-art ComplEx-LSTM, which uses LSTM embeddings in a ComplEx model (Broscheit et al., 2020). Considering the large number of target entities in OLPBENCH, we also experiment with ExtremeText (Wydmuch et al., 2018), an extreme classification model, which builds a hierarchical softmax tree (Morin and Bengio, 2005) over FastText (Joulin et al., 2016) embeddings. While training ExtremeText, we enriched the query by appending it with the top-5 most frequent entities seen with the relation in training data. We use ExtremeText and Complex-LSTM as Stage-1 in REFCOG.

[9]Similar formulation holds for head-entity prediction

Method	H1	H10	H50
ComplEx-LSTM	2.1	7.0	14.6
ExtremeText	6.4	16.3	**26.0**
CEAR (ComplEx-LSTM)	3.8	9.1	14.6
CEAR (ExtremeText)	**7.4**	**17.9**	**26.0**

Table 7.8: Link Prediction performance on OLPBENCH.

Model	Dataset	k=10	20	30	40
CEAR (ExtremeText)	OLPBENCH	6.9	7.1	**7.4**	6.8

Table 7.9: H@1 with increasing top-k Stage-1 samples.

7.2.4 Experiments

In Table 7.8, we evaluate the link prediction performance on OLPBENCH using different methods. We find that ExtremeText performs 4.3 HITS@1 higher than the previous state of art model, LSTM-ComplEx. This demonstrates the effectiveness of modeling the task as an extreme classification problem over the 2.47 million entities. We observe consistent gains by applying Stage-2 BERT on top of both LSTM-Complex (+1 HITS@1) and ExtremeText (+1 HITS@1). Thus our final model CEAR(ExtremeText) represents a 5.3 HITS@1 gain over the current state of art model, LSTM-ComplEx. We trained the Stage-2 model with only a fraction of the training data (1M out of 30M available) as we didn't observe much performance gains on adding more examples. Note that we train the Stage-2 model with only a fraction of the training data (1M out of 30M) due to the computational costs associated with training BERT.
Ablation: In Table 7.10, we compute the performance of CEAR(ExtremeText) by (1) replacing pretrained BERT parameters with random initialization, (2) scoring each Stage-1 entity independently (similar to KG-BERT applied only on Stage-1 entities), and (3) randomly shuffling top-k Stage-1 entities before passing them to Stage-2. We find that all three components of CEAR are essential for achieving the final model performance. Apart from pretrained knowledge, knowing all the top-k Stage-1 entities (in ranked order) is crucial for the model performance.

7.3 Conclusion

In this chapter, we look at the connection between Open IE and Knowledge Bases. We introduce the task of multilingual fact linking for connecting natural language text or Open IE triples to

Model	OLPBench	
	H1	H10
CEAR (ExtremeText)	**7.4**	**17.9**
- Pretraining	6.0	15.6
- CE Attention	5.0	16.3
- Stage1 Ranks	6.2	16.8

Table 7.10: Ablation of the best CEAR model, which shows the importance of BERT pretrained knowledge, Cross-Entity Attention and Stage-1 Entity Ranks.

KBs and present a new evaluation dataset IndicLink containing examples in English and six Indian languages. We explore various dual encoder and cross encoder architectures and find that the proposed Retrieval+Generation model, ReFCoG, outperforms classification-based reranking systems by 10.2 pts in P@1. We also presented a novel Open KBC model, CEAR, that uses the pretrained parameters in BERT and global view of competing entities (using cross-entity attention) to achieve a new state-of-the-art performance for link prediction on the OLPBench Open KBC dataset.

Chapter 8

Conclusion and Future Work

Open Information Extraction, paralleling the broader development in the field of NLP, has gone through a paradigm shift in the last few years, moving from primarily rule-based or statistical systems to deep learning-based systems. The move to deep learning has opened up a wide range of possibilities to develop more robust systems that can support multiple languages and additional features. It also presents exciting challenges to use the generated Open IE triples in conjunction with other neural models developed to use them in downstream applications.

In this dissertation, we have addressed multiple aspects of extending Open IE systems to use the latest advances in deep learning for developing monolingual and multilingual models. In particular, we have considered techniques for (1) building stronger neural models (IMoJIE, Gen2OIE in Chapter 3, CIGL in Chapter 4), (2) adding support for linguistic phenomena (co-ordination analyzer, proper noun compound interpretation in Chapter 5), (3) curating training data in multiple languages (AACTrans in Chapter 6), and (4) extending it for KB applications (MFL, CEAR in Chapter 7). We also release a new software package OpenIE-6.2.[1]

However, this dissertation only serves as an initial step toward realizing the full potential of deep learning for Open IE. It opens up a broad set of problems for the research community to tackle for realizing the next generation of Open IE systems. Broadly, the potential improvements can be categorised into three classes:

1. Improving the existing neural frameworks used for Open IE:
    - Building non-autoregressive models for Open IE that can be faster.
    - Extending multilingual support for hundreds of languages.
    - Designing better evaluation metrics to test the quality of generated triples.
2. Extending the task of Open IE to novel settings:
    - Adding support for implicit relations that have to be inferred from the sentences.
    - Canonicalizing triples in a task-based fashion can help reduce sparsity.
3. Applying Open IE extractions in service of downstream tasks:
    - Applying to knowledge-seeking applications like Question-Answering.
    - Designing for user-facing tasks to achieve maximum utility.
    - Developing customizable Open IE solutions that can support generating extractions based on templatized patterns.

[1] https://github.com/dair-iitd/openie6

We briefly describe the scope for each of the above categories.

8.1 Non-Autoregressive models

Autoregressive models are fundamentally bottlenecked by their sequential decoding, reducing their inference speed significantly. In comparison, the labeling models are much faster. Still, they are limited by their ability to use only words in the sentence and always have to follow the same order as they appear in the input sentence. Non-Autoregressive models (NAR) (Ren et al., 2020) offer a promising alternative using parallelizable techniques to generate the output sequence. Models like Felix (Mallinson et al., 2020) tag the sequence of tokens that have to be present in the output along with the order in which they have to be present and the points at which new tokens have to be predicted. These models are particularly suitable for the task of Open IE as the extractions are generally biased to include words in the original sentence with a similar order. Since only a limited set of cases need changes in the word order or new words to be introduced, NAR models can potentially allow faster models while maintaining the same levels of accuracy as generative models.

However, some changes in the model are needed to allow the prediction of multiple extractions. Since Open IE involves generating a set of extractions, the set nature of the problem needs to be carefully handled as there is often a tradeoff involved in introducing dependence among the extractions and ensuring fast inference. Solutions from other problems where similar issues with set generation are encountered can help resolve this challenge. For example, image captioning (Vijayakumar et al., 2018) also requires generating a set of textual captions for an image dissimilar to each other.

8.2 Large-scale multilingual support

Developing Open IE support for a broad range of languages is critical for supporting a diverse set of users around the globe. The technique suggested in Chapter 6 provides a good initial solution but cannot be scaled to the size of 100+ languages. This is due to dependence on language-specific resources such as a translation system and the challenges of maintaining a separate model for each language. A good solution to this problem can be found in zero-shot multilingual models already pre-trained to include knowledge of 100+ languages (Xue et al., 2020). Training these models with representatives of various language families can enable generalization to the remaining languages, even without any training data (zero-shot). This will also align with how multilingual support is being embraced for other NLP problems (Kim et al., 2021). However, the challenge remains in managing the linguistic transfer to low-resource languages while maintaining the performance of high-resource languages. Multilingual adapters (Baziotis et al., 2022) can give insights into resolving this inherent tradeoff commonly encountered with multilingual models.

8.3 Evaluation metrics

In the past few years, various evaluation metrics have been proposed for the task of Open IE, which has been summarized in Section 2.2. However, they all need gold standard Open IE triples for comparison, which fundamentally limits the scope and domain of evaluation to a few hundred sentences picked from their respective sources. Moreover, Open IE is a broadly

defined task with independent research groups focussing on varied styles of extractions. Often, their individual biases about what is a good extraction creep in during the creation of gold extractions.

To overcome both of the above challenges, an extrinsic evaluation of Open IE should be adopted to ensure that the extraction quality is determined by an independent measure in a large-scale fashion. For example, the quality of an open-domain question-answering system with the given Open IE extractions as the source can give information about the knowledge contained in the extracted tuples. This also gives the value of distilling these open triples from the corpus instead of the QA retrieval module that makes direct use of the corpus at the query time. Similarly, training a language model on the sentence form of Open IE triples and testing its language perplexity can give information about the grammaticality of the generated tuples.

8.4 Downstream Applications

The initial versions of Open IE had been shown to help downstream applications such as Question Answering (Fader et al., 2013) by providing a large corpus of knowledge to help answer user queries. However, current deep learning-based end-to-end question-answering systems use the raw text as the source of factual knowledge (Borgeaud et al., 2022). The neural models can directly operate on the input text and use only the information required. Hence, Open IE has not been shown to benefit current neural models, and this remains true for other intermediate linguistic tasks such as SRL, pos tagging, dependency parsing or constituency parsing, which find limited to no presence in the current state-of-the-art systems for applications such as questions answering, summarization or dialogue generation. Therefore, it remains a challenge to show the downstream applicability of Open IE in the current neural era.

However, Open IE does show promise in addressing specific challenges that neural models face, like handling longer texts such as multiple documents or aiding in interpretability. The capability of Open IE to extract triples and automatically construct dynamic graphs from multiple documents has been shown to help multi-document summarization (Fan et al., 2019a). Since graphs are a compact way of expressing inter-connected information, they are helpful in tasks like summarization which needs access to all the information in the text. This needs to be further developed for question answering over multiple documents because QA requires narrowing down the important information to answer the query. Since all the information in the text is not required for the task, current retrieval-based neural models have reasonable performance. Moreover, QA benchmarks with different characteristics may further necessitate the importance of Open IE. For example, questions that need to aggregate information from multiple factual statements can potentially benefit from Open IE.

Similarly, neural models for the question-answering struggle to generate explanations for their predictions. Attention-based interpretability methods rely on assigning scores to individual input tokens. However, such attribution methods fail to convey which facts in the input sentences are responsible for the final output. Breaking the input into simple extractions can aid in assigning responsibility to the particular aspects of the sentence that have to lead to the final result.

8.5 Implicit Relations

Relations not explicitly present in the sentence but must be inferred semantically are important for deriving more value from Open IE. For example, the phrase, “Indian Prime Minister Naren-

dra Modi" can have an implicit extraction (Narendra Modi; has nationality; Indian) or the phrase, "Gopal's sister is married to Arjun " can have an implicit extraction (Gopal; is the brother-in-law of; Arjun). Chapter 5 provides a solution for noun compounds, but many unaddressed phenomena remain. Either identifying specific cases where implicit relations are possible and developing solutions for them or developing a generic solution that can automatically handle varied types of implicit relations is necessary. Improving the generation of implicit relations would also provide value over existing end-to-end neural solutions for understanding the text as they keep the knowledge implicitly in the parameters without exposing it to users.

8.6 Entity and Relation Canonicalization

There are many ways to express the same subject, relation or object phrase. For example, "Barack", "Barack Obama" or "Barack H. Obama" represent the same entity while simultaneously being valid entity (subject or object) phrases. As a result, several extractions express similar information. Existing canonicalization schemes rely on clustering either the entities or relations (Vashishth et al., 2018; Krishnamurthy and Mitchell, 2011), but these operate task-independently. They work well for short relation phrases. For example, the relations "was born in" and "took birth at" are easily clustered together by these schemes. However, in the case of complex relations, it may be challenging to assign a unique cluster to them. For example, the relation phrase, "cook and bake," can belong to the cluster for cooking or the cluster for baking. Thus, existing schemes lead to canonicalizations that may be either too big or small and may not be ideal for all tasks. Developing a scheme for jointly canonicalizing both entities and relations in a task-specific manner can lead to the generation of canonical triples which are dense in relevant information content.

8.7 User-Facing tasks

Open IE in its current form has significant benefits as a tool to organize information from a large corpus of text and present it in an easily understandable manner.[2] In general, neural QA systems need to be provided with specific queries that assume a certain level of knowledge about what is present in the corpus. This may not always be possible when domain-specific knowledge is required, such as when searching bio-medical corpora.

Compared to such guided information-seeking systems, Open IE provides a way to explore a corpus in a completely unguided fashion. It can display the common entities and relations in the corpus, which enables one to gain knowledge of the domain quickly and provides a good starting point for exploration. Although Open IE has these benefits, current systems have not been designed explicitly for such unguided explorations. Using HCI techniques, progress needs to be made to understand the best way to present the corpus information to help the end-user gain insights efficiently.

8.8 Customizability

The current version of neural Open IE systems is often biased to generate extractions similar to those seen in training. However, specific patterns in corpora may express information in different styles, which would be missed by the Open IE system (Soderland et al., 2010; Krishnamurthy

[2]https://github.com/knowitall/openie-demo

and Mitchell, 2011). For example, Wikipedia articles usually start with "Name-of-Person (Date-of-Birth to Date-of-Death)" for deceased persons. In the case of "Shinzo Abe (21 September 1954 – 8 July 2022)", generating tuples of the form (Shinzo Abe; date of birth; 21 September 1954) and (Shinzo Abe; date of death; 8 July 2022) would be useful extractions but are missed by current Open IE systems.

Providing solutions that can use user-provided templates to change the generated extractions dynamically would help solve this challenge. The example template for the above case would be "Name-of-Person (Date-of-Birth to Date-of-Death)" –> (Name-of-Person; date of birth; Date-of-Birth), (Name-of-Person; date of death; Date-of-Death). However, treating them as soft templates will allow the potential to use the power of neural generalization. This will allow proper handling of near matches, such as when the input becomes "Shinzo Abe (Japanese, 21 September 1954 – 8 July 2022)". Allowing such on-the-fly customizations to Open IE extractions will enable wider utility.

Bibliography

K. Ahn, J. Bos, D. Kor, M. Nissim, B. L. Webber, and J. R. Curran. Question answering with qed at trec 2005. In *TREC*, 2005.

A. V. Aho and J. D. Ullman. *The theory of parsing, translation, and compiling*. Prentice-Hall Englewood Cliffs, NJ, 1973.

A. Alexiadou. Proper name compounds: a comparative perspective. *English Language & Linguistics*, 2019.

G. Angeli, M. J. J. Premkumar, and C. D. Manning. Leveraging Linguistic Structure for Open Domain Information Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL), 2015*, 2015.

F. Aslam, T. M. Awan, J. H. Syed, A. Kashif, and M. Parveen. Sentiments and emotions evoked by news headlines of coronavirus disease (covid-19) outbreak. *Humanities and Social Sciences Communications*, 2020.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR), 2015*, 2015.

N. Balasubramanian, S. Soderland, Mausam, and O. Etzioni. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

T. Baldwin and T. Tanaka. Translation by machine of complex nominals: Getting it right. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, 2004.

R. Balyan and N. Chatterjee. Translating noun compounds using semantic relations. *Computer Speech & Language*, 2015.

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *International Joint Conference on Artificial Intelligence (IJCAI), 2007*, 2007.

M. Bañón, P. Chen, B. Haddow, K. Heafield, H. Hoang, M. Esplà-Gomis, M. L. Forcada, A. Kamran, F. Kirefu, P. Koehn, S. Ortiz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarrías, M. Strelec, B. Thompson, W. Waites, D. Wiggins, and J. Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2019.

C. Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

L. Barrault, O. Bojar, M. R. Costa-Jussa, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, et al. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, 2019.

A. Bassa, M. Kröll, and R. Kern. Gerie-an open information extraction system for the german language. *J. Univers. Comput. Sci.*, 2018.

C. Baziotis, M. Artetxe, J. Cross, and S. Bhosale. Multilingual machine translation with hyper-adapters. *arXiv preprint arXiv:2205.10835*, 2022.

S. Bhardwaj, S. Aggarwal, and Mausam. CaRB: A Crowdsourced Benchmark for OpenIE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019*, 2019.

A. Bhartiya, K. Badola, and M. . DiS-ReX: A multilingual dataset for distantly supervised relation extraction. In *Association for Computational Linguistics*, Dublin, Ireland, May 2022.

N. Bhutani, H. V. Jagadish, and D. Radev. Nested propositions in open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NeurIPS)*, 2013.

S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, D. De Las Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. Rae, E. Elsen, and L. Sifre. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

J. A. Botha, Z. Shan, and D. Gillick. Entity linking in 100 languages. In *Empirical Methods in Natural Language Processing, EMNLP*, 2020.

T. Breban, T. Breban, and J. Kolkmann. Different perspectives on proper noun modifiers. *English Language & Linguistics*, 2019.

S. Broscheit, K. Gashteovski, Y. Wang, and R. Gemulla. Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

S. Buchholz and E. Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, 2006.

C. Butnariu, S. N. Kim, P. Nakov, D. O. Séaghdha, S. Szpakowicz, and T. Veale. Semeval-2010 task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, 2009.

B. S. Cabral, R. Glauber, M. Souza, and D. B. Claro. Crossoie: Cross-lingual classifier for open information extraction. In *PROPOR*, 2020.

J. Callan, M. Hoy, C. Yoo, and L. Zhao. Clueweb09 data set, 2009.

N. D. Cao, L. Wu, K. Popat, M. Artetxe, N. Goyal, M. Plekhanov, L. Zettlemoyer, N. Cancedda, S. Riedel, and F. Petroni. Multilingual autoregressive entity linking. *ArXiv*, abs/2103.12528, 2021.

X. Carreras and L. Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 2005.

A. Chen, G. Stanovsky, S. Singh, and M. Gardner. MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

J. Christensen, Mausam, S. Soderland, and O. Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, 2011.

J. Christensen, S. Soderland, G. Bansal, and Mausam. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, 2014.

D. B. Claro, M. Souza, C. Castellã Xavier, and L. Oliveira. Multilingual open information extraction: Challenges and opportunities. *Information*, 2019.

L. Cui, F. Wei, and M. Zhou. Neural open information extraction. In *Association for Computational Linguistics (ACL), 2018*, 2018.

A. Daza and A. Frank. X-srl: A parallel cross-lingual semantic role labeling dataset. In *arXiv preprint arXiv:2010.01998*, 2020.

L. Del Corro and R. Gemulla. ClausIE: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web (WWW), 2013*, 2013.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

K. Dong, Z. Yilin, A. Sun, J.-J. Kim, and X. Li. DocOIE: A document-level context-aware dataset for OpenIE. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, 2021.

L. Dong. Amazon product graph, 2017. URL `http://lunadong.com/talks/PG.pdf`.

Z.-Y. Dou and G. Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, and E. Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

O. Ernst, O. Shapira, R. Pasunuru, M. Lepioshkin, J. Goldberger, M. Bansal, and I. Dagan. Summary-source proposition-level alignment: Task, datasets and supervised baseline. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, 2021.

O. Ernst, A. Caciularu, O. Shapira, R. Pasunuru, M. Bansal, J. Goldberger, and I. Dagan. Proposition-level clustering for multi-document summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open Information Extraction: The Second Generation. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, 2011.

A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP Conference*, 2011.

A. Fader, L. Zettlemoyer, and O. Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013.

W. Falcon. Pytorch lightning. *GitHub. Note: https://github.com/williamFalcon/pytorch-lightning*, 2019.

A. Fan, C. Gardent, C. Braud, and A. Bordes. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019a.

A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019b.

M. Fares. A dataset for joint noun-noun compound bracketing and interpretation. In *Proceedings of the ACL 2016 Student Research Workshop*, 2016.

M. Faruqui. Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.

C. Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, 2010.

F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

D. Fensel, U. Simsek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, and A. Wahler. Knowledge graphs: Methodology, tools and selected use cases. *Knowledge Graphs*, 2020.

J. Ficler and Y. Goldberg. Coordination annotation extension in the Penn Tree Bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016a.

J. Ficler and Y. Goldberg. Coordination annotation extension in the penn tree bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016b.

J. Ficler and Y. Goldberg. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016c.

C. J. Fillmore. Frames and the semantics of understanding. *Quaderni di semantica*, 1985.

L. Galárraga, G. Heitz, K. P. Murphy, and F. M. Suchanek. Canonicalizing open knowledge bases. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014.

P. Gamallo and M. Garcia. Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, pages 711–722. Springer, 2015.

J. Gao, C. Sun, Z. Yang, and R. Nevatia. Tall: Temporal activity localization via language query. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 2018.

K. Gashteovski, R. Gemulla, and L. d. Corro. MinIE: minimizing facts in open information extraction. In *Association for Computational Linguistics (ACL), 2017*, 2017.

K. Gashteovski, S. Wanner, S. Hertling, S. Broscheit, and R. Gemulla. Opiec: an open information extraction corpus. *arXiv preprint arXiv:1904.12324*, 2019.

P. Groth, M. Lauruhn, A. Scerri, and R. Daniel Jr. Open information extraction on scientific text: An evaluation. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

J. Gu, Z. Lu, H. Li, and V. O. K. Li. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of Association for Computational Linguistics (ACL), 2016*, 2016.

R. Guarasci, E. Damiano, A. Minutolo, M. Esposito, and G. De Pietro. Lexicon-grammar based open information extraction from natural language sentences in italian. *Expert Systems with Applications*, 2020.

N. Habash and B. Dorr. A categorial variation database for English. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 96–102, 2003. URL `https://aclanthology.org/N03-1013`.

L. He, M. Lewis, and L. Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

L. He, K. Lee, M. Lewis, and L. Zettlemoyer. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. C. Russell. Localizing moments in video with natural language. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

I. Hendrickx, P. Nakov, S. Szpakowicz, Z. Kozareva, D. O. Séaghdha, and T. Veale. Semeval-2013 task 4: Free paraphrases of noun compounds. *arXiv preprint arXiv:1911.10421*, 2019.

J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

P. Hohenecker, F. Mtumbuka, V. Kocijan, and T. Lukasiewicz. Systematic comparison of neural architectures and training approaches for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

M. Honnibal, I. Montani, M. Honnibal, H. Peters, S. V. Landeghem, M. Samsonov, J. Geovedi, J. Regan, G. Orosz, S. L. Kristiansen, P. O. McCann, D. Altinok, Roman, G. Howard, S. Bozek, E. Bot, M. Amery, W. Phatthiyaphaibun, L. U. Vogelsang, B. Böing, P. K. Tippa, jeannefukumaru, GregDubbin, V. Mazaev, R. Balakrishnan, J. D. Møllerhøj, wbwseeker, M. Burton, thomasO, and A. Patel. explosion/spaCy: v2.1.7. In *explosion/spaCy: v2.1.7*, 2019.

P.-L. Huguet Cabot and R. Navigli. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.

P. Jain and Mausam. Knowledge-guided linguistic rewrites for inference rule verification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

P. Jain, P. Kumar, Mausam, and S. Chakrabarti. Type-sensitive knowledge base inference without explicit type supervision. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018.

Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 2020.

J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *arXiv preprint arXiv:1607.01759*, 2016.

L.-A. Kaffee, A. Piscopo, P. Vougiouklis, E. Simperl, L. Carr, and L. Pintscher. A glimpse into babel: An analysis of multilinguality in wikidata. In *Association for Computing Machinery*, 2017.

S. M. Kazemi and D. Poole. Simple embedding for link prediction in knowledge graphs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*, 2021.

T. Khot, A. Sabharwal, and P. Clark. Answering complex questions using open information extraction. In *Association for Computational Linguistics*, 2017.

B. Kim, T. Hong, Y. Ko, and J. Seo. Multi-task learning for knowledge graph completion with pre-trained language models. In *COLING*, 2020.

T. Kim, B. Li, and S.-g. Lee. Multilingual chart-based constituency parse extraction from pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.

P. Koehn et al. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, 2005.

A. Koksal and A. Ozgur. The relx dataset and matching the multilingual blanks for cross-lingual relation classification. *Findings of the Association for Computational Linguistics: EMNLP*, 2020.

K. Kolluru, V. Adlakha, S. Aggarwal, Mausam, and S. Chakrabarti. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020a.

K. Kolluru, S. Aggarwal, V. Rathore, Mausam, and S. Chakrabarti. Imojie: Iterative memory-based joint open information extraction. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020b.

K. Kolluru, M. S. Chauhan, Y. Nandwani, P. Singla, and Mausam. CEAR: Cross-entity aware reranker for knowledge base completion. In *arXiv preprint arXiv:2104.08741*, 2021a.

K. Kolluru, M. Rezk, P. Verga, W. W. Cohen, and P. Talukdar. Multilingual fact linking. *Proceedings of the 3rd Conference on Automated Knowledge Base Construction (AKBC)*, 2021b.

K. Kolluru, M. Muqeeth, S. Mittal, S. Chakrabarti, and Mausam. Alignment-Augmented Consistent Translation for Multilingual Open Information Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022a.

K. Kolluru, G. Stanovsky, and Mausam. "covid vaccine is against covid but oxford vaccine is made at oxford!" semantic interpretation of proper noun compounds. In *Under Review*, 2022b.

J. Krishnamurthy and T. M. Mitchell. Which noun phrases denote which concepts? In *Annual Meeting of the Association for Computational Linguistics*, 2011.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

A. Kunchukuttan, P. Mehta, and P. Bhattacharyya. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

T. Lacroix, N. Usunier, and G. Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, 2018.

M. Lauer. Designing statistical language learners: Experiments on noun compounds. In *Ph.D.thesis, Macquarie University*, 1995.

W. Léchelle, F. Gotti, and P. Langlais. Wire57: A fine-grained benchmark for open information extraction. *arXiv preprint arXiv:1809.08962*, 2018.

P. Liang. Lambda dependency-based compositional semantics. In *ArXiv*, volume abs/1309.4408, 2013.

X. Lin, L. Chen, and C. Zhang. Tenet: Joint entity and relation linking with coherence relaxation. In *SIGMOD '21*, 2021.

G. Liu, X. Li, J. Wang, M. Sun, and P. Li. Extracting knowledge from web text with monte carlo tree search. *Proceedings of The Web Conference 2020*, 2020a.

J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022a.

P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. M. Shazeer. Generating wikipedia by summarizing long sequences. *ArXiv*, abs/1801.10198, 2018.

Q. Liu, D. Yogatama, and P. Blunsom. Relational memory augmented language models. *arXiv preprint arXiv:2201.09680*, 2022b.

Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 2020b.

A. Lopes, R. Nogueira, R. Lotufo, and H. Pedrini. Lite training strategies for Portuguese-English and English-Portuguese translation. In *Proceedings of the Fifth Conference on Machine Translation*, 2020.

P. Maini, K. Kolluru, D. Pruthi, and Mausam. Why and when should you pool? analyzing pooling in recurrent architectures. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, 2020.

J. Mallinson, A. Severyn, E. Malmi, and G. Garrido. Felix: Flexible text editing through tagging and insertion. *arXiv preprint arXiv:2003.10687*, 2020.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2005.

D. Marcheggiani and I. Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *ArXiv*, abs/1703.04826, 2017.

Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

M. Mausam. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

S. V. Mehta, J. Y. Lee, and J. G. Carbonell. Towards semi-supervised learning for deep semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018.

I. A. Mel'cuk et al. *Dependency syntax: theory and practice*. SUNY press, 1988.

J. Michael, G. Stanovsky, L. He, I. Dagan, and L. Zettlemoyer. Crowdsourcing Question-Answer Meaning Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2018, Volume 2 (Short Papers)*, 2018.

F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, 2005.

P. Nakov. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 2013.

G. Nan, R. Qiao, Y. Xiao, J. Liu, S. Leng, H. H. Zhang, and W. Lu. Interventional video grounding with dual contrastive learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2764–2774, 2021.

Y. Nandwani, A. Pathak, P. Singla, and Mausam. A primal dual formulation for deep learning with constraints. In *Advances in Neural Information Processing Systems*, 2019.

T. Nayak and H. T. Ng. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

J. Nivre, M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016.

R. Ormándi, M. Saleh, E. Winter, and V. Rao. Webred: Effective pretraining and finetuning for relation extraction on the web. *ArXiv*, abs/2102.09681, 2021.

M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

L. Pai, W. Gao, W. Dong, S. Huang, and Y. Zhang. Open information extraction from 2007 to 2022 - a survey. In *arXiv preprint arXiv:2208.08690*, 2022.

H. Pal and Mausam. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, 2016.

D. Papadopoulos, N. Papadakis, and N. Matsatsinis. PENELOPIE: Enabling open information extraction for the Greek language through machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, 2021.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

S. Paul, P. Mathur, and S. Kishore. Syntactic construct: An aid for translating english nominal compound into hindi. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, 2010.

F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. D. Cao, J. Thorne, Y. Jernite, V. Plachouras, T. Rocktäschel, and S. Riedel. Kilt: a benchmark for knowledge intensive language tasks. In *arXiv:2009.02252*, 2020.

G. Ponkiya, K. Patel, P. Bhattacharyya, and G. K. Palshikar. Towards a standardized dataset for noun compound interpretation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, 2018.

G. Ponkiya, R. Murthy, P. Bhattacharyya, and G. Palshikar. Looking inside noun compounds: Unsupervised prepositional and free paraphrasing using language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020.

M. Ponza, L. Del Corro, and G. Weikum. Facts that matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, 2018.

A. Pratapa, Z. Liu, K. Hasegawa, L. Li, Y. Yamakawa, S. Zhang, and T. Mitamura. Cross-document event identity via dense annotation. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, 2021.

J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

M. Rahat and A. Talebpour. Parsa: An open information extraction system for persian. *Digital Scholarship in the Humanities*, 2018.

G. Ramesh, S. Doddapaneni, A. Bheemaraj, M. Jobanputra, R. AK, A. Sharma, S. Sahoo, H. Diddee, M. J, D. Kakwani, N. Kumar, A. Pradeep, K. Deepak, V. Raghavan, A. Kunchukuttan, P. Kumar, and M. S. Khapra. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages, 2021.

S. Reddy, D. McCarthy, and S. Manandhar. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, Chiang Mai, Thailand, Nov. 2011. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I11-1024.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

Y. Ren, J. Liu, X. Tan, Z. Zhao, S. Zhao, and T.-Y. Liu. A study of non-autoregressive model for sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

L. F. R. Ribeiro, M. Liu, I. Gurevych, D. Markus, and M. Bansal. Factgraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

Y. Ro, Y. Lee, and P. Kang. Multi^2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

A. Romadhony, A. Purwarianti, and D. H. Widyantoro. Rule-based indonesian open information extraction. In *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018.

A. Rosenbach. Emerging variation: Determiner genitives and noun modifiers in english. *English Language & Linguistics*, 2007.

S. Saha and Mausam. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

S. Saha, H. Pal, and Mausam. Bootstrapping for numerical OpenIE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017.

A. Saxena, A. Tripathi, and P. Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

T. Sellam, D. Das, and A. P. Parikh. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*, 2020.

H. Shah, J. Villmow, and A. Ulges. Relation specific transformations for open world knowledge graph completion. In *TEXTGRAPHS*, 2020.

J. Shaw. Segregatory coordination and ellipsis in text generation. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, 1998.

V. Shwartz, V. Shwartz, and C. Waterson. Olive oil is made *of* olives, baby oil is made *for* babies: Interpreting noun compounds using paraphrases in a neural model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018.

S. Soderland, B. Roof, B. Qin, S. Xu, Mausam, and O. Etzioni. Adapting open information extraction to domain-specific relations. *AI Mag.*, 2010.

G. Stanovsky and I. Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

G. Stanovsky, I. Dagan, et al. Open ie as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015.

G. Stanovsky, J. Ficler, I. Dagan, and Y. Goldberg. Getting more out of syntax with PropS. *CoRR*, abs/1603.01648, 2016.

G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan. Supervised Open Information Extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long Papers)*, 2018.

F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *WWW Conference*, 2007.

D. Sui, Y. Chen, K. Liu, J. Zhao, X. Zeng, and S. Liu. Joint entity and relation extraction with set prediction networks. *arXiv preprint arXiv:2011.01675*, 2020.

M. Sun, X. Li, and P. Li. Logician and orator: Learning from the duality between language and knowledge in open domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018a.

M. Sun, X. Li, X. Wang, M. Fan, Y. Feng, and P. Li. Logician: A unified end-to-end neural approach for open-domain information extraction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018b.

Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

t, Y. Park, T. Lee, and S. Pan. Supervising unsupervised open information extraction models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

H. Teranishi, H. Shindo, and Y. Matsumoto. Coordination boundary identification with similarity and replaceability. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017.

H. Teranishi, H. Shindo, and Y. Matsumoto. Decomposed local models for coordinate structure parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

S. Tratz. *Semantically-enriched parsing for natural language understanding*. University of Southern California, 2011.

T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2016.

V. M. S. Valencia. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam, 1991.

S. Vashishth, P. Jain, and P. Talukdar. Cesi: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference (WWW)*, 2018.

P. Verga, H. Sun, L. B. Soares, and W. Cohen. Adaptable and interpretable neural memoryover symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, and D. Batra. Diverse Beam Search for Improved Description of Complex Scenes. In *AAAI Conference on Artificial Intelligence, 2018*, 2018.

R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, G. Cao, D. Jiang, M. Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

A. S. White, D. Reisinger, K. Sakaguchi, T. Vieira, S. Zhang, R. Rudinger, K. Rawlins, and B. Van Durme. Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

M. Wydmuch, K. Jasinska, M. Kuznetsov, R. Busa-Fekete, and K. Dembczynski. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

M. Xiao and C. Liu. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016.

L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2020.

Z. Yan, D. Tang, N. Duan, S. Liu, W. Wang, D. Jiang, M. Zhou, and Z. Li. Assertion-based QA with question-aware open information extraction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

L. Yao, C. Mao, and Y. Luo. Kg-bert: Bert for knowledge graph completion. *AAAI*, 2019.

M. Yazdani, M. Farahmand, and J. Henderson. Learning semantic composition to detect non-compositionality of multiword expressions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1733–1742, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1201. URL `https://aclanthology.org/D15-1201`.

S. J. Yong, K. Dong, and A. Sun. Docor: Document-level openie with coreference resolution. In *WSDM '23*, 2023.

D. H. Younger. Recognition and parsing of context-free languages in time n3. *Information and control*, 1967.

H. Yu, R. Jiang, B. Zhou, and A. Li. Knowledge-infused pre-trained models for kg completion. In *International Conference on Web Information Systems Engineering*, 2020.

J. Zhan and H. Zhao. Span Model for Open Information Extraction on Accurate Corpus. In *AAAI Conference on Artificial Intelligence, 2020*, 2020.

T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

Z. Zhang, X. Liu, Y. Zhang, Q. Su, X. Sun, and B. He. Pretrain-kge: Learning knowledge representation from pretrained language models. In *Empirical Methods in Natural Language Processing, EMNLP*, 2020a.

Z. Zhang, Z. Zhao, Z. Lin, J. Zhu, and X. He. Counterfactual contrastive learning for weakly-supervised vision-language grounding. In *NeurIPS*, 2020b.

Z. Zhong and D. Chen. A frustratingly easy approach for entity and relation extraction. In *North American Association for Computational Linguistics (NAACL)*, 2021.

List of Included Papers

This thesis is based on the following publications:

1. "IMoJIE: Iterative Memory-Based Joint Open Information Extraction". Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam and Soumen Chakrabarti. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online, July 2020.

2. "OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction". Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam and Soumen Chakrabarti. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online, Nov 2020.

3. "Multilingual Fact Linking". Keshav Kolluru, Martin Rezk, Pat Verga, William W. Cohen, and Partha Talukdar. Proceedings of the 3rd Conference on Automated Knowledge Base Construction (AKBC). Online, October 2021.

4. "Cross Entity Aware Reranking for Open Knowledge Base Completion". Keshav Kolluru, Mayank Kumar, Yatin Nandwani, Parag Singla and Mausam. Arxiv, 2021.

5. "Alignment-Augmented Consistent Translation for Multilingual Open Information Extraction". Keshav Kolluru, Muqeeth Mohammad, Shubham Mittal, Mausam and Soumen Chakrabarti. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Online, July 2022.

6. "Covid vaccine is against Covid but Oxford vaccine is made at Oxford!" Semantic Interpretation of Proper Noun Compounds. Keshav Kolluru, Gabriel Stanovsky, and Mausam. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP). Abu Dhabi, December 2022.

Biography

Keshav Sai Kolluru hails from Visakhapatnam, Andhra Pradesh. He completed his graduation from Computer Science Deparment at IIT Bhubaneswar in 2017 and joined the PhD program of Computer Science Department at IIT Delhi in 2017. He has published several research papers in top tier venues such as EMNLP and ACL and has served as a reviewer for these conferences as well. He has successfully completed multiple internships at KnowDis Data Science, Google Research, IBM Research, and Microsoft IDC.