

Neural Architectures and Evaluation Protocols for Open Information Extraction

Thesis submitted by

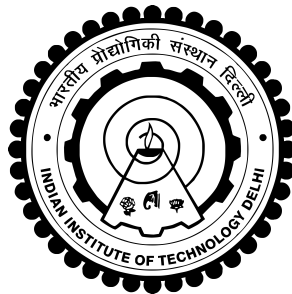
Samarth Aggarwal
2016CS10395

under the guidance of

Prof. Mausam and Prof. Soumen Chakrabarti

*in partial fulfilment of the requirements
for the award of the degree of*

Bachelor of Technology



Department Of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY DELHI

July 2020

THESIS CERTIFICATE

This is to certify that the thesis titled **Neural Architectures and Evaluation Protocols for Open Information Extraction**, submitted by **Samarth Aggarwal**, to the Indian Institute of Technology, Delhi, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Mausam

Professor

Dept. of Computer Science

IIT-Delhi, 110016

Prof. Soumen Chakrabarti

Professor

Dept. of Computer Science

IIT-Bombay, 400076

Place: New Delhi

Date: 10th July 2020

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisors, Prof. Mausam and Prof. Soumen Chakrabarti for their constant encouragement throughout this project, for their patience, motivation, enthusiasm and immense knowledge. They consistently allowed me to frame my own work, but steered me in the right direction whenever I needed it. I could not have imagined better advisors and mentors for my research.

I would also like to thank Keshav Kolluru for being an excellent collaborator. He brought structure to both my research and thought process allowing me to grow into this field of study. I am also obliged to Vipul Rathore for his help on a short notice during paper submissions.

I would pay my special regards to Yatin Nandwani, Prachi Jain, Lovish Madaan and Sushant Rath for their insightful discussions, advices and quick work-arounds that proved to be very helpful.

I thank IIT Delhi HPC Facility for compute resources.

Lastly, I would like to dedicate this work to my beloved parents and my brother Yatharth for their love and constant support throughout my life.

ABSTRACT

Open Information Extraction refers to the task of obtaining relation tuples from a sentence. For eg. the sentence ‘‘Donald Trump is the president of United States.’’ yields (Donald Trump ; is the president of ; United States) as its OpenIE tuple.

The Open IE paradigm is a useful intermediary for a variety of down-stream tasks such as sentence similarity, event schema induction, text comprehension, knowledge base completion, and more. There have been several attempts at building OpenIE systems that explored rule-based such as ClausIE, OpenIE-4 and OpenIE-5. Another wave of OpenIE systems that followed, comprised of neural approaches such as RnnOIE and Cui et al. (2018a). However, the rule-based systems suffered from cascading errors from a large number of components in succession. The existing neural OpenIE systems, although have improved, yet are far from ideal. Infact, they introduced other problems such as redundancy in their outputs. Together these factors solicit an OpenIE system that is able to overcome the issues pertaining to OpenIE.

Human inspection revealed that the existing systems were not upto the mark, yet these systems scored high on the existing OpenIE benchmarks such as OIE2016 (Stanovsky and Dagan, 2016a). This meant that the existing benchmarks do not correlate well with how humans evaluate OpenIE. In response, we contribute CaRB (Bhardwaj et al., 2019), with a high-quality crowdsourced gold dataset and intuitive evaluation policies that correlate well with human judgement of OpenIE. CaRB establishes itself as the new state of the art OpenIE benchmark.

CaRB evaluation of the Cui et al. (2018a), then state of the art OpenIE systems, confirms its inept performance. We contribute IMoJIE (Kolluru et al., 2020), a neural OpenIE model that outperforms the previous state of the art by about 18 F1 points. It reduces the redundancy in output extractions significantly. Along with it, IMoJIE also presents a novel approach that can be used to generation high-quality training data from multiple low quality datasets.

Although IMoJIE improves the quality of OpenIE tuples significantly, this improvement comes at the cost of speed of extraction. We design a MLIL architecture to overcome the issue of speed of extraction and also obtain further performance nudges from it. The MLIL approach also yields a coordination analyzer that establishes itself as a state-of-the-art in the task and also increases the yield of the MLIL based OpenIE to 1.7x.

We also analyse OpenIE milestones and contribute some ideas for future research.

Contents

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF TABLES	vii
LIST OF FIGURES	1
ABBREVIATIONS	2
NOTATION	3
1 Introduction	1
1.1 Overview	1
1.1.1 What is OpenIE	1
1.1.2 Performance Gaps	2
1.2 Problem Statement	3
1.3 Contributions	3
2 Literature Survey	5
2.1 Existing OIE Systems	5
2.1.1 Rule-Based Systems	5
2.1.2 Neural Systems	5
2.2 Benckmarks	7
2.2.1 OIE2016	7
2.2.2 RelVis	8
2.2.3 Wire57	8
3 CaRB - A Crowdsourced Benchmark for Open IE	9
3.1 Overview	9

3.1.1	Need for a New Benchmark	9
3.1.2	Establishing a new State-of-the-Art	9
3.2	Crowdsourcing CaRB Dataset	10
3.3	The CaRB Scorer	11
3.4	Evaluation	12
3.4.1	Dataset Quality	12
3.4.2	Comparison of Open IE Systems	13
3.4.3	Human Verification	15
3.5	Conclusion	15
4	IMoJIE - Iterative Memory Based Joint Open IE	16
4.1	Overview	16
4.2	Sequential Decoding Model	16
4.3	Aggregating Bootstrapped Data	18
4.3.1	Single Bootstrapping System	18
4.3.2	Multiple Bootstrapping Systems	18
4.4	Experimental Setup	20
4.4.1	Training Data Construction	20
4.4.2	Dataset and Evaluation Metrics	20
4.4.3	Comparison Systems	21
4.4.4	Implementation	22
4.5	Results and Analysis	22
4.5.1	Performance of Existing Systems	22
4.5.2	Performance of IMoJIE	22
4.5.3	Redundancy	24
4.5.4	The Value of Iterative Memory	26
4.5.5	The value of Score-and-Filter	26
4.6	Error Analysis	28
4.7	Discussion	28
5	Conjunction Splitting	30
5.1	Problems Remaining in OpenIE	30

5.1.1	Missing Information	30
5.1.2	Incorrect Conjunction Splitting	30
5.1.3	Grammatical Errors	31
5.1.4	Missing Output Context	31
5.1.5	Coreference Resolution	32
5.1.6	Incorrect Identification of Boundaries	32
5.2	Why to Split Conjunctions?	33
5.2.1	Analysis of Existing Systems	33
5.3	HC-Tree Based Splitting	33
5.3.1	Properties of Conjuncts	34
5.3.2	Proposed Approach	35
5.3.3	Drawbacks	37
5.4	Patterns dictating Distributivity of Coordination Phrase	38
5.5	Analysis of Non-Distributive Sentences	39
6	MLIL - Multi Level Iterative Labelling	42
6.1	MLIL Based OpenIE	42
6.1.1	Major Drawback of IMOJIE: Speed	42
6.1.2	OpenIE as 2-D Grid Labelling	43
6.2	Coordination Analyzer	43
7	OpenIE Milestones and Future Ideas	45
7.1	OpenIE Milestones	45
7.2	Future Ideas	45
A	A SAMPLE APPENDIX	47
A.1	Performance of IMOJIE with varying sentence lengths	47
A.2	Measuring Performance of IMOJIE on Varying Beam Size	47
A.3	Evaluation of IMOJIE on other datasets	48
A.4	Visualizing Attention	49

List of Tables

1.1	Example of Open IE tuples of some sample sentences	1
3.1	One-to-One Match vs. Multi Match	11
3.2	Tuple Match vs. Lexical Match	12
3.3	Data quality using token-level match	12
3.4	Data quality using lexical match	12
3.5	Sample gold annotations for OIE2016 vs. CaRB	13
3.6	Performance of Open IE systems on CaRB	13
4.1	Comparison of various OpenIE systems - non-neural, neural and proposed models. (*) Cannot compute AUC as Sense-OIE, MinIE do not emit confidence values for extractions and released code for Span-OIE does not provision calculation of confidence values. In these cases, we report the Last F1 as the Opt. F1	21
4.2	IMoJIE vs. OpenIE-4. Pipeline nature of OpenIE-4 can get confused by long convoluted sentences, but IMoJIE responds gracefully.	23
4.3	IMoJIE vs. CopyAttention. CopyAttention suffers from stuttering, which IMoJIE does not.	23
4.4	Models to solve the redundancy issue prevalent in Generative Neural OpenIE systems. All systems are bootstrapped on OpenIE-4.	24
4.5	Evaluating models trained with different bootstrapping systems.	25
4.6	Measuring redundancy of extractions. MNO stands for Mean Number of Occurrences. IOU stands for Intersection over Union.	26
4.7	Performance of IMoJIE on aggregated dataset OpenIE-4+ClausIE+RnnOIE , with different filtering techniques. For comparison, SenseOIE trained on multiple system extractions gives an F1 of 17.2 on CaRB.	26
4.8	IMoJIE trained with different combinations of bootstrapping data from 3 systems - OpenIE-4, ClausIE, RNNOIE. Graph filtering is not used over single datasets.	27
5.1	Conjunction Splitting in Existing Rule-Based Systems	33
5.2	Reasons for non-distributivity of sentences along with their significance based on the frequency of occurrence	39

6.1	Evaluation of OpenIE. Using constrained learning, IMoJIE(CL) offers a 60x speedup and better performance scores on all metrics compared to IMoJIE. Adding a coordination analyzer, IMoJIE(CL) + IMoJIE-CA gives the best scores in 3 of the 4 metrics. MinIE, SenseOIE, SpanOIE do not output confidences. Code of SenseOIE is not available to compute speed.	42
6.2	Manual comparison of Precision and Yield on 100 random conjunctive sentences from CaRB Gold.	44
A.1	Evaluation on other datasets with the CaRB evaluation strategy	48

List of Figures

3.1	Comparison of Open IE systems using OIE2016	14
3.2	Evaluation of Open IE systems using CaRB	14
4.1	One step of the sequential decoding process, for generating the i^{th} extraction, which takes the original sentence and all extractions numbered $1, \dots, i - 1$, previously generated, as input.	17
4.2	Ranking-Filtering subsystem for combining extractions from multiple open IE systems in an unsupervised fashion. (‘Exts’=extractions.)	18
4.3	Precision-Recall curve of OpenIE Systems.	24
5.1	Example of a sentence and its Hierarchical Coordination (HC) Tree	34
6.1	The extractions (<i>Rome; [is] the capital of; Italy</i>) and (<i>Rome; is known for; it’s rich history</i>) can be seen as the output of the grid labeling. We additionally introduce a token <i>[is]</i> to the input.	43
A.1	Measuring performance with varying input sentence lengths	47
A.2	Measuring performance of CopyAttention with BERT model upon changing the beam size	48
A.3	BERT attention for the word ‘founding’	50
A.4	BERT attention for the word ‘justice’	51
A.5	BERT attention for the word ‘prime’	52
A.6	BERT attention for the word ‘minister’	53
A.7	Attention weights for the decoder	54

ABBREVIATIONS

IITD	Indian Institute of Technology, Delhi
RTFM	Read the Fine Manual

NOTATION

r	Radius, m
α	Angle of thesis in degrees
β	Flight path in degrees

Chapter 1

Introduction

1.1 Overview

1.1.1 What is OpenIE

Extracting structured information from unstructured text has been a key research area within NLP. The paradigm of Open Information Extraction (OpenIE) Banko et al. (2007a) uses an open vocabulary to convert natural text to semi-structured representations, by extracting a set of (subject, relation, object) tuples. OpenIE has found wide use in many downstream NLP tasks Mausam (2016a) like multi-document question answering and summarization Fan et al. (2019), event schema induction Balasubramanian et al. (2013) and word embedding generation Stanovsky et al. (2015b). Table 1.1 enlists a few examples of sentences along with their OpenIE tuples.

Sentence	Open IE Tuples
The US President Donald Trump gave his speech on Tuesday to thousands of people.	(Donald Trump ; is the president of ; US) (Donald Trump ; gave ; his speech) (Donald Trump ; gave his speech on ; Tuesday) (Donald Trump ; gave his speech to ; thousands of people)
John likes to play the piano.	(John ; likes to play ; the piano)
Solo Piano I is a great album of classical piano compositions.	(Solo Piano I ; is a great album of ; classical piano compositions)
John said, "Monday is the first day of the week."	(John ; said ; "Monday is the first day of the week") ([Context : John said] Monday ; is ; the first day of the week)

Table 1.1: Example of Open IE tuples of some sample sentences

The task of Open Information Extraction (OpenIE) involved listing out all possible inferences from a given sentence. Each OpenIE extraction typically comprises of a relation and two arguments. However, this structure is not strictly enforced and one or more of these components may be absent from a valid OpenIE tuple. Multiple formats have been proposed for output of OpenIE:

1. N-ary Format : In this format, all the arguments corresponding to the same relation and (subject)argument are stacked within a single extraction, albeit the argument boundary is maintained. For eg.

Sentence:

“The US president Donald Trump gave his speech on Tuesday to thousands of people.”

Extractions:

(Donald Trump ; gave ; his speech ; on Tuesday ; to thousands of people)

2. Binary Format : In this format, all the arguments corresponding to the same relation and (subject)argument are assigned to separate tuples. For eg.

Sentence:

“The US president Donald Trump gave his speech on Tuesday to thousands of people.”

Extractions:

(Donald Trump ; gave ; his speech)

(Donald Trump ; gave his speech on ; Tuesday)

(Donald Trump ; gave ; his speech to ; thousands of people)

Notice that the N-ary format keeps the prepositions preceding along with the respective arguments whereas the binary format moves them along with the relation. Although, the inter-conversion between the two formats is easy, we will focus on the binary format for the rest of the thesis due to its relatively higher popularity among the recent systems.

1.1.2 Performance Gaps

There have been many Open IE systems till date such as TextRunner (Banko et al., 2007b), ReVerb (Fader et al., 2011a; Etzioni et al., 2011a), OLLIE (Mausam et al., 2012a), ClausIE (Del Corro and Gemulla, 2013a), OpenIE 4 (Christensen et al., 2011a; Pal and Mausam, 2016), OpenIE 5 (Saha et al., 2017a; Saha and Mausam, 2018), PropS (Stanovsky et al., 2016a), NST (Jia et al., 2018), Neural Open IE (Cui et al., 2018a), and more. With the advent of so many systems, it is imperative to have a standardized mechanism for automatic evaluation so that they can be compared. This led to Open IE benchmarking systems such as RelVis (Schneider et al., 2017), Wire57 (Léchelle et al., 2018) and OIE2016 (Stanovsky and Dagan, 2016a).

However, OpenIE systems still perform unsatisfactorily when compared against the gold OpenIE outputs. The traditional rule-based OpenIE systems have a large number of components that cause errors to be cascaded. On the other hand, the neural systems introduce other problem such as high degree of redundancy among extractions, fixed number of extractions due to a beam search, and more. These problems indicate a need for an improved OpenIE system that can cater most, if not all, of these problems.

Upon manually examining the OpenIE systems and their evaluation, we figured out an even deeper rooted problem. The benchmarks used to evaluate these systems were

themselves imperfect. They suffer from two major flaws: small and noisy gold datasets, and non-standard evaluation policies. Wire57 (Léchelle et al., 2018) uses a tiny gold dataset of only 57 sentences. RelVis had non-intuitive evaluation policies. OIE2016, which was the state-of-the-art OpenIE benchmark at that time, also had a noisy gold dataset (refer to table 3.5) as well as non-intuitive schemes of evaluation.

1.2 Problem Statement

It was clear that creation of a reliable OpenIE benchmark had to precede the creation of a better OpenIE system. Hence, we were faced with the following two problem statements:

Problem 1: Establish a new state-of-the-art in OpenIE benchmarking, with a large high-quality gold dataset and evaluation policies that correlate well with human judgement.

Problem 2: Develop a new state-of-the-art OpenIE system to overcome the problems prevailing with existing systems.

The motivation to improve OpenIE came from the benefits that would percolate to its downstream applications. Open IE has numerous downstream applications such as knowledge base construction, relation extraction, summarisation and learning word embeddings (Stanovsky et al., 2015a; Mausam, 2016b). Improving the performance of OpenIE systems would directly translate to an enhancement in their performance as well.

1.3 Contributions

Our major contributions are:

- We contribute CaRB, an improved dataset and framework for testing Open IE systems. To the best of our knowledge, CaRB is the first *crowdsourced* Open IE dataset and it also makes substantive changes in the matching code and metrics. NLP experts annotate CaRB’s dataset to be more accurate than OIE2016. Moreover, we find that on one pair of Open IE systems, CaRB framework provides contradictory results to OIE2016. Human assessment verifies that CaRB’s ranking of the two systems is the accurate ranking. We release the CaRB framework along with its crowdsourced dataset.
- We contribute IMojIE, a neural OpenIE system that generates the next extraction, fully conditioned on the extractions produced so far. IMojIE produce a variable number of diverse extractions for a sentence,
- We present an unsupervised aggregation scheme to bootstrap training data by combining extractions from multiple OpenIE systems.

- We present a new state-of-the-art Coordination Analyzer which is 13 pts better in F1 over previous best reported results
- We contribute a Multi Level Iterative Labelling (MLIL) architecture based OpenIE system that is trained using constraints to improve recall. MLIL based OpenIE establishes itself as a new state-of-the-art in OpenIE systems. It is 25x faster than IMOJIE and improves the F1 score by as much as 3.8 pts.

Chapter 2

Literature Survey

2.1 Existing OIE Systems

The existing OpenIE systems can be broadly categorised into two categories - Rule-Based Systems, and Neural Systems.

2.1.1 Rule-Based Systems

Traditional open extractors are rule-based or statistical, e.g., Textrunner Banko et al. (2007a), ReVerb Fader et al. (2011b); Etzioni et al. (2011b), OLLIE Mausam et al. (2012b), Stanford-IE Angeli et al. (2015), ClausIE Del Corro and Gemulla (2013b), OpenIE-4 Christensen et al. (2011b); Pal and Mausam (2016), OpenIE-5 Saha et al. (2017b, 2018), PropS Stanovsky et al. (2016b), and MinIE Gashteovski et al. (2017). These systems are largely unsupervised in nature, or bootstrapped from extractions made by earlier systems. They use syntactic or semantic parsers combined with rules to extract tuples from sentences.

2.1.2 Neural Systems

To bypass error accumulation in rule-based systems with multiple subcomponents, end-to-end neural systems have been proposed recently. They belong to one of two paradigms: sequence *labeling* or sequence *generation*.

Sequence Labelling

Sequence Labeling involves tagging each word in the input sentence as belonging to the subject, predicate, object or other. The final extraction is obtained by collecting labeled spans into different fields and constructing a tuple. RnnOIE (Stanovsky et al., 2018a) is a labeling system that first identifies the relation words and then uses sequence labelling to get their arguments. It is trained on OIE2016 dataset, which postprocesses SRL data for OpenIE (Stanovsky and Dagan, 2016b).

SenseOIE (Roy et al., 2019), improves upon RnnOIE by using the extractions of multiple OpenIE systems as features in a sequence labeling setting. However, their training requires

manually annotated gold extractions, which is not scalable for the task. This restricts SenseOIE to train on a dataset of 3,000 sentences. In contrast, our proposed *Score-and-Filter* mechanism is unsupervised and can scale unboundedly. Jiang et al. (2019) is another labeling system that better calibrates extractions across sentences.

SpanOIE (Zhan and Zhao, 2020) uses a span selection model, a variant of the sequence labelling paradigm. Firstly, the predicate module finds the predicate spans in a sentence. Subsequently, the argument module outputs the arguments for this predicate. However, SpanOIE cannot extract nominal relations. Moreover, it bootstraps its training data over a single OpenIE system only.

Sequence Generation

Sequence Generation uses a Seq2Seq model to generate output extractions one word at a time. The generated sequence contains field demarcators, which are used to convert the generated flat sequence to a tuple.

Copy Attention (Cui et al., 2018a) is a neural generator trained over bootstrapped data generated from OpenIE-4 extractions on a large corpus. During inference, it uses beam search to get the predicted extractions. It uses a fixed-size beam, limiting it to output a constant number of extractions per sentence. Moreover, our analysis shows that CopyAttention extractions severely lack in diversity, as illustrated in Table 4.3.

Our analysis of Copy Attention 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.

These limitations arise because its decoder has no explicit mechanism to remember what parts of the sentence have already been ‘consumed’ or what triples have already been generated. Its decoder uses a fixed-size beam for inference. However, beam search can only ensure that the extractions are not exact duplicates.

Sun et al. (2018) propose the *Logician* model, a restricted sequence generation model for extracting tuples from Chinese text. Logician relies on coverage attention and gated-dependency attention, a language-specific heuristic for Chinese. Using coverage attention, the model also tackles generation of multiple extractions while being globally-aware. We compare against Logician’s coverage attention as one of the approaches for increasing diversity.

Comparison

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 always be ideal.

On the other hand, sequence-generation models subsume the former type of models. It 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.

2.2 Benckmarks

To the best of our knowledge, there were three benchmarks systems available for comparing Open IE systems - OIE2016, RelVis and Wire57.

2.2.1 OIE2016

The first and the most prominent is OIE2016 Stanovsky and Dagan (2016a). This has been widely adopted as the standard evaluation framework to test new systems on (e.g., OIE2016 is used by the NST (Jia et al., 2018) and Neural Open IE (Cui et al., 2018a) systems). In OIE2016, gold tuples are generated using an automated rule-based system built on top of a QA-SRL dataset He et al. (2015). In early analysis we find this dataset to be rather noisy. Table 3.5 illustrates some sample sentences from this gold dateset. These tuples look obviously wrong, and unfit to be in the gold set.

In addition to the dataset, Stanovsky and Dagan (2016a) release a scorer that compares a set of gold tuples with a set of system tuples to estimate word-level precision and recall. This scorer has been identified to not penalize long extractions. It also does not penalise extractions for misidentifying parts of a relation in an argument slot (or vice versa), leading to trivial systems that score much better than genuine Open IE systems L  chelle et al. (2018). We also observe that the scorer compares words all-to-all allowing multiple same words in an extraction to match a corresponding one in the gold. Thus, simply repeating a word in the extraction will give it a high precision score. Finally, the scorer loops over gold tuples in an arbitrary order, and matches them to predicted extractions in a sequential manner. Once a gold matches to a predicted extraction, it is rendered unavailable for any subsequent, potentially better-matched, extraction.

2.2.2 RelVis

Another dataset is RelVis Schneider et al. (2017), a benchmark that borrows its data from four different datasets including OIE2016. Since OIE2016 forms a major part of this dataset, it has similar issues with noise. Its scorer makes some modifications to OIE2016. However, it does not reward partial coverage of gold tuples, and forces one system prediction to match just one gold. It also does not penalize overlong extractions.

2.2.3 Wire57

Finally, Wire57 L  chelle et al. (2018) makes further improvements in the scorer. It penalises overlong extractions and assigns a token-level precision and recall score to all gold-prediction pairs for a sentence. Moreover, it considers all pairs of extractions in its matching phase. However, it still forces one prediction to match just one gold. It also reports just one score for a system, ignoring the confidence values of the individual predictions that make the precision-recall curve of OIE2016 possible. Our scorer is inspired by theirs, with some changes. More importantly, the dataset used in Wire57 is manually curated, but with only 57 sentences, which is too small to suffice as a comprehensive test dataset.

Chapter 3

CaRB - A Crowdsourced Benchmark for Open IE

3.1 Overview

3.1.1 Need for a New Benchmark

Traditionally, these systems have been evaluated over small manually curated gold datasets (e.g., Fader et al. (2011a); Mausam et al. (2012a)). There are two problems with this approach. One, it is not reliable due to the small size of annotation. Second, it lacks standardization, since there is no single gold dataset over which all systems are evaluated. Moreover, the guidelines to annotate may vary across datasets and annotators. Recently, some standard benchmarks datasets and evaluators have been proposed: OIE2016 (Stanovsky and Dagan, 2016a), RelVis (Schneider et al., 2017), and Wire57 (Léchelle et al., 2018). Unfortunately, these datasets are either too small or too noisy to meaningfully compare Open IE systems.

3.1.2 Establishing a new State-of-the-Art

In response, we propose a new benchmark system CaRB: **C**rowdsourced **a**utomatic open **R**elation extraction **B**enchmark (Bhardwaj et al., 2019), which has a good sized and high quality dataset, along with better evaluation metrics. In order to create this gold dataset, we crowdsource human annotation of extractions using Amazon Mechanical Turk (MTurk) using the same original sentences as OIE2016. Our MTurk task has an automated system for training and qualifying workers, which makes crowdsourcing this annotation feasible.

Two Open IE experts (authors of this paper) manually annotate 50 random sentences, which are then used as expert ground truth to evaluate the respective tuples in OIE2016’s and CaRB’s gold datasets (Tables 3.3,3.4). We find that CaRB outperforms OIE2016 by 21 points in precision and 16 points in recall in token level match. This demonstrates that CaRB’s gold dataset is significantly more accurate than OIE2016’s. Additionally, when evaluating all systems using our benchmark, we notice that CaRB reverses OIE2016’s ranking of PropS and ClausIE. Human verification, again through crowdsourcing, verifies that two systems are ranked more accurately by CaRB. We release CaRB’s dataset, along with its evaluator as a novel benchmark for further use by research community.¹

¹<https://github.com/dair-iitd/CaRB>

3.2 Crowdsourcing CaRB Dataset

To overcome the shortcomings of dataset noise and size, we crowdsource a high-quality gold dataset for Open IE. We ask workers over Amazon Mechanical Turk (MTurk) to annotate extractions for the 1,282 sentences in dev and test splits of OIE2016. The workers annotate tuples in the form (arg1, rel, arg2), and also annotate location and time attributes for each tuple, when possible.

Open IE annotations are not easy to obtain from non-expert workers. To get acceptable quality, we train workers using a tutorial that doubles up as a qualification test. Their performance in the test is automatically graded. Only workers that pass this are allowed to move on to the main task. The qualification is integrated with the task so that a new worker is served the tutorial and test first, but a qualified worker is directly taken to the main task. This makes the crowdsourcing process scalable.

We divide the task of annotating a sentence into three steps: (1) identifying the relation, (2) identifying the arguments for that relation, and (3) optionally identifying the location and time attributes for the tuple. The training process for the annotators is split into four steps, each of which focuses on a different guideline for Open IE. These are:

1. **Completeness:** The worker must attempt to extract *all* assertions from the sentence.
2. **Assertedness:** Each tuple must be implied by the original sentence.
3. **Informativeness:** The worker must include the maximum amount of relevant information in an argument.
4. **Atomicity:** Each tuple must be an indivisible unit. Whenever possible, the worker must extract multiple atomic tuples from a sentence that has conjunctions.

We also develop a user-friendly interface for annotating the sentences, which almost eliminates the need for workers to type anything. However, we note that several workers got frustrated in our qualification test, could not understand the task and left the job. However, several good workers completed the task successfully, and annotated significant high-quality data for us.

For sentences involving reporting verbs like *said*, *told*, *asked*, etc., some systems annotate additional attributional context for every utterance Mausam et al. (2012a). For this, we create a separate task, so as to prevent workers from being bombarded with all the rules at the same time.

We post-process the data to remove obvious incorrect annotations, like ones with a missing arg1 or rel. We also follow the convention of ending a relation with a preposition instead of beginning arg2 with one, so all prepositions are shifted to rel.

3.3 The CaRB Scorer

We now describe CaRB’s approach for scoring system predictions against the gold. Instead of greedily matching gold tuples to system tuples in arbitrary order, CaRB creates an all-pair matching table, with each column as system tuple and each row as gold tuple. It computes precision and recall scores between each pair of tuples. Then, for computing overall recall, the maximum recall score is taken in each row, and averaged. By taking the maximum, recall computation matches a gold tuple with the closest system extraction. For computing precision, the system predictions are matched one to one with gold tuples, in the order of best match score to worst. The match precision scores are then averaged to compute precision. To compute precision-recall curve this computation is done at different confidence thresholds of system extractions.

Sentence	<i>I ate an apple and an orange.</i>	(prec,rec)	
Gold	(I; ate; an apple) (I; ate; an orange)	OIE2016	CaRB
System 1	(I; ate; an apple and an orange)	(1,0.5)	(0.57,1)
System 2	(I; ate; an apple)	(1,0.5)	(1,0.87)

Table 3.1: One-to-One Match vs. Multi Match

In this way, CaRB’s recall computation uses the notion of *multi-match*, wherein a gold tuple can match multiple system extractions. This is helpful in avoiding penalizing a system very heavily if it stuffs information from multiple gold tuples in a single extraction. Table 3.1 displays an example wherein system 1 combines information from two gold tuples in a single extraction, and system 2 only extracts one of the gold tuples. One-to-one match (OIE2016) is indifferent between the two which means that for OIE2016, adding more information in the same extraction has no value at all. However, multi match (CaRB) assigns higher recall to system 1, since it contains strictly more information, and higher precision to system 2, since its prediction exactly matched a gold extraction.

On the other hand, CaRB uses *single match* for precision. This is because CaRB’s gold tuples are atomic, and cannot be further divided into more tuples. By single matching for precision, CaRB penalizes Open IE systems that produce several very similar and redundant extractions.

Another significant change from OIE2016 scorer is in the use of *tuple match* instead of *lexical match*. CaRB matches relation with relation, and arguments with arguments, however OIE2016 serialized the tuples into a sentence and just computed lexical matches.

Table 3.2 illustrates an example when the arguments are shuffled, lexical match (OIE2016) shows no effect but tuple match (CaRB) rightfully decreases the scores. To avoid spurious

Sentence	<i>I ate an apple.</i>	(prec,rec)	
Gold	(I; ate; an apple)	OIE2016	CaRB
System 1	(I; ate; an apple)	(1,1)	(1,1)
System 2	(ate; an apple; I)	(1,1)	(0,0)

Table 3.2: Tuple Match vs. Lexical Match

matches, CaRB considers only matches with atleast one common word in the relation field.

Finally, some Open IE systems extract n-ary tuples and others do not. To treat all systems on equal footing, we follow previous work and append all higher numbered arguments into arg2.

3.4 Evaluation

3.4.1 Dataset Quality

We first estimate the overall quality of the crowdsourced dataset. To this end, two authors of this paper annotate 50 dev sentences from OIE2016 to create an expert dataset. They first independently annotate tuples from these sentences, achieving an agreement F1 score of 83. They then resolve the differences and merge these independent sets. This is taken as an expert gold against which both OIE2016 and CaRB datasets are assessed.

Dataset	Precision	Recall	F1
OIE2016	0.65	0.55	0.60
CaRB	0.87	0.71	0.78

Table 3.3: Data quality using token-level match

	Precision	Recall	F1
OIE2016	0.67	0.51	0.57
CaRB	0.74	0.73	0.73

Table 3.4: Data quality using lexical match

Tables 3.3 and 3.4 estimate dataset quality of OIE2016 and CaRB. We find that CaRB has enormously high precision and recall values, suggesting that it is a much cleaner dataset. Table 3.5 compares the crowd sourced annotations and OIE2016 gold annotations for some sample sentences. While there is still scope for improvement, CaRB dataset appears much better than the OIE2016’s gold.

Stanovsky and Dagan (2016a) remark that their gold dataset reaches an F1 of 95.8 on their expert annotation, whereas our assessment suggest values around 60. We surmise that this discrepancy is due to the different gold-prediction scoring schemes used. In original OIE2016 paper, the authors “match an automated extraction with a gold proposition if

Sent. # 1	<i>Butters Drive in the Canberra suburb of Phillip is named in his honour .</i>
OIE2016	(in the Canberra suburb of Phillip is named in his honour . ; drive;), (Butters Drive in the Canberra suburb of Phillip ; named ; his honour)
CaRB	(Butters Drive in the Canberra suburb of Phillip ; is named ; in his honour), (Butters Drive ; is ; in the Canberra suburb of Phillip)
Sent. # 2	<i>It was only incidentally that economic issues appeared in nationalist political forms .</i>
OIE2016	(incidentally ; appeared ; economic issues ; nationalist political forms .)
CaRB	(economic issues ; appeared only incidentally in ; nationalist political forms)
Sent. # 3	<i>The main reason for this adoption over mainline gimp was its support for high bit depths which can be required for film work .</i>
OIE2016	(high bit depths ; required ; film work)
CaRB	(this adoption ; has support for ; high bit depths), (high bit depths ; can be required for ; film work), (this adoption ; was over ; mainline gimp), (mainline gimp ; has no support for ; high bit depths), (its support for high bit depths which can be required for film work ; was The main reason for ; this adoption over mainline gimp)
Sent. # 4	<i>The number of ones equals the number of zeros plus one , since the state containing only zeros can not occur .</i>
OIE2016	(The number of ones ; equals ; the number of zeros plus one ; since the state containing only zeros can not occur), (the state ; containing ; only zeros), (the state containing only zeros ; occur)
CaRB	(The number of ones ; equals ; the number of zeros plus one), (the state containing only zeros ; can not occur)

Table 3.5: Sample gold annotations for OIE2016 vs. CaRB

both agree on the grammatical head of all of their elements (predicate and arguments)".² The head match criterion is a much laxer scheme than ours and can explain the very high F1 score against their expert annotation.

3.4.2 Comparison of Open IE Systems

System	Precision	Recall	F1	AUC
Ollie	0.505	0.346	0.411	0.224
PropS	0.340	0.300	0.319	0.126
OpenIE 4	0.553	0.437	0.488	0.272
OpenIE 5	0.521	0.424	0.467	0.245
ClausIE	0.411	0.496	0.450	0.224

Table 3.6: Performance of Open IE systems on CaRB

²This scheme is later changed in their github repository to a lexical match, where if the fraction of words in the prediction also present in the gold is above a threshold, the pair is declared a match.

We test the different Open IE systems depicted in Stanovsky and Dagan (2016a), using the CaRB dataset and scorer. The p-r curves obtained using OIE2016 and CaRB are outlined in figures 3.1 (reproduced from Stanovsky et al. (2018b)) and 3.2. Precision, recall and F1 scores (at max F1 point) and area under precision-recall curve are reported in Table 3.6. It can be seen that the curve for PropS lies above ClausIE at all times in OIE2016, but PropS performs the worse of all systems in CaRB. To verify that CaRB indeed gives the correct ranking, we turn back to human verification.

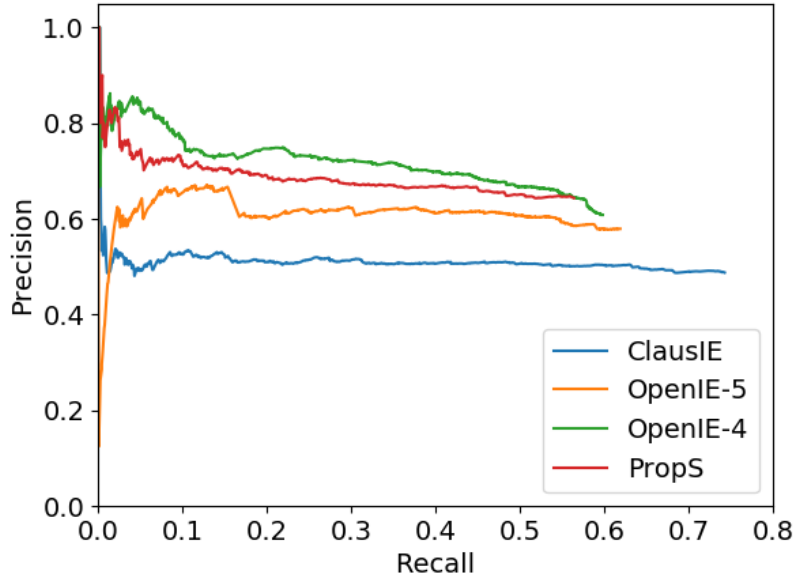


Figure 3.1: Comparison of Open IE systems using OIE2016

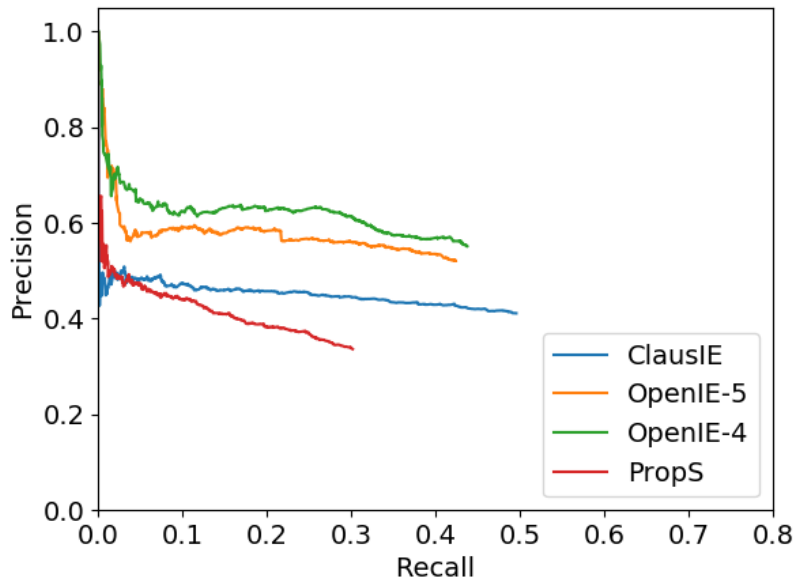


Figure 3.2: Evaluation of Open IE systems using CaRB

3.4.3 Human Verification

Through human verification, our goal is to learn the accurate ranking for ClausIE and PropS. We randomly select 100 test sentences and evaluate both system extractions on this subset.

We assess the correct ranking between PropS and ClausIE using MTurk. Four workers are shown the extractions from both systems in random order and asked to either choose one of the systems as the better one or indicate that both are equal. The majority opinion of these four is considered as the correct ranking for that sentence, an equal split leading to a tie. In this experiment, we only allow MTurk workers who have been trained for Open IE for the crowdsourcing task to participate.

Of these 100 sentences, PropS is chosen to have performed better for 15, ClausIE for 69 whereas 16 ended up in a tie. ClausIE is indeed considered the better system in human evaluation, and we verify that CaRB gives an accurate ranking of these two systems compared to OIE2016.

3.5 Conclusion

We contribute CaRB (Bhardwaj et al., 2019), a crowdsourced dataset for evaluation and comparison of Open IE systems. We assess this dataset against an expert-annotated dataset and find that it is dramatically more accurate than the existing OIE2016 benchmark dataset.

We also implement a scorer that computes precision, recall and area under p-r curve for a given system output by matching it with the CaRB dataset. In designing our scorer, we make several design choices that deviate from prior work in both match scores and also in finding the best match for a tuple. We believe our scheme treats various systems fairly. And in one case where CaRB and OIE2016 give different rankings to two Open IE systems, we demonstrate via human evaluation that the ranking given by CaRB is the accurate one. We release the dataset and scorer for further use by research community.

Chapter 4

IMoJIE - Iterative Memory Based Joint Open IE

4.1 Overview

We design the first neural OpenIE system that uses sequential decoding of tuples conditioned on previous tuples. We achieve this by adding every generated extraction so far to the encoder. This iterative process stops when the *EndOfExtractions* tag is generated by the decoder, allowing it to produce a variable number of extractions. We name our system **Iterative MemOry Joint Open Information Extraction (IMoJIE)**.

CopyAttention uses a bootstrapping strategy, where the extractions from OpenIE-4 Christensen et al. (2011b); Pal and Mausam (2016) are used as training data. However, we believe that training on extractions of multiple systems is preferable. For example, OpenIE-4 benefits from high precision compared to ClausIE Del Corro and Gemulla (2013b), which offers high recall. By aggregating extractions from both, IMoJIE could potentially obtain a better precision-recall balance.

However, simply concatenating extractions from multiple systems does not work well, as it leads to redundancy as well as exaggerated noise in the dataset. We devise an unsupervised **Score-and-Filter** mechanism to automatically select a subset of these extractions that are non-redundant and expected to be of high quality. Our approach scores all extractions with a scoring model, followed by filtering to reduce redundancy.

We compare IMoJIE against several neural and non-neural systems, including our extension of CopyAttention that uses BERT Devlin et al. (2019) instead of an LSTM at encoding time, which forms a very strong baseline. On the recently proposed CaRB metric, which penalizes redundant extractions Bhardwaj et al. (2019), IMoJIE outperforms CopyAttention by about 18 pts in F1 and our strong BERT baseline by 2 pts, establishing a new state of the art for OpenIE. We release IMoJIE & all related resources for further research¹.

4.2 Sequential Decoding Model

We now describe IMoJIE, our generative approach that can output a variable number of diverse extractions per sentence. The architecture of our model is illustrated in Figure

¹<https://github.com/dair-iitd/imojie>

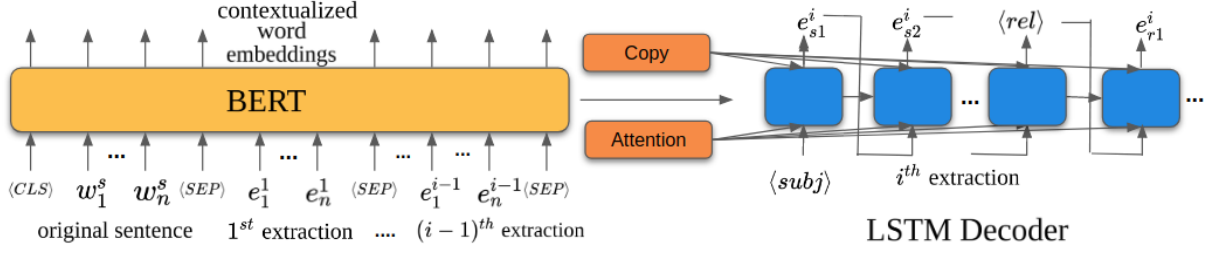


Figure 4.1: One step of the sequential decoding process, for generating the i^{th} extraction, which takes the original sentence and all extractions numbered $1, \dots, i - 1$, previously generated, as input.

4.1. At a high level, the next extraction from a sentence is best determined in context of all other 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.

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 $\langle rel \rangle$ and $\langle obj \rangle$ tokens to indicate the start of relation and object respectively. The iterative process continues until the *EndOfExtractions* token is generated.

The overall process can be summarized as:

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 output and the gold output.

4.3 Aggregating Bootstrapped Data

4.3.1 Single Bootstrapping System

To train generative neural models for the task of OpenIE, we need a set of sentence-extraction pairs. It is ideal to curate such a training dataset via human annotation, but that is impractical, considering the scale of training data required for a neural model. We follow Cui et al. (2018a), and use bootstrapping — using extractions from a pre-existing OpenIE system as ‘silver’-labeled (as distinct from ‘gold’-labeled) instances to train the neural model. We first order all extractions in the decreasing order of confidences output by the original system. We then construct training data in IMOJIE’s input-output format, assuming that this is the order in which it should produce its extractions.

4.3.2 Multiple Bootstrapping Systems

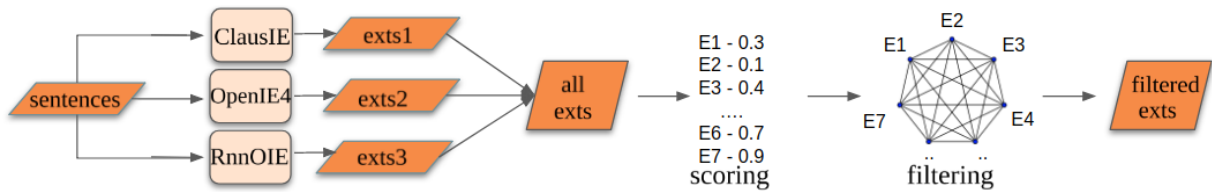


Figure 4.2: Ranking-Filtering subsystem for combining extractions from multiple open IE systems in an unsupervised fashion. (‘Exts’=extractions.)

Different OpenIE systems have diverse quality characteristics. For example, the human-estimated (precision, recall) of OpenIE-4 is (61, 43) while that of ClausIE is (40, 50). Thus, by using their combined extractions as the bootstrapping dataset, we might potentially benefit from the high precision of OpenIE-4 and high recall of ClausIE.

However, simply pooling all extractions would not work, because of the following serious hurdles.

No calibration: Confidence scores assigned by different systems are not calibrated to a comparable scale.

Redundant extractions: Beyond exact duplicates, multiple systems produce similar extractions with low marginal utility.

Wrong extractions: Pooling inevitably pollutes the silver data and can amplify incorrect instances, forcing the downstream open IE system to learn poor-quality extractions.

We solve these problems using a **Score-and-Filter** framework, shown in Figure 4.2.

Scoring: All systems are applied on a given sentence, and the pooled set of extractions are scored such that good (correct, informative) extractions generally achieve higher values compared to bad (incorrect) and redundant ones. In principle, this score may be estimated by the generation score from IMOJIE, trained on a single system.

In practice, such a system is likely to consider extractions similar to its bootstrapping training data as good, while disregarding extractions of other systems, even though those extractions may also be of high quality. To mitigate this bias, we use an IMOJIE model, pre-trained on a *random bootstrapping dataset*. The random bootstrapping dataset is generated by picking extractions for each sentence randomly from any one of the bootstrapping systems being aggregated. We assign a score to each extraction in the pool based on the confidence value given to it by this IMOJIE (Random) model.

Filtering: We now filter this set of extractions for redundancy. Given the set of ranked extractions in the pool, we wish to select that subset of extractions that have the best confidence scores (assigned by the random-bootstrap model), while having minimum similarity to the other selected extractions.

We model this goal as the selection of an optimal subgraph from a suitably designed complete weighted graph. Each node in the graph corresponds to one extraction in the pool. Every pair of nodes (u, v) are connected by an edge. Every edge has an associated weight $R(u, v)$ signifying the similarity between the two corresponding extractions. Each node u is assigned a score $f(u)$ equal to the confidence given by the random-bootstrap model.

Given this graph $G = (V, E)$ of all pooled extractions of a sentence, we aim at selecting a subgraph $G' = (V', E')$ with $V' \subseteq V$, such that the most significant ones are selected, whereas the extractions redundant with respect to already-selected ones are discarded. Our objective is

$$\max_{G' \subseteq G} \sum_{i=1}^{|V'|} f(u_i) - \sum_{j=1}^{|V'|-1} \sum_{k=j+1}^{|V'|} R(u_j, u_k), \quad (4.1)$$

where u_i represents node $i \in V'$. We compute $R(u, v)$ as the ROUGE2 score between the serialized triples represented by nodes u and v . We can intuitively understand the first term as the aggregated sum of significance of all selected triples and second term as the redundancy among these triples.

If G has n nodes, we can pose the above objective as:

$$\max_{\mathbf{x} \in \{0,1\}^n} \mathbf{x}^\top \mathbf{f} - \mathbf{x}^\top \mathbf{R} \mathbf{x}, \quad (4.2)$$

where $\mathbf{f} \in \mathbb{R}^n$ representing the node scores, i.e., $f[i] = f(u_i)$, and $\mathbf{R} \in \mathbb{R}^{n \times n}$ is a symmetric matrix with entries $R_{j,k} = \text{ROUGE2}(u_j, u_k)$. \mathbf{x} is the decision vector, with $x[i]$ indicating

whether a particular node $u_i \in V'$ or not. This is an instance of Quadratic Boolean Programming and is NP-hard, but in our application n is modest enough that this is not a concern. We use the QPBO (Quadratic Pseudo Boolean Optimizer) solver² Rother et al. (2007) to find the optimal \mathbf{x}^* and recover V' .

4.4 Experimental Setup

4.4.1 Training Data Construction

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 ensures that our model is not biased towards data from any single domain.

We run OpenIE-4³, ClausIE⁴ and RnnOIE⁵ on these sentences to generate a set of OpenIE tuples for every sentence, which are then ranked and filtered using our Score-and-Filter technique. These tuples are further processed to generate training instances in IMOJIE’s input-output format.

Each sentence contributes to multiple (input, output) pairs for the IMOJIE model. The first training instance contains the sentence itself as input and the first tuple as output. For example, (“I ate an apple and an orange.”, “I; ate; an apple”). The next training instance, contains the sentence concatenated with previous tuple as input and the next tuple as output (“I ate an apple and an orange. [SEP] I; ate; an apple”, “I; ate; an orange”). The final training instance generated from this sentence includes all the extractions appended to the sentence as input and *EndOfExtractions* token as the output. Every sentence gives the seq2seq learner one training instance more than the number of tuples.

While forming these training instances, the tuples are considered in decreasing order of their confidence scores. If some OpenIE system does not provide confidence scores for extracted tuples, then the output order of the tuples may be used.

4.4.2 Dataset and Evaluation Metrics

We use the CaRB data and evaluation framework Bhardwaj et al. (2019) to evaluate the systems at different confidence thresholds, yielding a precision-recall curve. We identify three important summary metrics from the P-R curve.

²<https://pypi.org/project/thingpbo/>

³<https://github.com/knowitall/openie>

⁴<https://www.mpi-inf.mpg.de/clausie>

⁵<https://github.com/gabrielStanovsky/supervised-oie>

Optimal F1: We find the point in the P-R curve corresponding to the largest F1 value and report that. 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.

Last F1: This is the F1 score computed at the point of zero confidence. This is of importance when we cannot compute the optimal threshold, due to lack of any gold-extractions for the domain. Many downstream applications of OpenIE, such as text comprehension (Stanovsky et al., 2015b) and sentence similarity estimation (Christensen et al., 2014), use *all* the extractions output by the OpenIE system. Last F1 is an important measure for such applications.

4.4.3 Comparison Systems

System	Metric		
	Opt. F1	AUC	Last F1
Stanford-IE	23	13.4	22.9
OLLIE	41.1	22.5	40.9
PropS	31.9	12.6	31.8
MinIE	41.9	-*	41.9
OpenIE-4	51.6	29.5	51.5
OpenIE-5	48.5	25.7	48.5
ClausIE	45.1	22.4	45.1
CopyAttention	35.4	20.4	32.8
RNN-OIE	49.2	26.5	49.2
Sense-OIE	17.2	-*	17.2
Span-OIE	47.9	-*	47.9
CopyAttention + BERT	51.6	32.8	49.6
IMoJIE	53.5	33.3	53.3

Table 4.1: Comparison of various OpenIE systems - non-neural, neural and proposed models.

(*) Cannot compute AUC as Sense-OIE, MinIE do not emit confidence values for extractions and released code for Span-OIE does not provision calculation of confidence values. In these cases, we report the Last F1 as the Opt. F1

We compare IMoJIE against several non-neural baselines, including Stanford-IE, OpenIE-4, OpenIE-5, ClausIE, PropS, MinIE, and OLLIE. We also compare against the sequence labeling baselines of RnnOIE, SenseOIE, and the span selection baseline of SpanOIE. Probably the most closely related baseline to us is the neural generation baseline of CopyAttention. To increase CopyAttention’s diversity, we compare against an English version of Logician, which adds coverage attention 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.

Finally, because our model is based on BERT, we reimplement CopyAttention with a BERT encoder — this forms a very strong baseline for our task. Table 4.1 enlists the CaRB scores of these systems.

4.4.4 Implementation

We implement IMOJIE in the AllenNLP framework⁶ (Gardner et al., 2018) using Pytorch 1.2. We use “BERT-small” model for faster training. 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.

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

4.5 Results and Analysis

4.5.1 Performance of Existing Systems

How well do the neural systems perform as compared to the rule-based systems?

Using CaRB evaluation, we find that, contrary to previous papers, neural OpenIE systems are not necessarily better than prior non-neural systems (Table 4.1). Among the systems under consideration, the best non-neural system reached Last F1 of 51.5, whereas the best existing neural model could only reach 49.2. Deeper analysis reveals that CopyAttention produces redundant extractions conveying nearly the same information, which CaRB effectively penalizes. RnnOIE performs much better, however suffers due to its lack of generating auxilliary verbs and implied prepositions. Example, 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), which does not take advantage of bootstrapping techniques.

4.5.2 Performance of IMOJIE

How does IMOJIE perform compared to the previous neural and rule-based systems?

In comparison with existing neural and non-neural systems, IMOJIE trained on aggregated bootstrapped data performs the best. It outperforms OpenIE-4, the best existing

⁶<https://github.com/allenai/allennlp>

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

OpenIE system, by 1.9 F1 pts, 3.8 pts of AUC, and 1.8 pts of Last-F1. Qualitatively, we find that it makes fewer mistakes than OpenIE-4, probably because OpenIE-4 accumulates errors from upstream parsing modules (see Table 4.2).

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

IMoJIE outperforms CopyAttention by large margins – about 18 Optimal F1 pts and 13 AUC pts. Qualitatively, it outputs non-redundant extractions through the use of its iterative memory (see Table 4.3), and a variable number of extractions owing to the *EndofExtractions* token. It also outperforms CopyAttention with BERT, which is a very strong baseline, by 1.9 Opt. F1 pts, 0.5 AUC and 3.7 Last F1 pts. IMoJIE consistently outperforms CopyAttention with BERT over different bootstrapping datasets (see Table 4.5).

Figure 4.3 shows that the precision-recall curve of IMoJIE is consistently above that of existing OpenIE systems, emphasizing that IMoJIE is consistently better than them across the 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. CaRB evaluation of SpanOIE⁷ results in (precision, recall, F1) of (58.9, 40.3, 47.9). SpanOIE sources its training data only from OpenIE-4. In order to be fair, we compare it against IMoJIE trained only on data from OpenIE-4 which evaluates to (60.4, 46.3, 52.4). Hence, IMoJIE outperforms SpanOIE, both in precision and recall.

Attention is typically used to make the model focus on words which are considered

⁷https://github.com/zhanjunlang/Span_OIE

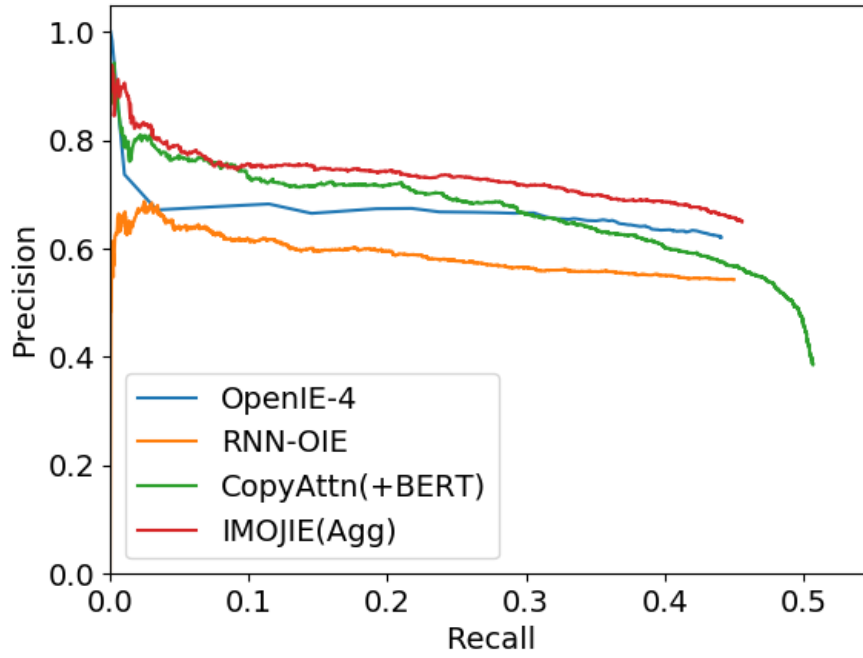


Figure 4.3: Precision-Recall curve of OpenIE Systems.

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 figure out that the words ‘prime’ and ‘minister’ have already been covered, and thus push the decoder to prioritize ‘founding’ and ‘justice’. Appendix A.4 analyzes the attention patterns of the model when generating the intermediate extraction in the above example and shows that IMOJIE gives less attention to already covered words.

4.5.3 Redundancy

What is the extent of redundancy in IMOJIE when compared to earlier OpenIE systems?

System	Metric		
	Opt. F1	AUC	Last F1
CopyAttention	35.4	20.4	32.8
CoverageAttention	41.8	22.1	41.8
CoverageAttention+BERT	47.9	27.9	47.9
Diverse Beam Search	46.1	26.1	39.6
IMOJIE (w/o BERT)	37.9	19.1	36.6
IMOJIE	53.2	33.1	52.4

Table 4.4: Models to solve the redundancy issue prevalent in Generative Neural OpenIE systems. All systems are bootstrapped on OpenIE-4.

System	Bootstrapping System			
	OpenIE-4	OpenIE-5	ClausIE	RnnOIE
Base	50.7, 29, 50.7	47.4, 25.1, 47.4	45.1, 22.4, 45.1	49.2, 26.5, 49.2
CopyAttention+BERT	51.6, 32.8, 49.6	48.7, 29.4 , 48.0	47.4, 30.2, 43.6	47.9, 30.6, 41.1
IMoJIE	53.2, 33.1, 52.4	48.8 , 27.9, 48.7	49.2, 31.4, 45.5	51.3, 31.1, 50.8

Table 4.5: Evaluating models trained with different bootstrapping systems.

We also investigate other approaches to 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 4.4 reports that both these approaches indeed make significant improvements on top of CopyAttention scores. In particular, 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 CopyAttention baseline by 12.1 pts in AUC and 4.4 pts in Last 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, 19.6**) pts in (Optimal F1, AUC, Last F1) over existing seq2seq OpenIE 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 tuples. Higher value of these measures suggest higher redundancy among the extractions. IMoJIE is significantly better than CopyAttention+BERT, the strongest baseline, on both these measures (Table 4.6). Interestingly, IMoJIE has a lower redundancy than even the gold triples; this is due to imperfect recall.

Extractions	Metric		
	MNO	IOU	#Tuples
CopyAttention+BERT	2.805	0.463	3159
IMOJIE	1.282	0.208	1620
Gold	1.927	0.31	2650

Table 4.6: Measuring redundancy of extractions. MNO stands for Mean Number of Occurrences. IOU stands for Intersection over Union.

4.5.4 The Value of Iterative Memory

To what extent does the IMOJIE style of generating tuples improve performance, over and above the use of BERT?

We add BERT to CopyAttention model to generate another baseline for a fair comparison against the IMOJIE model. When trained only on OpenIE-4, IMOJIE continues to outperform CopyAttention+BERT baseline by (1.6, 0.3, 2.8) pts in (Optimal F1, AUC, Last F1), which provides strong evidence that the improvements are not solely by virtue of using a better encoder. We repeat this experiment over different (single) bootstrapping datasets. Table 4.5 depicts that IMOJIE consistently outperforms CopyAttention+BERT model.

We also note that the order in which the extractions are presented to the model (during training) is indeed important. On training IMOJIE using a randomized-order of extractions, we find a decrease of 1.6 pts in AUC (averaged over 3 runs).

4.5.5 The value of Score-and-Filter

To what extent does the scoring and filtering approach lead to improvement in performance?

IMOJIE aggregates extractions from multiple systems through the scoring and filtering approach. It uses extractions from OpenIE-4 (190K), ClausIE (202K) and RnnOIE (230K) to generate a set of 215K tuples. Table 4.7 reports that IMOJIE does not perform well when this aggregation mechanism is turned off. We also try two supervised approaches to aggregation, by utilizing the gold extractions from CaRB’s dev set.

Filtering	Metric		
	Opt. F1	AUC	Last F1
None	49.7	34.5	37.4
Extraction-based	46	29.2	44.9
Sentence-based	49.5	32.7	48.6
Score-And-Filter	53.5	33.3	53.3

Table 4.7: Performance of IMOJIE on aggregated dataset **OpenIE-4+ClausIE+RnnOIE**, with different filtering techniques. For comparison, SenseOIE trained on multiple system extractions gives an F1 of 17.2 on CaRB.

- **Extraction Filtering:** For every sentence-tuple pair, we use a binary classifier that decides whether or not to consider that extraction. The input features of the classifier are the $[CLS]$ -embeddings generated from BERT after processing the concatenated sentence and extraction. The classifier is trained over tuples from CaRB’s dev set.
- **Sentence Filtering:** We use an IMOJIE model (bootstrapped over OpenIE-4), to score all the tuples. Then, a Multilayer Perceptron (MLP) predicts a confidence threshold to perform the filtering. Only extractions with scores greater than this threshold will be considered. The input features of the MLP include the length of sentence, IMOJIE (OpenIE-4) scores, and GPT Radford et al. (2018) scores of each extraction. This MLP is trained over sentences from CaRB’s dev set and the gold optimal confidence threshold calculated by CaRB.

We observe that the Extraction, Sentence Filtering are better than no filtering by 7.5, 11.2 pts in Last F1, but worse at Opt. F1 and AUC. We hypothesise that this is because the training data for the MLP (640 sentences in CaRB’s dev set), is not sufficient and the features given to it are not sufficiently discriminative. Thereby, we see the value of our unsupervised Score-and-Filter that improves the performance of IMOJIE by (3.8, 15.9) pts in (Optimal F1, Last F1). The 1.2 pt decrease in AUC is due to the fact that the IMOJIE (no filtering) produces many low-precision extractions, that inflates the AUC.

Bootstrapping Systems	Metric		
	Opt. F1	AUC	Last F1
ClausIE	49.2	31.4	45.5
RnnOIE	51.3	31.1	50.8
OpenIE-4	53.2	33.1	52.4
OpenIE-4+ClausIE	51.5	32.5	47.1
OpenIE-4+RnnOIE	53.1	32.1	53.0
ClausIE+RnnOIE	50.9	32.2	49.8
All	53.5	33.3	53.3

Table 4.8: IMOJIE trained with different combinations of bootstrapping data from 3 systems - OpenIE-4, ClausIE, RNNIOE. Graph filtering is not used over single datasets.

Table 4.8 suggests that the model trained on all three aggregated datasets perform better than models trained on any of the single/doubly-aggregated datasets. Directly applying the Score-and-Filter method on the test-extractions of RnnOIE+OpenIE-4+ClausIE gives (Optimal F1, AUC, Last F1) of (50.1, 32.4, 49.8). This shows that training the model on the aggregated dataset is important.

Computational Cost: The training times for CopyAttention+BERT, IMOJIE (OpenIE-4) and IMOJIE (including the time taken for Score-and-Filter) are 5 hrs, 13 hrs and 30 hrs respectively. This shows that the performance improvements come with an increased computational cost, and we leave it to future work to improve the computational efficiency of these models.

4.6 Error Analysis

We randomly selected 50 sentences from the CaRB validation set. We consider only sentences where at least one of its extractions shows the error. We identified four major phenomena contributing to errors in the IMoJIE model:

- (1) **Missing information:** 66% of the sentences have at least one of the relations or arguments or both missing in predicted extractions, which are present in gold extractions. This leads to incomplete information.
- (2) **Incorrect demarcation:** Extractions in 60% of the sentences have the separator between relation and argument identified at the wrong place.
- (3) **Missing conjunction splitting:** In 32% of the sentences, our system fails to separate out extractions by splitting a conjunction. E.g., in the sentence “US 258 and NC 122 parallel the river north ...”, IMoJIE predicts just one extraction (US 258 and NC 122; parallel; ...) as opposed to two separate extractions (US 258; parallel; ...) and (NC 122; parallel; ...) as in gold.
- (4) **Grammatically incorrect extractions:** 38% sentences have a grammatically incorrect extraction (when serialized into a sentence).

Additionally, we observe 12% sentences still suffering from **redundant** extractions and 4% **miscellaneous** errors.

4.7 Discussion

We propose IMoJIE for the task of OpenIE. IMoJIE significantly improves upon the existing OpenIE systems in all three metrics, Optimal F1, AUC, and Last F1, establishing a new State Of the Art system. Unlike existing neural OpenIE systems, IMoJIE produces non-redundant as well as a variable number of OpenIE tuples depending on the sentence, by iteratively generating them conditioned on the previous tuples. Additionally, we also contribute a novel technique to combine multiple OpenIE datasets to create a high-quality dataset in a completely unsupervised manner. We release the training data, code, and the pretrained models.⁸

IMoJIE presents a novel way of using attention for text generation. Bahdanau et al. (2015) showed that attending over the input words is important for text generation. See et al. (2017) showed that using a coverage loss to track the attention over the decoded words improves the quality of the generated output. We add to this narrative by showing that deep inter-attention between the input and the partially-decoded words (achieved by adding previous output in the input) creates a better representation for iterative generation of triples. This general observation may be of independent interest beyond OpenIE, such as

⁸<https://github.com/dair-iitd/imojie>

in text summarization.

Chapter 5

Conjunction Splitting

5.1 Problems Remaining in OpenIE

After addressing multiple key issues of OpenIE systems through IMoJIE, we sought to analyse the problems that are still prevalent with OpenIE systems. To begin with, we carefully analysed the erroneous outputs of IMoJIE and other systems. We could broadly categorise the remaining errors into the following types.

5.1.1 Missing Information

The major source of OpenIE errors is missing information in the OpenIE tuples. For IMoJIE, about 66% of the sentences have at least one of the relations or arguments or both missing in predicted extractions, which are present in gold extractions. This leads to incomplete information. For eg.

Sentence:

“George got married to Alice in March beside the lake.”

Set of Extractions with Missing Information:

(George ; got married ;)

Gold Extractions:

(George ; got married to ; Alice)

(George ; got married in ; March)

(George ; got married beside ; the lake)

5.1.2 Incorrect Conjunction Splitting

One the guidelines of OpenIE is *atomicity*. The principle of atomicity states that every tuple must be indivisible. In case of sentences with conjunctions, this translates to deriving multiple OpenIE tuples from the sentence, one with each conjunct. The output tuples where the coordination phrase has not been split are counted under the category of incorrect conjunction splitting. For eg.

Sentence:

‘‘I ate an apple and an orange.’’

Incorrect Extraction:

(I ; ate ; an apple and an orange)

Gold Extractions:

(I ; ate ; an apple)

(I ; ate ; an orange)

It is worth noting that as much as 32% of IMOJIE’s errors are attributed to incorrect conjunction splitting.

5.1.3 Grammatical Errors

OpenIE tuples should not only identify the different parts of an extraction correctly, but should also lay them down in a grammatical structure. This means that when the extraction is serialised, the yielded sentence should be grammatical. For eg.

Sentence:

‘‘US President Donald Trump’’

Incorrect Extraction:

(Donald Trump ; President ; US)

Gold Extractions:

(Donald Trump ; is the president of ; US)

Roughly 38% of IMOJIE’s errors were due to grammatical errors in its extractions.

5.1.4 Missing Output Context

OpenIE systems should be capable of generating the context of the output tuples whenever necessary. Rule-based systems like OpenIE-5 were able to generate context alongside the extractions yet IMOJIE and many recent OpenIE systems do not possess this ability. For eg.

Sentence:

‘‘The police claimed, "Brutus broke into the house." ’’

Incorrect Extraction:

(Brutus ; broke into ; the house)

Gold Extractions:

[Context: The police claimed] (Brutus ; broke into ; the house)

5.1.5 Coreference Resolution

OpenIE systems should be able to resolve coreferences in the sentence. They should generate each tuple such that the tuples do not depend on each other to clarify their meaning. However, most OpenIE systems are not able to resolve coreferences till data. For eg.

Sentence:

‘‘Alice ate an apple and then she went for a walk.’’

Incorrect Extraction:

(she ; went for ; a walk)

Gold Extractions:

(Alice ; went for ; a walk)

5.1.6 Incorrect Identification of Boundaries

OpenIE systems are also known to incorrectly identify the relation and argument boundaries within a sentence. This includes errors like the object argument starts with a preposition instead of placing that preposition within the relation. In case of IMOJIE, nearly 60% of the sentences have the separator between relation and argument identified at the wrong place. The major cause of this type of error is attributed to fallacies in the training data, however it may even creep in otherwise. For eg.

Sentence:

‘‘Jonathon is going for a walk.’’

Incorrect Extraction:

(Jonathon ; is going ; for a walk)

Gold Extractions:

(Jonathon ; is going for ; a walk)

5.2 Why to Split Conjunctions?

OpenIE systems perform much better on short simple sentence as compared to long and complicated ones. This seems obvious since long sentences encompass multiple informations often intertwined in complex sentence structures. It requires much more expertise even for a human annotator to pick out all of them in the form of OpenIE tuples. However, when viewed in light of incorrect conjunction splitting done by OpenIE systems, this gives rise to a potent idea. We could design a separate module to explicitly break a complex sentence into several simple ones by splitting on conjunctions. Passing these simpler sentences through OpenIE systems in place of the original sentence would yield higher quality extractions. This seemed to be a low-hanging fruit, hence we decided to go ahead with this idea.

5.2.1 Analysis of Existing Systems

In order to validate our hypothesis that a better module for conjunction splitting was actually required, we analysed the conjunction splitting ability of some popular rule-based systems. A separate analysis had shown that **and** is the most frequent conjunction occurring in sentences with conjunctions. Hence, in order to expedite the current analysis, we chose only sentences with **and**. There were about 285 of them in the CaRB dev set. We manually annotated (refer table 5.1) each of these sentences to determine how many of them were distributive i.e. how many sentences were such that they could be split into multiple simpler sentences.

System	No. of Sentences with AND	No. of Distributive Sentences	Percentage of Distributive Sentences
Gold	285	207	72.6
OpenIE-4		130	45.6
OpenIE-5		261	91.5

Table 5.1: Conjunction Splitting in Existing Rule-Based Systems

Of the 285 sentences annotated, 72.6% were distributive. OpenIE-4 and OpenIE-5, two popular rule-based systems used for this analysis, split up 45.6% and 91.5% of these sentences respectively. This confirms that OpenIE-4 is too conservative in splitting conjunctions whereas OpenIE-5 is too aggressive. Moreover, none of them is close enough to the ideal split ratio of about 72%. This analysis, therefore, confirms our hypothesis that a better module for conjunction splitting is required.

5.3 HC-Tree Based Splitting

In this section, we discuss an approach for a conjunction-splitting model. The idea is based on **Hierarchical Coordination Trees**. An HC tree is an efficient representation of nested

coordination phrases occurring in a sentence. Figure 5.1 depicts an example of a complex sentence with nested coordination phrases and the corresponding HC tree.

“[(Jeff Bezos, an American [(electrical engineer) and [(technology) and (retail)] entrepreneur]), founded [(Amazon.com) and (Blue Origin)] and (his diversified business interests include [(books), (aerospace) and (newspapers)])].”

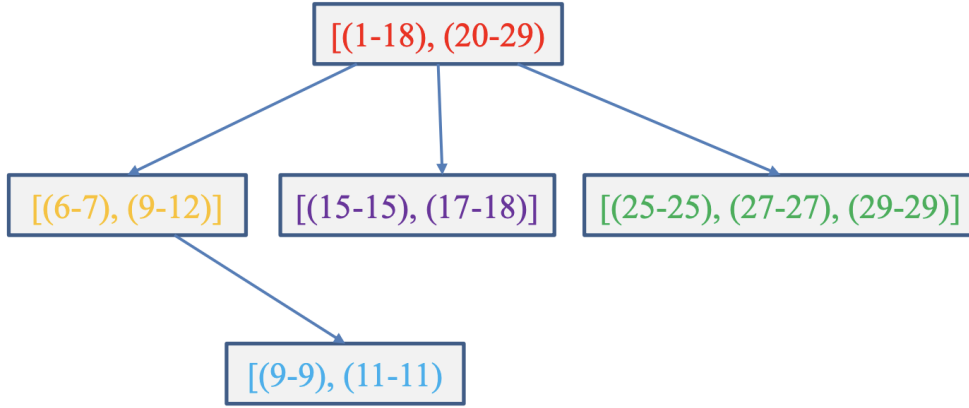


Figure 5.1: Example of a sentence and its Hierarchical Coordination (HC) Tree

For our analysis, we use $[D_i]$ token at the beginning and end of each coordination phrase. Also, each conjunct is encapsulated with $[SS]$ and $[ES]$ tokens.

5.3.1 Properties of Conjuncts

It is important to discuss two key properties of conjuncts before diving into the approach. Consider the sentence I ate $[D1]$ $[SS]$ an apple $[ES]$ and $[SS]$ an orange $[ES]$ $[D1]$. to understand the properties.

1. **Similarity:** The conjuncts are similar to each other in their syntactic form.
For eg., an apple and an orange are syntactically similar.
2. **Replaceability:** Each of the conjunct can potentially replace the entire coordination phrase in the original sentence and to yield a meaning sentence.
For eg. ‘‘I ate an apple.’’ and ‘‘I ate an orange.’’ are both meaningful sentences.

5.3.2 Proposed Approach

Step 1: Generation of HC Tree

First of all, we need to generate an HC tree for the given sentence. For this purpose, we use a sequence to sequence model with BERT as encoder and a Tree-Based decoder (Wang et al., 2018). We use data from PennTree Bank (Marcus et al., 1993) dataset due to the following reasons:

1. *Sufficient number of sentences:* The PennTree Bank has a dev and test set of 1700 and 2416 sentences respectively.
2. *Contains coordinations upto depth 3:* PennTree Bank also has a significant number of sentences with coordinations nested at depths upto 3. It contains 756, 87 and 5 sentences with conjunctions at depth 1, 2 and 3 respectively in dev set.
3. *Resembling Fraction of Distributive Conjunctions:* The fraction of sentences that are distributive are similar for PennTree Bank (64.5%) and the CaRB gold dataset (72.6%) (refer to table 5.1). This indicates that the distribution of sentences in the PennTree bank is similar to that found in a general corpus.

Step 2: Obtaining Split Sentences from HC Tree

The next step is to split the original sentence and obtain simpler sentences. This is done in an iterative manner with increasing depths of split. At each depth, we replace each coordination phrase with all the conjuncts one-by-one. Then for each of the simpler sentences obtained, we repeat this procedure at the next depth till the sentence contains no more coordination phrase. For eg.

Input - Premise (HC tree) :

Today, [D1] [SS] Ram [ES] and [SS] Shyam [ES] [D1] ate [D1] [SS] an apple [ES] and [SS] an orange [ES] [D1] .

Output - Hypothesis (simple sentences) :

Today, Ram ate an apple .

Today, Ram ate an orange .

Today, Shyam ate an apple .

Today, Shyam ate an orange .

Step 3: Validation of Hypothesis

In this step, we check whether or not the coordination phrase split at step 2 was actually distributive. Consider the following examples of a distributive and non-distributive sentence.

Example: Distributive Sentence

PaineWebber Inc. [D1] [SS] filmed a new television commercial at 4 p.m. EDT yesterday [ES] and [SS] had it on the air by last night [ES] [D1] .

PaineWebber Inc. filmed a new television commercial at 4 p.m. EDT yesterday .
PaineWebber Inc. had it on the air by last night .

Example: Non-Distributive Sentence

The [D1] [SS] Ways [ES] and [SS] Means [ES] [D1] Committee will hold a hearing on the bill next Tuesday .

The Ways Committee will hold a hearing on the bill next Tuesday .
The Means Committee will hold a hearing on the bill next Tuesday .

We use pretrained FastText vectors (trained on Wikipedia and Common Crawl datasets) to encode:

- The Premise i.e. the original complex sentence
- The Hypothesis i.e. all the simple sentences obtained after splitting
- General Sentence Corpus (we use 3.9M sentences from Wikipedia of which 700K sentences contain conjunctions)

Next, we compare the *Cosine Similarity score* of the encoded premise and hypothesis with the encoded sentences of the corpus. The similarity-score of these sentences with the general sentences is treated as a proxy for the probability of encountering such a sentence in a general corpus. If the similarity-score of the premise is higher than the least among all similarity-scores of the hypothesis, then we ignore the split and use the premise as it is. Else, we use the split sentences.

Note that steps 2 and 3 have to be done in succession for each depth. This means, that we will split (step 2) the coordination phrases at depth 1, then verify (step 3) each split. Only after that will we move towards splitting (step 2) at depth 2.

Step 4: Feeding the sentence to IMoJIE

By the time we reach this step, we would have split all distributive sentences and identified simpler sentences had they existed. This is the final step wherein we simply feed these sentences to an OpenIE system, IMoJIE in our case. For eg.

Input to IMoJIE, previously :

In these locations , Mr. Friedman says , Retailers are increasingly cautious about expanding and rents have remained steady or in some cases have declined .

Input to IMoJIE, now :

In these locations , Mr. Friedman says , Retailers are increasingly cautious about expanding .

In these locations , Mr. Friedman says , rents have remained steady .

In these locations , Mr. Friedman says , rents in some cases have declined .

5.3.3 Drawbacks

The approach seems promising, however it has several drawbacks.

1. *Sentences closer in encoded space are not semantically similar* The example below demonstrates the top two sentences from the general corpus that are most similar to the given sentence. Clearly, the retrieved sentences are not particularly similar.

Joe Russo (musician) (born December 18 , 1976) is an American drummer and half of the Benevento/Russo Duo .

0.957 Tiffany Shepis (born September 11 , 1979) is an American actress from New York City , who has been involved in film-making since the age of 12 .

0.956 Vincent Grier (born March 14 , 1983) is an American former college basketball player for the University of Minnesota Golden Gophers .

2. *Lack of standard metric to decide extent of similarity of sentences*

The absence of any standard metric to quantify similarity of sentences makes it difficult to comment on the degree of similarity of the retrieved sentences.

3. *Cosine similarity is not representative*

The aim of fetching similar sentences is to calculate a proxy for the probability of the coordination phrase appearing in the context of the sentence i.e.

Prob(coordination phrase | context of the sentence). However, we find that *cosine similarity score* is not representative of the likelihood of the encoded sentence occurring in a general sentence corpus. There are two main reasons for this:

- Cosine similarity is high even for non-similar sentences. As evident from the previous example, the cosine similarity for the best matching sentences were more than 0.95 even though the sentences did not appear to be similar.
- It does not vary much for top few matches. For most of the sentences, the best similarity scores were around 0.95, which is counter intuitive since this indicates that all sentences are almost equally likely to appear in the general corpus. This is trivially false.

On these grounds, we digress away from the HC-Tree based approach to conjunction splitting. We move towards rule-based systems for conjunction splitting.

5.4 Patterns dictating Distributivity of Coordination Phrase

In order to develop a better understanding of distributivity of coordination phrases, we identify the patterns and sentence structures that ultimately dictate whether a sentence will be distributive or not. We notice that any coordination phrase can be matched to a logical operator that it. In our description of the observed patterns, we capitalise the logical operators to distinguish them from similarly spelled conjunctions. Following are the rules that we came up with.

- XOR can never be split
- BUT behaves similar to AND in most cases, it is just that the intonation is different
- OR cannot be split in the consequent. For eg.

If X happens, then Y or Z will happen.

Incorrect to split as:

If X happens, then Y will happen.

If X happens, then Z will happen.

- OR can be split in the antecedent. For eg.

If X or Y happen, then Z will happen.

Correct to split as:

If X happens, then Z will happen.

If Y happens, then Z will happen.

- AND cannot be split in the antecedent. For eg.

If X and Y happen, then Z will happen.

Incorrect to split as:

If X happens, then Z will happen.

If Y happens, then Z will happen.

- AND can be split in the consequent. For eg.

If X happens, then Y and Z will happen.

Correct to split as:

If X happens, then Y will happen.

If X happens, then Z will happen.

- Sequential Constructs ($A \rightarrow B \rightarrow C$) cannot be split. For eg.

Mr. Farley followed a similar pattern when he acquired Northwest Industries Inc. and then sold much of its assets.

5.5 Analysis of Non-Distributive Sentences

We also manually annotated about 220 sentences to identify reasons that lead to non-distributivity of sentences. Of these 220 sentences, 39 were found to be non-distributive due to the reasons listed below. Table 5.2 shows the significance of the various reasons for non-distributivity of sentences.

Reason	Significance
Unsplittable Entity	38.5%
Relation forbids Splitting	25.6%
Antecedent AND	15.4%
Between construct	7.7%
Either-Or construct	2.6%
Consequent OR	2.6%
Idiomatic Phrase	2.6%
BUT changes meaning	2.6%
Sequential construct	2.6%

Table 5.2: Reasons for non-distributivity of sentences along with their significance based on the frequency of occurrence

- **Unsplittable Entity:** When an entity is being referred to in the conjuncts and it cannot be divided. For eg.

[D1] [SS] Ways [ES] and [SS] Means [ES] [D1] Committee

[D1] [SS] green [ES] and [SS] gold [ES] [D1]

[D1] [SS] a run [ES] and [SS] four hits [ES] [D1]

- **Relation forbids splitting:** When the relation is such that splitting the sentence leads to meaningless conjuncts. For eg.

In the aftermath of the 1987 debacle , the brokerage firm began taping commercials in-house , ultimately getting its timing down fast enough to [D1] [SS] tape a commercial after the market closed [ES] and [SS] rush it on the air that night [ES] [D1] .

We 're saying the worst thing that anyone can do is to see the market [D1] [SS] go down [ES] and [SS] dump everything [ES] [D1] , which just drives the prices down further , says John Lampe , PaineWebber's director of advertising .

The team that dumped runs by the bushel on the Chicago Cubs in the National League playoffs was held to just one in two games by the home-team Oakland A 's , the gang that had been done unto similarly by [D1] [SS] the Los Angeles Dodgers [ES] and [SS] Orel Hershiser [ES] [D1] in last year 's tournament .

I [D1] [SS] 'm for the Giants today [ES] , but [SS] only because they lost yesterday [ES] [D1] .

- **Antecedent AND:** For eg.

If you [D1] [SS] get your pitch [ES] , and [SS] take a good swing [ES] [D1] , anything can happen , he later remarked .

The collapsed plan to acquire UAL Corp. , parent of United Airlines, spurred quick action on the legislation , [D1] [SS] introduced Wednesday [ES] and [SS] approved by the subcommittee on a voice vote yesterday [ES] [D1] .

NCR said revenue declined both [D1] [SS] in the U.S. [ES] and [SS] overseas [ES] [D1] , reflecting a world-wide softening of the computer markets.

- **Between construct:** When the two conjuncts are related by *between*, they cannot make meaningful sentences individually. For eg.

In a possible prelude to the resumption of talks between [D1] [SS] Boeing Co. [ES] and [SS] striking Machinists union members [ES] [D1] , a federal mediator said representatives of the two sides will meet with him tomorrow .

Differences remained between the [D1] [SS] North [ES] and [SS] South [ES] [D1] Korean governments , however , over conditions for the exchanges .

[D1] [SS] Seoul [ES] and [SS] Pyongyang [ES] [D1] reached a tentative

agreement to allow visits between families on the divided Korean peninsula .

- **Either-Or construct:** For eg.

The above represents a triumph of either [D1] [SS] apathy [ES] or [SS] civility [ES] [D1].

- **Consequent OR:** For eg.

Today 's Fidelity ad goes a step further , encouraging investors [D1] [SS] to stay in the market [ES] or [SS] even to plunge in with Fidelity [ES] [D1].

- **Idiomatic Phrase:** For eg.

We have [D1] [SS] half the experts saying one thing [ES] and [SS] half the other [ES] [D1] about the course of the economy .

- **BUT changes meaning:** When separating the conjuncts related by the conjunction *but* conveys a meaning different from the original sentence. For eg.

Tokyo stocks closed off a [D1] [SS] significant [ES] but [SS] less-than-alarming [ES] [D1] 1.8 % on thin volume ; Hong Kong stocks declined 6.5 % in orderly trading .

- **Sequential construct ($A \rightarrow B \rightarrow C$):** For eg.

Mr. Farley followed a similar pattern when he [D1] [SS] acquired Northwest Industries Inc. [ES] and [SS] then sold much of its assets [ES] [D1].

This analysis revealed that a rule-based system to tackle conjunction splitting might not be effective since the errors are spread across multiple interesting linguistic phenomenon. Also, it is tough to deal with all types of conjunctions and rules pertaining to conjunction splitting are either not followed all the time, or are too specific to occur frequently.

Chapter 6

MLIL - Multi Level Iterative Labelling

6.1 MLIL Based OpenIE

6.1.1 Major Drawback of IMoJIE: Speed

Despite achieving significant performance boost over the existing OpenIE systems, IMoJIE is not efficient for large scale use because of one major disadvantage: *Speed*. Table 6.1 compares the different OpenIE systems on their performance as well as speed of execution. IMoJIE falls much behinds other OpenIE systems since it is able to process only 3 sentences per second.

System	CaRB		CaRB(1-1)		OIE16-C		Wire57-C	Speed
	F1	AUC	F1	AUC	F1	AUC	F1	Sentences/sec.
MinIE	41.9	-	38.4	-	52.3	-	28.5	8.9
ClausIE	45.0	22.0	40.2	17.7	61.0	38.0	33.2	4.0
OpenIE-4	51.6	29.5	40.5	20.1	54.3	37.1	34.4	20.1
OpenIE-5	48.0	25.0	42.7	20.6	59.9	39.9	35.4	3.1
SenseOIE	28.2	-	23.9	-	31.1	-	10.7	-
SpanOIE	48.5	-	37.9	-	54.0	-	31.9	19.4
RnnOIE	49.0	26.0	39.5	18.3	56.0	32.0	26.4	149.2
Cui et al. (2018b)	51.6	32.8	38.7	19.8	53.5	37.0	33.3	11.5
IMoJIE (Agg)	53.5	33.3	41.4	22.2	56.8	39.6	36.0	2.6
IMoJIE (OIE4)	53.2	33.1	42.2	22.8	56.1	38.1	34.9	2.6
IMoJIE	52.5	32.0	41.0	21.7	55.5	35.0	35.1	157.1
IMoJIE(CL)	54.2	34.2	43.0	23.3	59.4	39.1	37.1	157.1
IMoJIE(CL) + IMoJIE-CA	52.7	33.7	46.2	26.6	65.1	48.0	40.2	83.0

Table 6.1: Evaluation of OpenIE. Using constrained learning, IMoJIE(CL) offers a 60x speedup and better performance scores on all metrics compared to IMoJIE. Adding a coordination analyzer, IMoJIE(CL) + IMoJIE-CA gives the best scores in 3 of the 4 metrics. MinIE, SenseOIE, SpanOIE do not output confidences. Code of SenseOIE is not available to compute speed.

The low speed of execution of IMoJIE is attributed to its high computational cost which comes by virtue of its iterative generation procedure. IMoJIE is a state-of-the-art OpenIE system that re-encodes the partial set of extractions so far when generating the next extraction. This captures dependencies among extractions, reducing the overall redundancy of the output set. However, this repeated re-encoding causes a significant reduction in speed, which limits use at web scale.

On the other hand, *labeling*-based systems like RnnOIE (Stanovsky et al., 2015b) are much faster (150 sentences per second, compared to 3 sentences of IMoJIE) but relatively less accurate. They label each word in the sentence as either *S* (Subject), *R* (Relation), *O* (Object) or *N* (None) for each extraction. However, as the extractions are predicted independently, this does not model the inherent dependencies among the extractions.

The challenge at hand now is to bridge this gap by a system that is both fast and accurate.

6.1.2 OpenIE as 2-D Grid Labelling

Subject	Relation				Object						
Rome	the	capital	of	Italy	is	known	for	it's	rich	history	[is]
Rome	the	capital	of	Italy	is	known	for	it's	rich	history	[is]

Rome the capital of Italy is known for it's rich history [is]

Figure 6.1: The extractions *(Rome; [is] the capital of; Italy)* and *(Rome; is known for; it's rich history)* can be seen as the output of the grid labeling. We additionally introduce a token *[is]* to the input.

We present a novel labelling-based architecture, **Multi Level Iterative Labeling (MLIL)**, which models OpenIE as 2-D grid labeling problem of size (M, N) where M is a pre-defined maximum number of extractions and N is the sentence length, as shown in Figure 6.1.

Each extraction corresponds to one row in the grid. Iterative assignment of labels in the grid helps MLIL capture dependencies among extractions without the need for re-encoding. This makes it 25 times faster (80 sentences per second) than IMoJIE.

While MLIL 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) 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.

6.2 Coordination Analyzer

Furthermore, the existing neural OpenIE models struggle in handling coordination structures, and do not split conjunctive extractions adequately. In response, we first design a new coordination analyzer (Ficler and Goldberg, 2016). It is built with the same MLIL 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 13 pt improvement in F1 over previous best reported result (Teranishi et al., 2019), and a 2.5 pts gain in F1 over a strong BERT baseline.

We then combine the output of our coordination analyzer with our OpenIE system, resulting in a further increase in performance (Table 4.3). We evaluate our final OpenIE system on four metrics from the literature and find that it exceeds in three of them by at least 4.2 pts in F1.

We undertake manual evaluation to reaffirm the gains. Two annotators, blind to the underlying systems, 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 6.2. Here, yield is the number of correct extractions generated by a system. It is a surrogate for recall, since its denominator, number of all correct extractions, is hard to annotate for OpenIE.

We find that MLIL(CL)+CA significantly increases the yield (1.7x) compared to MLIL(CL) at a marginal increase in precision. This result underscores the importance of splitting coordination structures for OpenIE. Our final OpenIE system, MLIL(CL)+CA, is 25x faster than IMoJIE.

System	Precision	Yield	Total Extrs
IMoJIE(CL)	77.9	131	174
IMoJIE(CL) + MLIL-CA	78.8	222	291

Table 6.2: Manual comparison of Precision and Yield on 100 random conjunctive sentences from CaRB Gold.

Chapter 7

OpenIE Milestones and Future Ideas

7.1 OpenIE Milestones

This thesis covers several OpenIE milestones that mark our progress in the world of Open Information Extraction.

1. **Benchmark:** The problem of standardised and reliable benchmarking was solved using CaRB (Bhardwaj et al., 2019)
2. **Redundancy Removal:** IMOJIE(Kolluru et al., 2020) and MLIL have been able to produce non-redundant extractions
3. **Variable Number of OpenIE tuples:** The iterative style architecture of IMOJIE and MLIL overcome the problem of fixed number of extractions output by neural OpenIE systems using beam search.
4. **Conjunction Splitting:** The MLIL-based Coordination Analyzer is now established as a new state-of-the-art.
5. **Coreference Resolution:** Although OpenIE systems do resolve some coreferences, the problem is largely unsolved. It still requires careful attention and is left to future research.
6. **Enhancing the Grammar of Extractions:** As neural systems lead to the rise of generative models in the OpenIE realm, generating grammatically correct extractions becomes increasingly vital. Improving the grammar of extractions has also been left to future research.

7.2 Future Ideas

In the end, we contribute a few ideas to unravel interesting research directions that are yet unexplored.

- **Adding a language model to the extractor:** We have seen that the output extractions of generative OpenIE systems often do not lead to grammatical sentences when serialised. Adding a language model to the extractor seems to be a promising approach to tackle this issue.

- **Set Prediction Approach:** The task of OpenIE is inherently a set prediction task. The output needs to be a *set* of OpenIE tuples. The order in which the tuples are spit out typically holds no significance. However, most of the OpenIE systems, including the state-of-the-art MLIL-based OpenIE as well as IMOJIE, treat it as sequence prediction task. The basic premise of the iterative architecture is also to predict a *sequence-of-sequences*, rather than a *set-of-sequences*. This violates the basic nature of the problem. Trying out set prediction techniques for OpenIE is a direction worth exploring. Gao et al. (2019) present a sequential set generation (SSG) framework which is a good point to start.
- **Optimally Ordering Training Data:** Vinyals et al. (2016) show that the order in which we arrange the input/output data significantly affects performance while learning an underlying model. While training OpenIE models, we arrange the input data based on its confidence score. However, multiple other orders are possible such as increasing order of tuple length. It will be interesting to study what affect does the order of training extractions have on the OpenIE models trained downstream. We might also derive cues to the optimal order for training OpenIE systems from here.
- **Incorporating CaRB’s loss in MLIL:** We can look for ways to incorporate the CaRB metric into the objective function of our model so that it can explicitly be optimised for. Even if this approach does not succeed in improving the OpenIE model, this experiment will reveal flaws in the metric used in our benchmark which we can use to improve our benchmark further.

Appendix A

A SAMPLE APPENDIX

A.1 Performance of IMoJIE 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 6 datasets 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 Optimal F1 score of both Copy Attention + BERT and IMoJIE (Bootstrapped on OpenIE-4) on these partitions as depicted in Figure A.1.

We observe that the performance deteriorates with increasing sentence length which is expected as well. Also, for each of the partitions, IMoJIE marginally performs better as compared to Copy Attention + BERT.

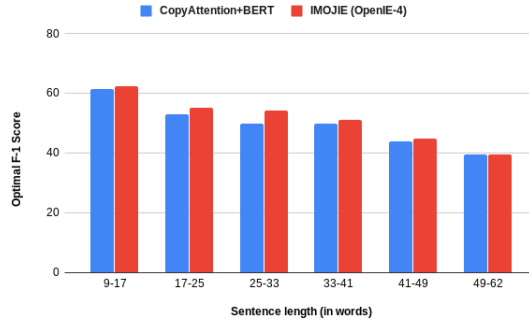


Figure A.1: Measuring performance with varying input sentence lengths

A.2 Measuring Performance of IMoJIE on Varying Beam Size

We perform inference of the CopyAttention with BERT model on CaRB test set with beam sizes of 1, 3, 5, 7, and 11. We observe in Figure A.2 that AUC increases with increasing beam size. A system can surge its AUC by adding several low confidence tuples to its predicted set of tuples. This adds low precision - high recall points to the Precision-Recall curve of

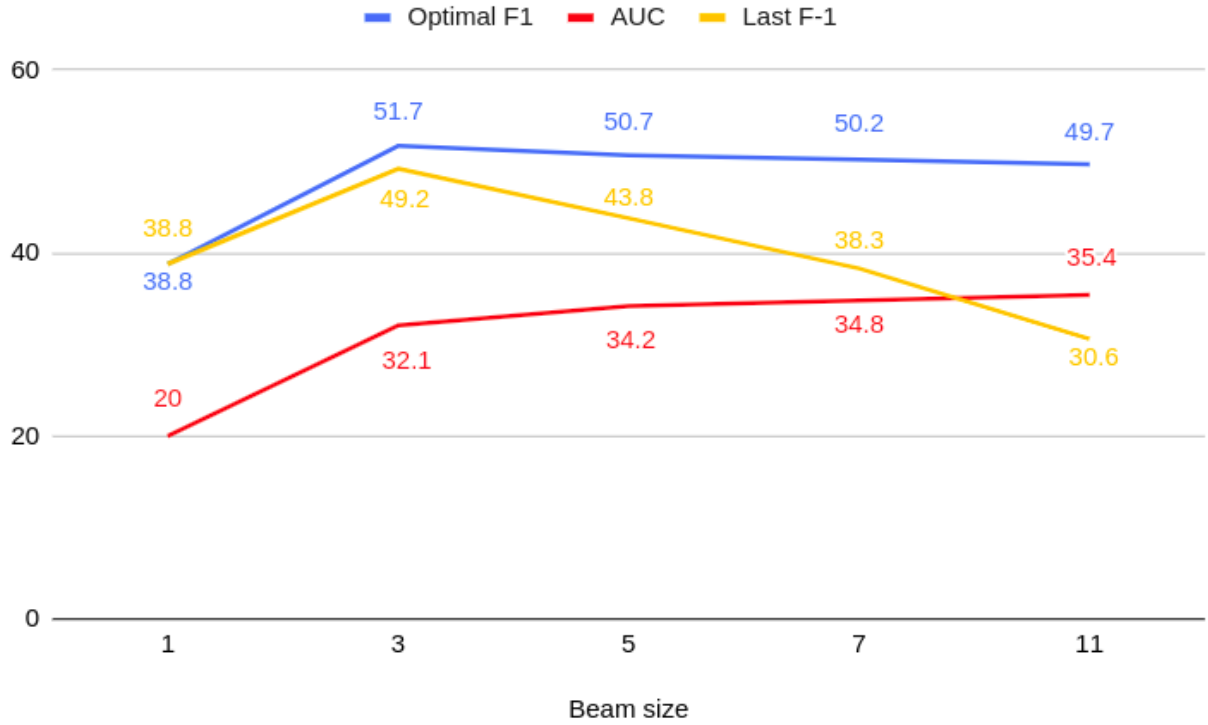


Figure A.2: Measuring performance of CopyAttention with BERT model upon changing the beam size

Model	Dataset					
	Wire57		Penn		Web	
CopyAttention + BERT	45.60,	27.70 , 39.70	18.20,	7.9, 12.40	30.10,	18.00 , 14.60
IMoJIE	46.20 ,	26.60, 46.20	20.20 ,	8.70 , 15.50	30.40 ,	15.50, 26.40

Table A.1: Evaluation on other datasets with the CaRB evaluation strategy

the system leading to higher AUC.

On the other hand, Last F1 experiences a drop at very high beam sizes, thereby capturing the decline in performance. Optimal F1 saturates at high beam sizes since its calculation ignores the extractions below the optimal confidence threshold.

This analysis also shows the importance of using Last F1 as a metric for measuring the performance of OpenIE systems.

A.3 Evaluation of IMoJIE on other datasets

We use sentences from other benchmarks with the CaRB evaluation policy and we find similar improvements, as shown in Table A.1. IMoJIE consistently outperforms our strongest baseline, CopyAttention with BERT, over different test sets. This confirms that IMoJIE is domain agnostic.

A.4 Visualizing Attention

Attention has been used in a wide variety of settings to help the model learn to focus on important things Bahdanau et al. (2015); Xu et al. (2015); Lu et al. (2019). However, the IMOJIE model is able to use attention to understand which words have already been generated, to focus on remaining words. In order to understand how the model achieves this, we visualize the learnt attention weights. There are two attention weights of importance, the learnt attention inside the BERT encoder and the attention between the decoder and encoder. We use BertViz Vig (2019) to visualize the attention inside BERT.

We consider the following sentence as the running example - "he served as the first prime minister of australia and became a founding justice of the high court of australia". We visualize the attention after producing the first extraction - "he; served; as the first prime minister of australia". Intuitively, we understand that the model must focus on the words "founding" and "justice" in order to generate the next extraction - "he; became; a founding justice of the high court of australia". In Figure A.5 and Figure A.6 (where the left-hand column contains the words which are used to attend while right-hand column contains the words which are attended over), we see that the words "prime" and "minister" of the original sentence have high attention over the same words in the first extraction. But the attention for "founding" and "justice" are limited to the original sentence.

Based on these patterns, the decoder is able to give a high attention to the words "founding" and "justice" (as shown in Figure A.7), in-order to successfully generate the second extraction "he; became; a founding justice of the high court of australia".

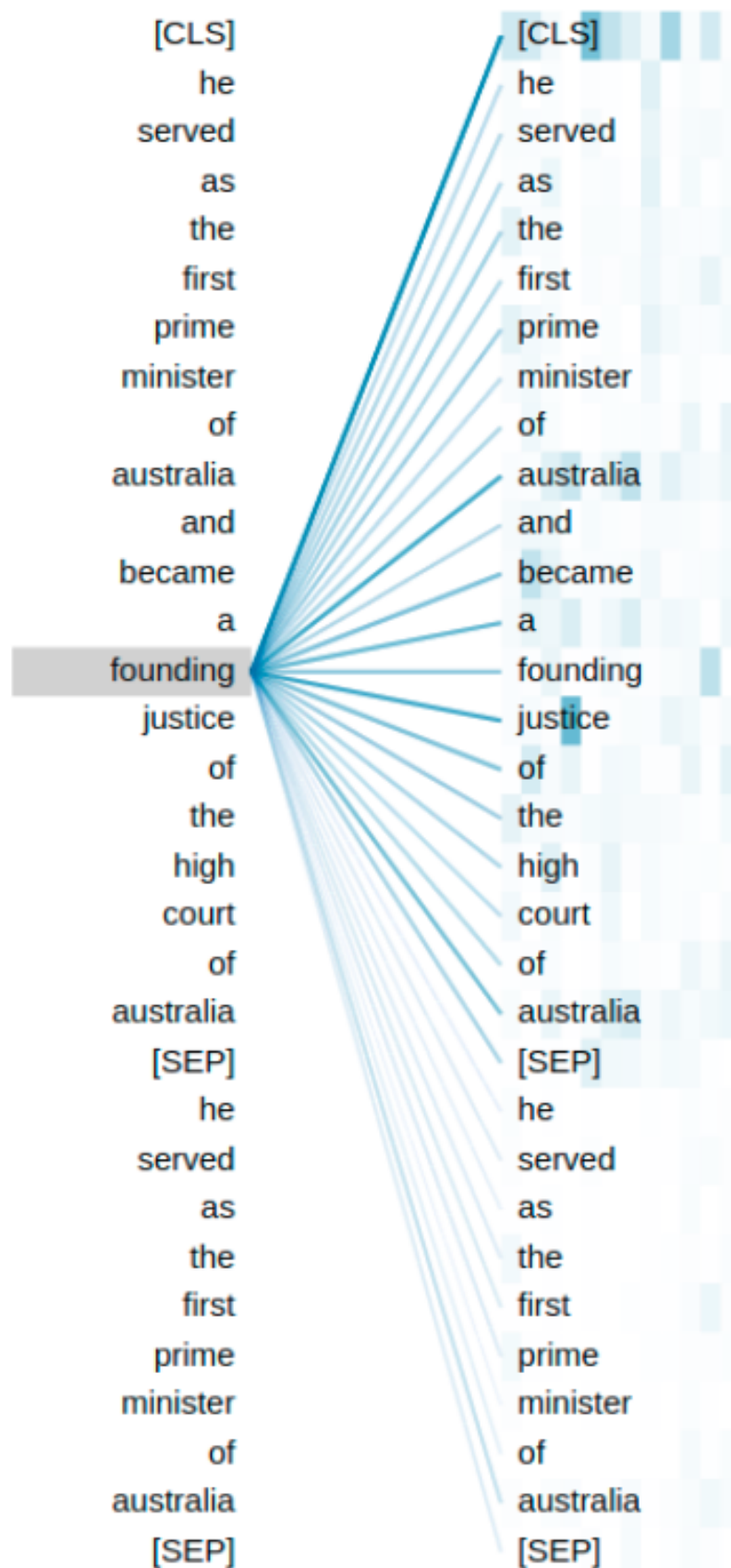


Figure A.3: BERT attention for the word 'founding'

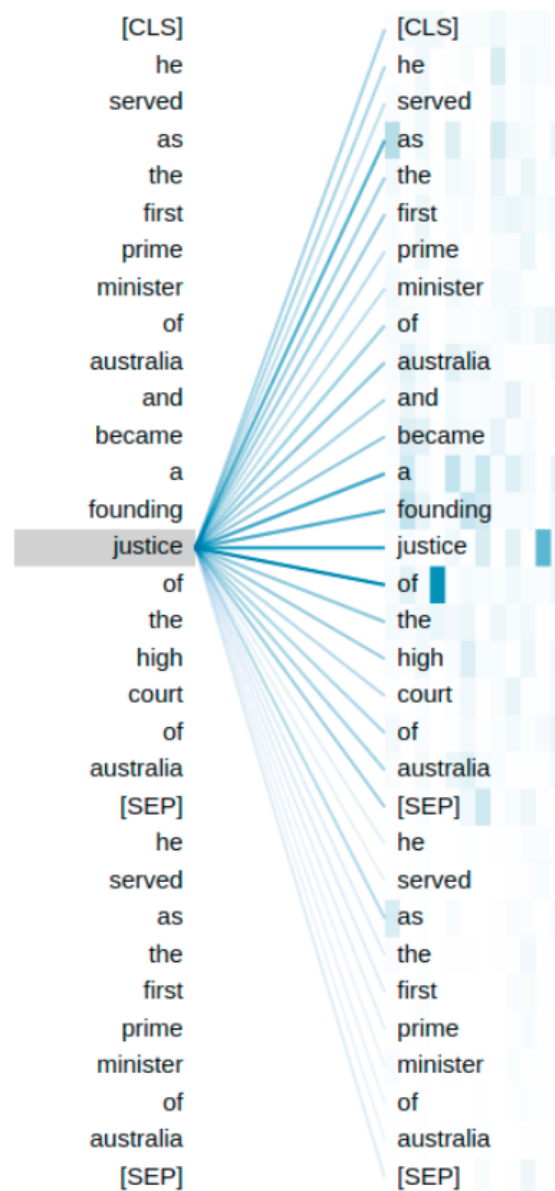


Figure A.4: BERT attention for the word 'justice'

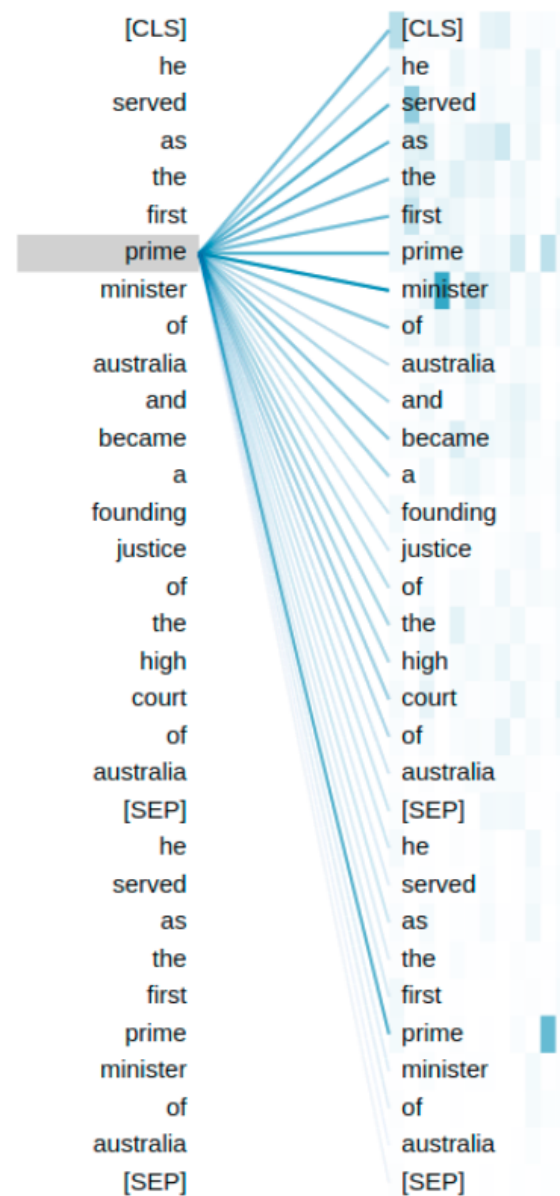


Figure A.5: BERT attention for the word 'prime'

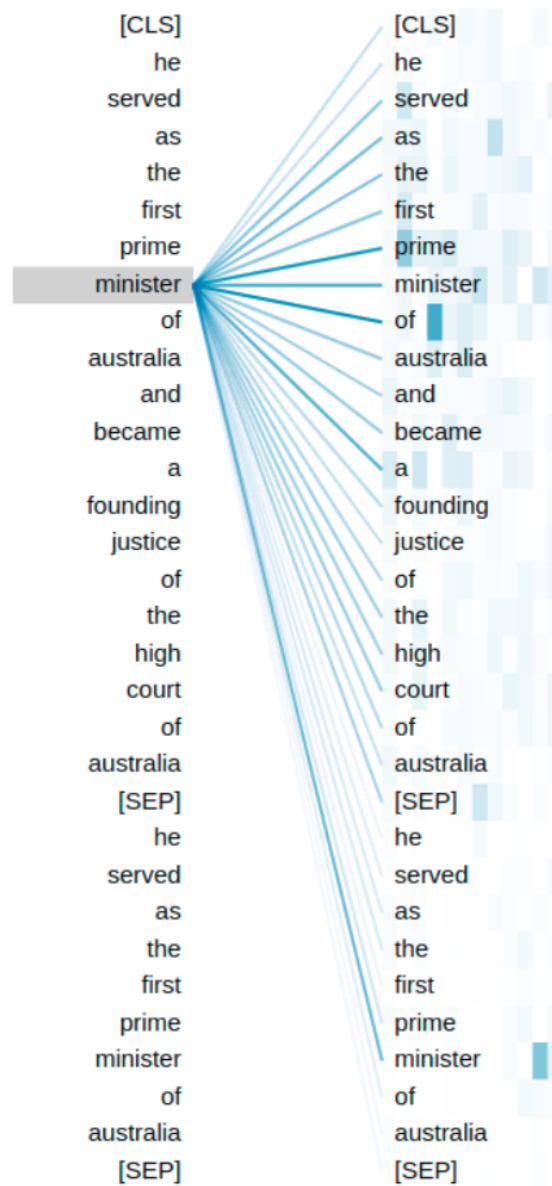


Figure A.6: BERT attention for the word 'minister'

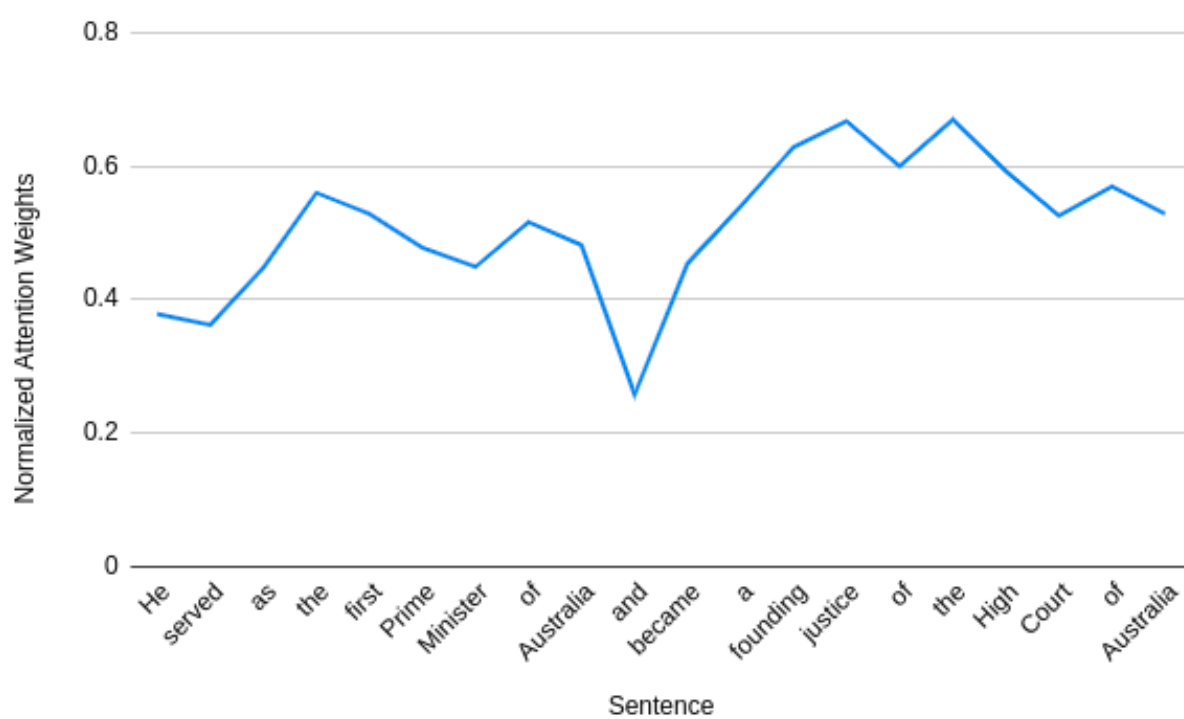


Figure A.7: Attention weights for the decoder

Bibliography

- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. 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*, pages 344–354.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR), 2015*.
- Niranjana Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating Coherent Event Schemas at Scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013*, pages 1721–1731.
- Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007a. Open information extraction from the web. In *International Joint Conference on Artificial Intelligence (IJCAI), 2007*, volume 7, pages 2670–2676.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007b. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.
- Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. 2019. 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*, pages 6263–6268.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011a. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26-29, 2011, Banff, Alberta, Canada*, pages 113–120.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011b. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120. ACM.
- Janara Christensen, Stephen Soderland, Gagan Bansal, et al. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 902–912.
- Lei Cui, Furu Wei, and Ming Zhou. 2018a. Neural open information extraction. In *Proceedings of Association for Computational Linguistics (ACL), 2018*, pages 407–413.

- Lei Cui, Furu Wei, and Ming Zhou. 2018b. Neural open information extraction. In *Proceedings of Association for Computational Linguistics (ACL), 2018*, pages 407–413.
- Luciano Del Corro and Rainer Gemulla. 2013a. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13*, pages 355–366, New York, NY, USA. ACM.
- Luciano Del Corro and Rainer Gemulla. 2013b. ClausIE: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web (WWW), 2013*, pages 355–366. ACM.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011a. Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One, IJCAI'11*, pages 3–10. AAAI Press.
- Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011b. 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*, pages 3–10. IJCAI/AAAI.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011a. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011b. Identifying Relations for Open Information Extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, Edinburgh, Scotland, UK.
- Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. 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), 2019*.
- Jessica Fidler and Yoav Goldberg. 2016. 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*, pages 23–32. The Association for Computational Linguistics.
- Tian Gao, Jian Jhen Chen, Vijil Chenthamarakshan, and Michael J. Witbrock. 2019. A sequential set generation method for predicting set-valued outputs. *AAAI*, abs/1903.05153.

- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.
- Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: minimizing facts in open information extraction. In *Association for Computational Linguistics (ACL), 2017*.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of Association for Computational Linguistics (ACL), 2016*. Association for Computational Linguistics.
- Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. 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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 643–653.
- Shengbin Jia, Yang Xiang, and Xiaojun Chen. 2018. Supervised neural models revitalize the open relation extraction. *CoRR*, abs/1809.09408.
- Zhengbao Jiang, Pengcheng Yin, and Graham Neubig. 2019. Improving Open Information Extraction via Iterative Rank-Aware Learning. In *Proceedings of the Association for Computational Linguistics (ACL), 2019*.
- Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020. IMoJIE: Iterative Memory-Based Joint Open Information Extraction. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Seattle, U.S.A.
- William L  chelle, Fabrizio Gotti, and Philippe Langlais. 2018. Wire57 : A fine-grained benchmark for open information extraction. *CoRR*, abs/1809.08962.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NIPS), 2019*, pages 13–23.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.
- Mausam. 2016a. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), 2016*, pages 4074–4077. AAAI Press.
- Mausam. 2016b. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI’16*, pages 4074–4077. AAAI Press.
- Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012a. 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, EMNLP-CoNLL ’12*, pages 523–534, Stroudsburg, PA, USA. Association for Computational Linguistics.

- Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012b. 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*, pages 523–534. Association for Computational Linguistics.
- Sanket Vaibhav Mehta, Jay Yoon Lee, and Jaime G. Carbonell. 2018. 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*, pages 4958–4963. Association for Computational Linguistics.
- Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. 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)*, pages 560–568.
- Harinder Pal and Mausam. 2016. Donyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, San Diego, CA. Association for Computational Linguistics.
- Harinder Pal and Mausam. 2016. Donyms and compound relational nouns in nominal OpenIE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training (2018).
- Carsten Rother, Vladimir Kolmogorov, Victor S. Lempitsky, and Martin Szummer. 2007. Optimizing Binary MRFs via Extended Roof Duality. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.
- Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. 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)*, pages 728–737.
- Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2288–2299.
- Swarnadeep Saha, Harinder Pal, and Mausam. 2017a. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics.
- Swarnadeep Saha, Harinder Pal, and Mausam. 2017b. Bootstrapping for numerical OpenIE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323. Association for Computational Linguistics.
- Swarnadeep Saha et al. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299.

- Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. *CoRR*, abs/1707.07499.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics (ACL)*, 2017.
- Gabriel Stanovsky and Ido Dagan. 2016a. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page (to appear), Austin, Texas. Association for Computational Linguistics.
- Gabriel Stanovsky and Ido Dagan. 2016b. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas. Association for Computational Linguistics.
- Gabriel Stanovsky, Ido Dagan, and Mausam. 2015a. 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 303–308.
- Gabriel Stanovsky, Jessica Fidler, Ido Dagan, and Yoav Goldberg. 2016a. Getting more out of syntax with props. *CoRR*, abs/1603.01648.
- Gabriel Stanovsky, Jessica Fidler, Ido Dagan, and Yoav Goldberg. 2016b. Getting more out of syntax with PropS. *CoRR*, abs/1603.01648.
- Gabriel Stanovsky, Mausam, and Ido Dagan. 2015b. OpenIE 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)*, pages 303–308.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018a. 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), pages 885–895.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018b. Supervised open information extraction. In *Proceedings of The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, page (to appear), New Orleans, Louisiana. Association for Computational Linguistics.
- Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. 2018. 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*, pages 556–564.
- Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. 2019. 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)*, pages 3394–3403, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of Association for Computational Linguistics (ACL), 2019*.
- Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse Beam Search for Improved Description of Complex Scenes. In *AAAI Conference on Artificial Intelligence, 2018*, pages 7371–7379.
- Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. *ICLR*, abs/1511.06391.
- Xinyi Wang, Hieu Pham, Pengcheng Yin, and Graham Neubig. 2018. A tree-based decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4772–4777, Brussels, Belgium. Association for Computational Linguistics.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML), 2015*, pages 2048–2057.
- Junlang Zhan and Hai Zhao. 2020. Span Model for Open Information Extraction on Accurate Corpus. In *AAAI Conference on Artificial Intelligence, 2020*, pages 5388–5399.