Neural Architectures and Evaluation Protocols for Open Information Extraction

Thesis submitted by

Samarth Aggarwal
2016CS10395

under the guidance of

Prof. Mausam

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 Physics
IIT-Delhi, 110 016

Place: New Delhi
Date: 10th July 2020

ACKNOWLEDGEMENTS

TO BE ADDED

I thank IIT Delhi HPC Facility for compute resources.

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 OllIE, OpenIE-4 and OpenIE-5. Another wave of OpenIE systems that followed, comprised of neural approaches such as RnnOIE and Cui et al. (2018). However, the existing openie systems suffer from a wide range of problems. The rule-based systems suffered from cascading errors from a large number of components in succession. The existing neural OpenIE systems, although were able to solve some of these issues to a certain extent, were still 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.

Although human inspection revealed that the existing systems were not upto the mark, yet these systems scored high on the existing state-of-the-art OpenIE benchmarks such as OIE2016 (Stanovsky and Dagan, 2016). This means 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. (2018), 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. This approach also yields a coordination analyzer that significantly improves the yield of the MLIL model.

In the end, we analyse the milestones covered in the world of OpenIE and contribute some ideas for future research.

Contents

List of Tables

List of Figures

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

Sample Chapter

This document provides a simple template of how the provided `iitddiss.cls` LaTeX class is to be used. Also provided are several useful tips to do various things that might be of use when you write your thesis.

To compile your sources run the following from the command line:

```
% pdflatex thesis.tex
% bibtex thesis
% pdflatex thesis.tex
% pdflatex thesis.tex
```

Modify this suitably for your sources.

To generate PDF's with the links from the `hyperref` package use the following command:

```
% dvipdfm -o thesis.pdf thesis.dvi
```

1.1 Package Options

Use this thesis as a basic template to format your thesis. The `iitddiss` class can be used by simply using something like this:

```
\documentclass[PhD]{iitddiss}
```

To change the title page for different degrees just change the option from `PhD` to one of `MS`, `MTech` or `BTech`. The dual degree pages are not supported yet but should be quite easy to add. The title page formatting really depends on how large or small your thesis title is. Consequently it might require some hand tuning. Edit your version of `iitddiss.cls` suitably to do this. I recommend that this be done once your title is final.

To write a synopsis simply use the `synopsis.tex` file as a simple template. The synopsis option turns this on and can be used as shown below.

```
\documentclass[PhD,synopsis]{iitddiss}
```

Once again the title page may require some small amount of fine tuning. This is again easily done by editing the class file.

This sample file uses the `hyperref` package that makes all labels and references clickable in both the generated DVI and PDF files. These are very useful when reading the document online and do not affect the output when the files are printed.

1.2 Example Figures and tables

Fig. 1.1 shows a simple figure for illustration along with a long caption. The formatting of the caption text is automatically single spaced and indented. Table 1.1 shows a sample table with the caption placed correctly. The caption for this should always be placed before the table as shown in the example.

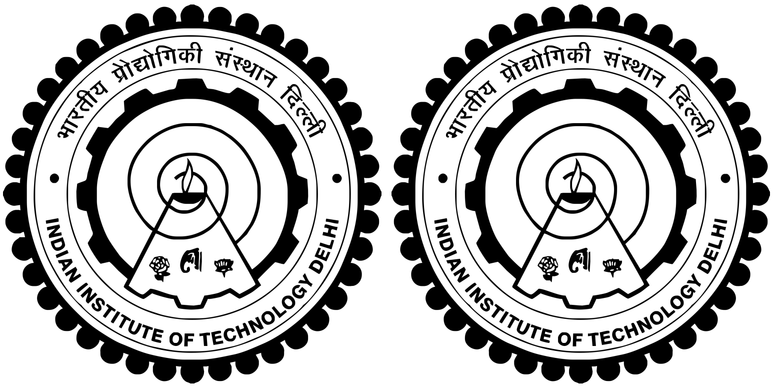
भारतीय प्रौद्योगिकी संस्थान दिल्ली
INDIAN INSTITUTE OF TECHNOLOGY DELHI
भारतीय प्रौद्योगिकी संस्थान दिल्ली
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Figure 1.1: Two IITD logos in a row. This is also an illustration of a very long figure caption that wraps around two two lines. Notice that the caption is single-spaced.

Table 1.1: A sample table with a table caption placed appropriately. This caption is also very long and is single-spaced. Also notice how the text is aligned.

x	x^2
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64

1.3 Bibliography with BIBTEX

I strongly recommend that you use BIBTEX to automatically generate your bibliography. It makes managing your references much easier. It is an excellent way to organize your references and reuse them. You can use one set of entries for your references and cite them in your thesis, papers and reports. If you haven't used it anytime before please invest some time learning how to use it.

I've included a simple example BIBTEX file along in this directory called `refs.bib`. The `iitddiss.cls` class package which is used in this thesis and for the synopsis uses the `natbib` package to format the references along with a customized bibliography style provided as the `iitd.bst` file in the directory containing `thesis.tex`. Documentation for the `natbib` package should be available in your distribution of LATEX. Basically, to cite the author along with the author name and year use `\cite{key}` where `key` is the citation key for your bibliography entry. You can also use `\citet{key}` to get the same effect. To make the citation without the author name in the main text but inside the parenthesis use `\citep{key}`. The following paragraph shows how citations can be used in text effectively.

More information on BIBTEX is available in the book by Lamport (1986). There are many references (Lamport, 1986; K, 2016) that explain how to use BIBTEX. Read the `natbib` package documentation for more details on how to cite things differently.

Here are other references for example. Ramachandran (2001) presents a Python based visualization system called MayaVi in a conference paper. Ramachandran et al. (2003) illustrates a journal article with multiple authors. Python (van Rossum et al., 1991–) is a programming language and is cited here to show how to cite something that is best identified with a URL.

1.4 Other useful LATEX packages

The following packages might be useful when writing your thesis.

- It is very useful to include line numbers in your document. That way, it is very easy for people to suggest corrections to your text. I recommend the use of the `lineno` package for this purpose. This is not a standard package but can be obtained on the internet. The directory containing this file should contain a lineno directory that includes the package along with documentation for it.
- The `listings` package should be available with your distribution of LATEX. This package is very useful when one needs to list source code or pseudo-code.
- For special figure captions the `ccaption` package may be useful. This is specially useful if one has a figure that spans more than two pages and you need to use the same figure number.

- The notation page can be entered manually or automatically generated using the `nomencl` package.

More details on how to use these specific packages are available along with the documentation of the respective packages.

Chapter 2

Introduction

2.1 Overview

2.1.1 What is OpenIE

Open Information Extraction (Open IE) refers to the task of forming relational tuples from sentences, without a fixed relation vocabulary Banko et al. (2007). Table 2.1 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 2.1: 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 preceeding 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.

2.1.2 Performance Gaps

There have been many Open IE systems till date such as TextRunner (Banko et al., 2007), ReVerb (Fader et al., 2011; Etzioni et al., 2011), OLLIE (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013), OpenIE 4 (Christensen et al., 2011; Pal and Mausam, 2016), OpenIE 5 (Saha et al., 2017; Saha and Mausam, 2018), PropS (Stanovsky et al., 2016), NST (Jia et al., 2018), Neural Open IE (Cui et al., 2018), 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, 2016).

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.

Traditionally, these systems have been evaluated over small manually curated gold datasets (e.g., Fader et al. (2011); Mausam et al. (2012)). 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 (2016), 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.

2.2 Problem Statement

It was clear that creation of a reliable OpenIE benchmark had to preceed 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 that could overcome that the problems of existing ones.

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., 2015; Mausam, 2016). Improving the performance of OpenIE systems would directly translate to an enhancement in their performance as well.

2.3 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.
-
-

add contributions from imojie and MLIL

Chapter 3

Literature Survey

3.1 Existing OIE Systems

Add lit.sur. of oie systems from imojie/mlil paper

3.1.1 Rule-Based Systems

ClausIE

PropS

OllIE

OpenIE - 4

OpenIE - 5

3.1.2 Neural Systems

Copy Attention Model

RnnOIE

3.2 Benckmarks

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

3.2.1 OIE2016

The first and the most prominent is OIE2016 Stanovsky and Dagan (2016). 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., 2018) 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 4.5 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 (2016) 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 Léchelle 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.

3.2.2 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.

3.2.3 Wire57

Finally, Wire57 Léchelle 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 4

CaRB - A Crowdsourced Benchmark for Open IE

4.1 Overview

4.1.1 Need for a New Benchmark

Traditionally, these systems have been evaluated over small manually curated gold datasets (e.g., Fader et al. (2011); Mausam et al. (2012)). 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 (2016), 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.

4.1.2 Establishing a new State-of-the-Art

rethink headings

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 4.3,4.4). 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`

4.2 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[2] 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.

Add ss to appendix

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. (2012). 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.

[2] Screenshots in Appendix

4.3 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 ate an apple and an orange.*	(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 4.1: 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 4.1 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 4.2 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 ate an apple.*	(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 4.2: 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.

4.4 Evaluation

4.4.1 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 4.3: Data quality using token-level match

	Precision	Recall	F1
OIE2016	0.67	0.51	0.57
CaRB	0.74	0.73	0.73

Table 4.4: Data quality using lexical match

Tables 4.3 and 4.4 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 4.5 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 (2016) 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	*Butters Drive in the Canberra suburb of Phillip is named in his honour .*
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	*It was only incidentally that economic issues appeared in nationalist political forms .*
OIE2016	(incidentally ; appeared ; economic issues ; nationalist political forms .)
CaRB	(economic issues ; appeared only incidentally in ; nationalist political forms)
Sent. # 3	*The main reason for this adoption over mainline gimp was its support for high bit depths which can be required for film work .*
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	*The number of ones equals the number of zeros plus one , since the state containing only zeros can not occur .*
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 4.5: Sample gold annotations for OIE2016 vs. CaRB

both agree on the grammatical head of all of their elements (predicate and arguments)".[3] The head match criterion is a much laxer scheme than ours and can explain the very high F1 score against their expert annotation.

4.4.2 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 4.6: Performance of Open IE systems on CaRB

Tally these values from IMoJIE table

[3] This 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 (2016), using the CaRB dataset and scorer. The p-r curves obtained using OIE2016 and CaRB are outlined in figures 4.1 (reproduced from Stanovsky et al. (2018)) and 4.2. Precision, recall and F1 scores (at max F1 point) and area under precision-recall curve are reported in Table 4.6. 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.

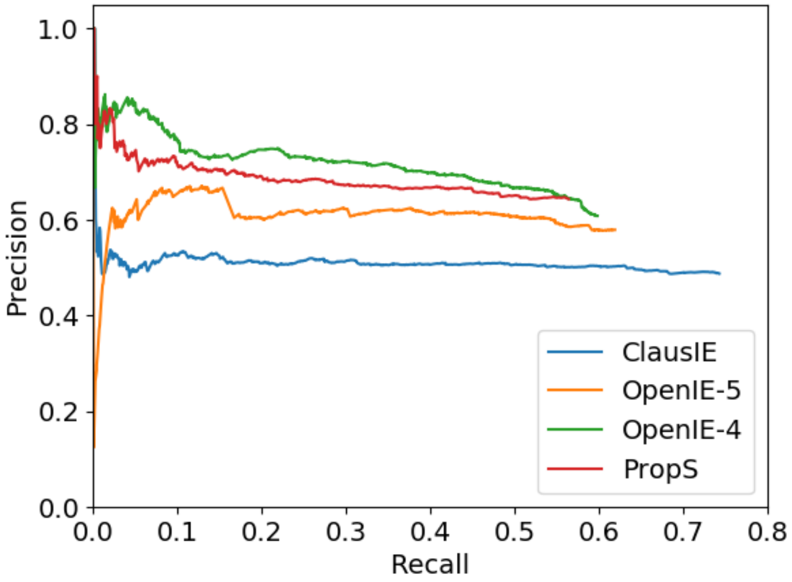

Figure 4.1: Comparison of Open IE systems using OIE2016

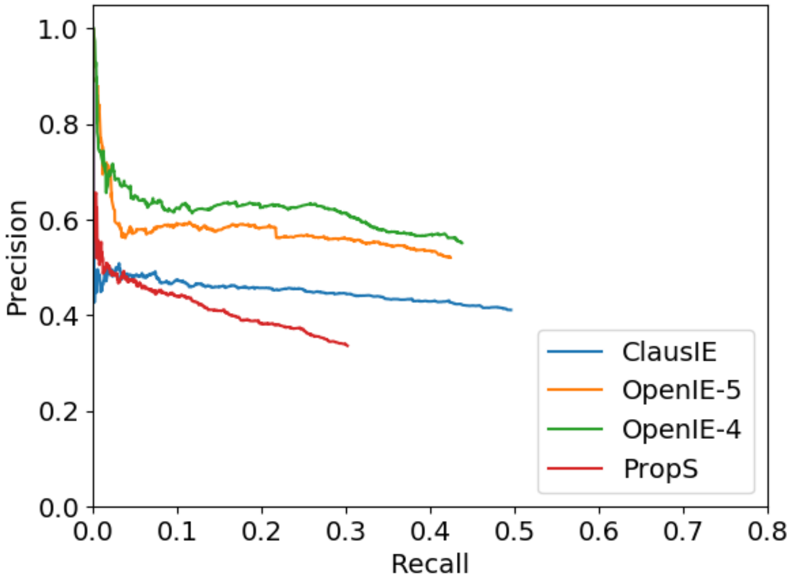

Figure 4.2: Evaluation of Open IE systems using CaRB

4.4.3 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.

4.5 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.

We expect that crowdsourced annotation will also be able to help the training of Open IE systems as it has helped their evaluation – we leave the creation of a suitably large crowdsourced training set for Open IE to future work.

link future work of carb to imojie, remove "future work" phrase

Chapter 5

IMoJIE - Iterative Memory Based Joint Open IE

Chapter 6

Remaining Problems

Chapter 7

Conjunction Splitting

Chapter 8

MLIL - Multi Level Iterative Labelling

Chapter 9

Milestones of OpenIE

Chapter 10

Future Ideas

Appendix A

A SAMPLE APPENDIX

Just put in text as you would into any chapter with sections and whatnot. Thats the end of it.

Bibliography

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. 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. 2011. 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.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of Association for Computational Linguistics (ACL), 2018*, pages 407–413.

Luciano Del Corro and Rainer Gemulla. 2013. 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.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. 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.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. 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.

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.

Sai Praneeth Reddy K. 2016. *LaTeX class for dissertations submitted to IIT-D.* Ph.D. thesis, Department of Computer Science and Engineering, IIT-Delhi, New Delhi – 110016.

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.

Leslie Lamport. 1986. *LaTeX: A document preparation system.* Addision-Wesley.

William Léchelle, Fabrizio Gotti, and Philippe Langlais. 2018. Wire57 : A fine-grained benchmark for open information extraction. *CoRR*, abs/1809.08962.

Mausam. 2016. 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. 2012. 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.

Harinder Pal and Mausam. 2016. Demonyms 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.

Prabhu Ramachandran. 2001. MayaVi: A free tool for CFD data visualization. In *4th Annual CFD Symposium.* Aeronautical Society of India. Software available at: http://mayavi.sf.net.

Prabhu Ramachandran, S. C. Rajan, and M. Ramakrishna. 2003. A fast, two-dimensional panel method. *SIAM Journal on Scientific Computing*, 24(6):1864–1878.

Guido van Rossum et al. 1991–. The Python programming language.

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. 2017. 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.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. *CoRR*, abs/1707.07499.

Gabriel Stanovsky and Ido Dagan. 2016. 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, Ido Dagan, and Mausam. 2015. 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 Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *CoRR*, abs/1603.01648.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. 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.

LIST OF PAPERS BASED ON THESIS

1. Authors.... Title... *Journal*, Volume, Page, (year).
2. Authors.... Title... *Journal*, Volume, Page, (year).