When to Use What: An In-Depth Comparative Empirical Analysis of OpenIE Systems for Downstream Applications

Kevin Pei[a], Ishan Jindal[b], Kevin Chen-Chuan Chang[a], Chengxiang Zhai[a], Yunyao Li[c]
[a]University of Illinois at Urbana-Champaign, [b]IBM Research, [c]Apple
{kspei2,kcchang,czhai}@illinois.edu, ishan.jindal@ibm.com,
yunyaoli@apple.com

Abstract

Open Information Extraction (OpenIE) has been used in the pipelines of various NLP tasks. Unfortunately, there is no clear consensus on which models to use for which tasks. Muddying things further is the lack of comparisons that take differing training sets into account. In this paper, we present an application-focused empirical survey of neural OpenIE models, training sets, and benchmarks in an effort to help users choose the most suitable OpenIE systems for their applications. We find that the different assumptions made by different models and datasets have a statistically significant effect on performance, making it important to choose the most appropriate model for one's applications. We demonstrate the applicability of our recommendations on a downstream Complex QA application.

1 Introduction

Open Information Extraction (OpenIE) is the task of extracting relation tuples from plain text (Angeli et al., 2015). In its simplest form, OpenIE extracts information in the form of tuples consisting of *subject*(S), *predicate*(P), *object*(O), and any *additional arguments*(A). OpenIE is an open domain, intended to be easy to deploy in different domains without fine-tuning, with all relations extracted regardless of type. The increasing availability of semi-automatically generated training datasets (Cui et al., 2018) as well as significant advances in deep learning techniques have led to the development of state-of-the-art neural models (Cui et al., 2018; Garg and Kalai, 2018).

Since its introduction in Etzioni et al. (2008), OpenIE has attracted a large amount of attention by the research community as a tool for a wide range of downstream NLP tasks (Mausam, 2016). However, there is no real consensus on which OpenIE model is best for each application. One example of this lack of consensus in summarization, where different papers use OLLIE (Christensen et al., 2014), MinIE (Ponza et al., 2018), and Stanford CoreNLP (Cao et al., 2018; Zhang et al., 2021) for extraction. Different applications may also have different requirements.As an example, choosing a model that assumes all relations only have a subject and object may not be suitable for event schema induction since that excludes any event schemas with more than two entities. The papers that introduce new OpenIE models and datasets do not specify how downstream applications would be impacted by the different assumptions those models make about extracted relations.

Sentence	
Bill Gates, former CEO of Microsoft, is a Harvard dropout.	
OpenIE Extractions	
(Bill Gates, was, former CEO of Microsoft) (Bill Gates, is, a Harvard dropout)	
Applications	
QA	Who was former CEO of Microsoft? Where did Bill Gates dropout of?
Slot Filling	(?, was, former CEO of Microsoft) (?, is, a Harvard dropout)

Table 1: Sample relation tuples and examples of how different applications use OpenIE extractions.

We find that prior OpenIE surveys are also insufficient to find the best OpenIE model for a given application. The only previous application-focused OpenIE survey we found was Mausam (2016). However, this survey does not identify the desired properties of OpenIE for those applications or provide an empirical comparison of OpenIE systems. Glauber and Claro (2018), Claro et al. (2019), and Zhou et al. (2022) also do not provide an empirical application-focused survey.

Another obstacle is the lack of apples-to-apples comparisons between OpenIE models. Comparisons should keep the training set, benchmark, and evaluation metric constant to eliminate confounders. Unfortunately, the papers that intro-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 929–949
July 9-14, 2023 ©2023 Association for Computational Linguistics

	Question Answering	Slot Filling	Event Schema Induction	Summarization	Knowledge Base Population
HR: Higher Recall	✓	✓	✓	✓	✓
HP: Higher Precision		✓		✓	
N-ary: N-ary Relation Extraction	✓	✓	✓		
IN: Inferred Relation Extraction	✓	✓	✓	✓	
FE: Fast Extraction					✓

Table 2: Properties explicitly mentioned in application papers as motivation for choosing a particular OpenIE model or as a way to improve performance within a case study. There are additional desired properties we omit that no existing OpenIE models or datasets possess, such as the canonicalization of extracted relations and the ability to extract relations from imperative sentences (Fader et al., 2013; Khot et al., 2017; Zhang et al., 2021).

duce new OpenIE models often do not provide this apples-to-apples comparison. For example, CopyAttention (Cui et al., 2018), SpanOIE (Zhan and Zhao, 2020), IMoJIE (Kolluru et al., 2020b), and OpenIE6 (Kolluru et al., 2020a) all compare their model to models trained on different training sets. OpenIE6 reports performance on the WiRe57 benchmark which Multi^{2}OIE (Ro et al., 2020) does not, but Multi^{2}OIE reports performance on the ReOIE2016 benchmark which OpenIE6 does not. Because the training set can greatly affect the performance of a neural model, we focus on selecting both the appropriate OpenIE model and training set, which we refer to as an *OpenIE System*.

To resolve our lack of understanding, we focus on one particular question: *How do I choose a particular OpenIE system for a given application?* Different implicit assumptions about OpenIE may have a significant impact on the performance of downstream applications such as the assumptions that all relations are verb-based (Zhan and Zhao, 2020) or that all relations have only a subject and object (Kolluru et al., 2020b). To answer this question an apples-to-apples comparison must be conducted for different application settings.

Because it is impractical to find the best model for every application given the many possible applications of OpenIE, we instead characterize applications based on what properties they desire from OpenIE such as the desire for N-ary relation extraction by event schema induction. We provide an extensive apples-to-apples comparison of neural OpenIE models such that a practitioner can utilize our practical observations to effectively select a neural OpenIE model and training set for their downstream application. Finally, we apply our recommendations to a downstream Complex QA task. In summary, our contributions are as follows:

- We propose a taxonomy that covers OpenIE training sets, benchmarks, and neural models.
- We present an extensive empirical comparison of different models on different datasets with recommendations based on the results.
- We perform a case study on Complex QA to show the efficacy of our recommendations.

To the best of our knowledge, our survey is the only application-focused empirical survey on OpenIE datasets, metrics, and neural OpenIE models.

2 Motivating Applications

In this section, we identify the properties of OpenIE desired by 5 downstream applications: *Slot Filling*, *Question Answering (QA)*, *Summarization*, *Event Schema Induction*, and *Knowledge Base Population*. We survey how OpenIE is used and the properties explicitly desired by papers corresponding to the application, either as motivation for choosing a given OpenIE model or within a case study as a property that would improve performance.

The desired properties we observe are **Higher Recall**, **Higher Precision**, **N-ary Relation Extraction**, **Inferred Relation Extraction**, and **Fast Extraction**. We define an "Inferred Relation" (*IN*) to be a relation that contains words that are not in the original sentence. For example, given the sentence "*Bill Gates, former CEO of Microsoft, is a Harvard dropout*", the relation *(Bill Gates, was, former CEO of Microsoft)* can be inferred even though "was" is not in the original sentence. We define an "N-ary Relation" (*N-ary*) to be a relation with more arguments than just (subject, predicate, object). For example, the relation *(Alice, went, to the store, today)* has an additional argument *today*. Table 2 provides a summary the explicitly desired properties of downstream applications.

	Dataset	Creation Method	Source	#Extractions	#IN	#N-ary
Training Sets	SpanOIE	Weak Labeling	Wikipedia	2,175K	2K	231K
	OIE4	Weak Labeling	Wikipedia	181K	3K	34K
	IMoJIE	Weak Labeling	Wikipedia	215K	3K	0
	LSOIE	Weak Labeling	QA-SRL 2.0 Wikipedia, Science	101K	0	32K
Test Sets	OIE2016	Weak Labeling	QA-SRL	1,730	359	708
	WiRe57	Manual Annotation	Wikipedia and Newswire	343	173	79
	ReOIE2016	Manual Annotation	OIE2016	1,508	155	611
	CaRB	Crowdsourced Annotation	OIE2016	5,263	736	683
	LSOIE	Weak Labeling	QA-SRL 2.0 Wikipedia, Science	22,376	0	4,920

Table 3: Comparison of the attributes of different datasets. #Extractions: Number of Extractions, #IN : Number of inferred relations, #N-ary: Number of N-ary Relations.

Slot Filling Slot Filling is a task where an incomplete tuple must be completed using information from a given corpus (Chen et al., 2019). For example, the incomplete tuple *(Obama, born in, ?)* must be completed as *(Obama, was born in, Honolulu)* using information from the corpus. OpenIE can be used to extract complete tuples which fill slots in an incomplete tuple using entity linking. Soderland et al. (2013), Angeli et al. (2015), Soderland et al. (2015b), and Soderland et al. (2015a) take advantage of how correct relations often appear multiple times to match empty slots to the highest precision OpenIE tuple. They state in their case studies they would benefit from *IN* extraction and Soderland et al. (2015b) and Soderland et al. (2015a) state they would benefit from *N-ary* extraction. These two properties allow more relation surface forms to be extracted, which increases the chance an incomplete tuple can be linked to a complete tuple.

Question Answering We focus on two subtasks of Question Answering (QA) that utilize OpenIE: Open-domain QA (OpenQA) and Complex QA. OpenQA involves answering questions given a large database (Fader et al., 2014a). Complex QA involves using information from multiple sentences to find answers and requires inferring relationships between multiple entities (Chali et al., 2009). Fader et al. (2013, 2014b), Yin et al. (2015), and Clark et al. (2018) are OpenQA methods that use retrieval-based methods to match OpenIE extractions to questions. By rewriting queries into incomplete tuples, such as rewriting "Where was Obama born?" into *(Obama, born in, ?)*, it is possible to use extracted relations to answer queries by filling in the missing slots in the query. For ComplexQA, Khot et al. (2017) and Lu et al. (2019) generate graphs from extracted relation tuples, then reason over these graphs to answer questions. In all QA applications surveyed, high recall (*HR*) is desired, with Lu et al. (2019) using a custom OpenIE method specifically for higher recall. Yin et al. (2015)'s case studies state that *N-ary* would be beneficial while Lu et al. (2019) uses a custom OpenIE method that supports *IN*.

Summarization OpenIE addresses the problems of redundancy and fact fabrication in summarization. Redundancy is when a fact is repeated multiple times in the summary. To combat redundancy, OpenIE is used to ensure that the generated summary does not have repeated relations (Christensen et al., 2014; Zhang et al., 2021). Fact fabrication is when a fact that is not supported by the text being summarized is in the summary. To combat fact fabrication, OpenIE is used to ensure that the generated summary only contains relations from the original text (Cao et al., 2018; Zhang et al., 2021). In summarization tasks, *HR* is useful to ensure summaries contain all information, with Ponza et al. (2018) citing greater diversity of extractions as a way to improve performance. high precision (*HP*) is also desired by Zhang et al. (2021) in order to reduce redundant extractions.

Event Schema Induction Event Schema Induction is the automatic discovery of patterns that indicate events, agents, and the agents' roles within that event. Extracted relations can be used to find surface forms of events, with redundant tuples being used to induce event schemas. The open nature of OpenIE allows for events to be found regardless of the domain or surface form. *HR* is useful for Event Schema Induction for the same reason it is useful for Slot Filling: finding more surface forms allows for more event schemas to be induced (Balasubramanian et al., 2013; Romadhony et al., 2019; Sahnoun et al., 2020). Sahnoun et al. (2020) also specifically desire *IN* so that more event schemas can be learned, while Balasubramanian et al. (2013) state that *N-ary* would improve performance.

Knowledge Base Population The relations extracted by OpenIE can be used to automatically populate knowledge bases (KBs), creating new nodes and edges. Muhammad et al. (2020) and Kroll et al. (2021) use learning-based OpenIE models because of their ability to generalize to unseen relations and achieve *HR*. Kroll et al. (2021) also explicitly chooses Stanford CoreNLP and OpenIE6 for their fast extraction times (*FE*).

3 OpenIE Datasets

In this section, we discuss the differences between different OpenIE training sets and benchmarks and their shortcomings. We provide statistics about different datasets in Table 3.

3.1 Training Datasets

Given how data-hungry deep learning models are and how costly it is to manually label OpenIE datasets, most OpenIE training sets are weakly labeled using high confidence extractions from prior OpenIE models.

CopyAttention (Cui et al., 2018), **SpanOIE** (Zhan and Zhao, 2020), and **OIE4** (Kolluru et al., 2020b) are training sets consisting of high confidence OpenIE4 extractions from Wikipedia.

SpanOIE includes extractions of all confidences unlike CopyAttention and OIE4 which only contain extractions above a certain confidence threshold.

The **IMoJIE** dataset (Kolluru et al., 2020b) attempts to get higher quality labels by combining Wikipedia extractions from OpenIE4, ClausIE, and RNNOIE, using a common scoring metric to combine extractions and filter out repeated extractions.

The **LSOIE** training set (Solawetz and Larson, 2021) is composed of automatically converted Semantic Role Labeling (SRL) extractions with high inter-annotator agreement from the Wikipedia and Science domain of the crowdsourced QA-SRL Bank 2.0 dataset. Because this dataset is derived from SRL, all relations are assumed to be verb-based and none are inferred.

Issues with existing training sets

Current OpenIE training sets are limited to Wikipedia and Science domains, which may not generalize to certain other domains. Additionally, all OpenIE training sets are weakly labeled, leading to noisy labels which may limit the capabilities of neural OpenIE models. For example, there are instances in LSOIE where the gold relation does not contain a negation it should, resulting in a completely different semantic meaning. It is an open question of how much noise exists within these training sets.

3.2 Benchmarks

OIE2016 (Stanovsky and Dagan, 2016) is a benchmark for OpenIE automatically derived from the crowdsourced QA-SRL dataset annotated on PropBank and Wikipedia sentences.

WiRe57 (Léchelle et al., 2018) consists of expert annotations for 57 sentences.

CaRB (Bhardwaj et al., 2019) uses crowdsourcing to re-annotate the sentences in the OIE2016 benchmark.

ReOIE2016 (Zhan and Zhao, 2020) uses manual annotation to re-annotate OIE2016 to attempt to resolve problems arising from incorrect extraction.

LSOIE (Solawetz and Larson, 2021) has benchmarks derived using the same sources and rules as the training sets.

BenchIE (Gashteovski et al., 2021) is derived from CaRB and is based on the idea that extracted relations need to exactly match at least one relation out of a "fact set" of semantically equivalent manually annotated gold standard relations.

Are existing benchmarks sufficient?

Given how the OIE2016 benchmark has been re-annotated three times, there is no real consensus on how to annotate OpenIE. For example, CaRB labels prepositions as part of the object and not the predicate, but OIE2016 and ReOIE2016 do not. As a result, it is very difficult for a single model to do well on all benchmarks because each one makes different assumptions. Although there are common principles that guide OpenIE labeling, namely *Assertedness*, *Minimal Propositions/Atomicity*, and *Completeness and Open Lexicon* (Stanovsky and Dagan, 2016; Léchelle et al., 2018; Bhardwaj et al., 2019), these principles are vague enough to be interpreted in different ways.

4 Evaluation Metrics

In this section, we describe the different evaluation metrics used to evaluate OpenIE models and discuss their shortcomings.

OIE2016 introduces *lexical matching*, which treats evaluation as a binary classification task. A predicted relation is matched to a gold standard relation if the heads of the predicate and all arguments

Model	Problem Formulation	N-ary	IN
SpanOIE	Labeling	✓	
IMoJIE	Generation		
Multi^{2}OIE	Labeling	✓	
IGL-OIE	Labeling		✓
CIGL-OIE	Labeling		✓
OpenIE6	Labeling		✓
DetIE	Labeling		✓

Table 4: Comparison of neural OpenIE models.

are the same.

WiRe57 and **CaRB** use *word-level matching*, which calculate recall and precision based on the proportion of matching tokens in the predicted and gold standard relations. WiRe57 gives a greater penalty to recall than CaRB if there are fewer predicted relations than gold standard relations.

BenchIE uses *sentence-level matching*, which requires an exact match of the predicate and arguments to a relation in the fact set. Because of BenchIE's reliance on fact sets which other benchmarks lack, the BenchIE metric is only compatible with BenchIE and no other metrics can be used with the BenchIE dataset. As a result, an apples-to-apples comparison of the BenchIE dataset and metric with other datasets and metrics is not possible, so we do not report performance on BenchIE.

Is AUC a useful metric?

When comparing OpenIE systems, we place a greater emphasis on F1 score than AUC. The original implementations of CaRB, OIE2016, and WiRe57 use the trapezoidal rule to calculate AUC which leads to inflated AUC scores for certain systems without low recall points. As a result, we consider the highest F1 score on the PR curve to be a better metric than AUC.

Are existing metrics sufficient?

All existing OpenIE metrics are lexical metrics, and lexical metrics are merely a proxy for comparing the semantic meanings of the predicted relations with the gold standard relations. For instance, existing OpenIE metrics only give small penalties for omitting negations from predicted relations, even though this changes the semantic meaning. This issue can be also observed in lexical metrics used for summarization (Saadany and Orasan, 2021).

5 Neural OpenIE Models

In this section, we describe neural OpenIE models and the properties and assumptions they make that set them apart. Neural OpenIE models can be categorized based on how they formulate the OpenIE problem: as a text generation or labeling problem. We provide overviews of the models in Table 4.

5.1 Generative Problem Formulation

Generative OpenIE models cast OpenIE as a sequence-to-sequence problem, taking the sentence as input and attempting to generate all relations in the sentence as output. The generative models we survey rely on a copy mechanism to copy vocabulary from the original sentence, meaning they can not extract *IN* relations.

CopyAttention (Cui et al., 2018) generates extractions using GloVe embeddings and a 3-layer stacked Long Short-Term Memory (LSTM) as the encoder and decoder.

IMoJIE (Kolluru et al., 2020b) builds upon CopyAttention by using BERT embeddings and introducing *iterative extraction* to combat repeated extractions. *Iterative extraction* is repeated extraction from the same sentence with previously extracted relations appended to the end so the model can identify what relations have previously been extracted.

5.2 Labeling Problem Formulation

Labeling OpenIE models cast OpenIE as a sequence labeling problem, usually using a BIO tagging scheme to label tokens in the sentence. They can be subdivided into Piecewise and Holistic Labeling models.

5.2.1 Piecewise Labeling

Piecewise labeling models first label predicates and then label arguments for each extracted predicate to extract relation tuples.

RnnOIE (Stanovsky et al., 2018) is a bi-directional LSTM (BiLSTM) transducer inspired by SRL that uses BIO tags.

SpanOIE (Zhan and Zhao, 2020) is also based on SRL, using a BiLSTM to perform span classification instead of BIO tagging. In span classification, spans of tokens of varying length are classified as parts of the relation instead of individual tokens. Span classification allows for the use of span features, which can be richer than word-level features.

Multi^{2}OIE's (Ro et al., 2020) novelty is multi-head attention and BERT embeddings. After labeling the predicates, multi-head attention is used between the predicate and the rest of the sentence to label the arguments.

MILIE (Kotnis et al., 2021) introduces *iterative prediction*, the process of extracting one argument of the relation tuple at a time, for multilingual OpenIE. Extraction can be performed predicate, subject, or object first, in case other languages benefit from different extraction orders.

Uniquely, piecewise labeling models label all predicates in a sentence simultaneously and assume that for each predicate, there is only one set of arguments. This means that they can not extract multiple relations that share the same predicate, unlike generative and holistic labeling models.

5.2.2 Holistic Labeling

Holistic labeling models label predicates and arguments simultaneously.

OpenIE6 (Kolluru et al., 2020a) introduces grid labeling, constraint rules, and conjunction rules. Grid labeling is the simultaneous extraction of multiple relations from a sentence. Constraint rules penalize certain things like repeated extractions or not extracting a relation for a head verb. Conjunction rules split relations containing conjunctions into two separate relations. IGL-OIE is the first stage, using only grid labeling; CIGL-OIE is the second stage, adding in constraint rules; OpenIE6 is the final stage, adding conjunction rules.

DetIE (Vasilkovsky et al., 2022) uses ideas from single-shot object detection to make predictions more quickly than previous methods. Labeling models generally can not label tokens that are not in the original sentence, meaning they can not extract *IN* relations. However, the more recent models IGL-OIE, CIGL-OIE, OpenIE6, and DetIE explicitly add "be", "of", and "from" to the end of sentences to allow for the extraction of inferred relations with those predicates.

5.3 Model Hyperparameters

The sensitivity to hyperparameters of the models we survey is unclear. Of the works we survey, Multi^{2}OIE and OpenIE6 describe how they perform hyperparameter tuning and provide the hyperparameters they tested. SpanOIE, IMoJIE, and DetIE do not provide details of how they obtained the hyperparameters they use. None of these works provide an in-depth analysis of how the performance was affected by different hyperparameter values. As a result, we perform our own sensitivity analysis using Multi^{2}OIE. The results of this analysis can be found in Appendix B.

In our own experiments, we observed only minor increases in performance from changing the hyperparameters in a few cases. On average, the performance changes were negligible. When making recommendations, we consider the performance over many different combinations of model, training, and test set. Minor differences in a handful of cases do not impact our overall conclusions. As a result, we use the default hyperparameters used by Ro et al. (2020) for Multi^{2}OIE. Because other models did not report any particular sensitivity to hyperparameters, we generalize this result to all models we use and use the final set of hyperparameters those authors use.

5.4 Existing Model Limitations

Models are often developed with specific datasets in mind. Some papers introducing new models also introduce new training sets such as CopyAttention (Cui et al., 2018), SpanOIE (Zhan and Zhao, 2020), and IMoJIE (Kolluru et al., 2020b) which may influence model assumptions. SpanOIE also introduces its own manually annotated benchmark, which may have informed the assumptions SpanOIE makes. The lack of consensus on how to label OpenIE makes it difficult to perform apples-to-apples comparisons because certain models can not extract some relations due to the assumptions they make.

OpenIE has also largely been limited to English. MILIE makes assumptions that allow for different extraction methods depending on the language, but other OpenIE models that support multilingual extraction largely treat extraction from other languages the same as extraction from English. Multilingual OpenIE remains an open field of study.

6 Experiments

In this section, we describe how we compare OpenIE models and datasets for the sake of recommendation. To find the best system for different applications, we test whether the properties of OpenIE models and training sets have a statistically significant effect on accuracy in test sets with corresponding properties.We are also interested in how the choice of model affects efficiency in order to satisfy the fast extraction property (*FE*). We answer the following questions:

R1: How does whether a model supports N-ary relation (*N-ary*) extraction and whether the training set contains *N-ary* affect the F1 score of a model on test sets with or without *N-ary*?

R2: How does whether a model supports inferred relation (*IN*) extraction and whether the training set contains *IN* affect the F1 score of a model on test sets with or without *IN*?

R3: How does the model type affect efficiency as measured by the number of sentences processed per second (Sen./Sec)?

6.1 Experimental Setup

Models: We compare *SpanOIE*, *IMoJIE*, *Multi^{2}OIE*, the 3 stages of OpenIE6: *IGL-OIE*, *CIGL-OIE*, and *OpenIE6*, and *DetIE*. For each model, we train them with their paper's original dev set and their original hyperparameters. We run all experiments on a Quadro RTX 5000 GPU.

Training Datasets: We train models on the *SpanOIE*, *OIE4*, *IMoJIE*, and *LSOIE* training sets. We combine the Science and Wikipedia domain for both the training and benchmark of LSOIE, ensuring there are no duplicate sentences from overlapping sentences in the domains. Due to the input structure of SpanOIE and Multi^{2}OIE, they can not be trained on training datasets with inferred relations. Subsequently, we remove any inferred relations from the training sets of those models. Similarly, as IMoJIE , OpenIE6, and DetIE can not extract N-ary relations, we convert all N-ary relations in the training set into binary relations by moving additional arguments into the object. For instance, the relation (Alice, went, to the store, today) is converted into (Alice, went, to the store today). Inferred and N-ary relations were not removed from the gold standards of the test sets.

Benchmarks: We evaluate all the models on the publicly available English benchmarks *OIE2016*, *WiRe57*, *ReOIE2016*, *CaRB*, and *LSOIE*.

Evaluation Metrics: We use *OIE2016*'s, *WiRe57*'s, and *CaRB*'s metrics for evaluation. We perform student's t-test between OpenIE system, test set, and evaluation metric configurations to answer **R1**, **R2**, and **R3**. For **R1** and **R2** the t-scores are computed using the per-sentence F1 scores of each method. For **R3** the t-scores are computed using the mean sentences per second for each training set and test set combination for a given model.

7 Results

In this section, we perform an apples-to-apples comparison among different OpenIE systems to determine the SoTA OpenIE model and the best general-purpose OpenIE training dataset.

Best OpenIE Model We compare the different models on different evaluation metrics averaged across different training and test sets in Table 5. We observe that across all evaluation metrics Multi^{2}OIE and CIGL-OIE have the highest or second highest F1 score. We also observe that IGL-OIE and CIGL-OIE are the most efficient models.

Best OpenIE Training Set Because performance on a test set is also greatly dependent on the training set depending on the domain and generation methods, we determine the best training set for each test set. In Table 6, we compare different training and test set combinations with different evaluation metrics averaged across models. We observe that the models trained on LSOIE perform best on the OIE2016 and LSOIE test sets. This is because the LSOIE training set and the OIE2016 and LSOIE test sets are derived from different versions of QA-SRL and generated using the same rules. On the WiRe57, ReOIE2016, and CaRB test sets, we observe that the models trained on the OIE4 and SpanOIE training sets generally perform the best. It is likely because the OIE4 and SpanOIE training sets contain both *N-ary* and *IN* relations like the WiRe57, ReOIE2016, and CaRB test sets while LSOIE and IMoJIE don't.

Of the two models with the highest average CaRB F1 scores, Multi^{2}OIE and CIGL-OIE, Multi^{2}OIE has higher average precision while CIGL-OIE has higher average recall. CIGL-OIE tends to extract longer objects than Multi^{2}OIE as seen in Table 7, which may explain this difference. Overall, OpenIE models have the poorest performance when extracting the object, which may be due to the variance in object length from additional arguments compared to the subject and predicate.

7.1 Research Questions

To answer our research questions, we perform student's t-test using the CaRB F1 scores of the highest scoring model, training set, and test set combinations for each setting. We perform comparisons of OpenIE systems, where one aspect (model or training set) is changed and the other aspects are kept constant. Then, we choose the test set and evaluation metric for the two settings that results in the highest t-score between methods.

For **R1**, we conclude (1) regardless of training set, the best *N-ary* models perform better than the best non-*N-ary* models; (2) regardless of the model, training on the best *N-ary* training sets results in higher performance than training on the best non-

Model	Sen./Sec.	CaRB			WiRe 57		
		P	R	F1	P	R	F1
SpanOIE	13.40	0.474	0.464	0.433	0.474	0.374	0.375
IMoJIE	2.07	0.598	0.431	0.488	0.598	0.355	0.428
Multi^{2}OIE	29.22	**0.626**	0.501	**0.552**	**0.624**	0.419	**0.488**
IGL-OIE	**84.07**	0.574	0.442	0.497	0.574	0.365	0.434
CIGL-OIE	68.80	0.490	**0.531**	0.503	0.489	0.429	0.442
OpenIE6	28.36	0.394	0.518	0.438	0.394	**0.463**	0.413
DetIE	29.06	0.603	0.436	0.502	0.603	0.353	0.435

Table 5: Performance of different models with different evaluation metrics averaged across training and test data.

Training Set	Test Set	CaRB			WiRe 57		
		P	R	F1	P	R	F1
SpanOIE	OIE2016	0.495	0.491	0.478	0.493	0.410	0.433
OIE4	OIE2016	0.541	0.487	0.510	0.540	0.404	0.458
LSOIE	OIE2016	**0.629**	**0.537**	**0.569**	**0.629**	**0.443**	**0.509**
IMoJIE	OIE2016	0.469	0.433	0.424	0.468	0.363	0.381
SpanOIE	WiRe57	0.420	0.372	0.386	0.423	0.199	0.263
OIE4	WiRe57	**0.473**	**0.378**	**0.420**	**0.472**	0.211	**0.290**
LSOIE	WiRe57	0.355	0.210	0.261	0.355	0.127	0.184
IMoJIE	WiRe57	0.436	0.364	0.378	0.434	**0.215**	0.264
SpanOIE	ReOIE2016	0.650	**0.625**	**0.618**	0.650	**0.612**	**0.612**
OIE4	ReOIE2016	**0.725**	0.568	0.606	**0.725**	0.555	0.599
LSOIE	ReOIE2016	0.632	0.525	0.562	0.632	0.513	0.555
IMoJIE	ReOIE2016	0.620	0.570	0.560	0.619	0.551	0.548
SpanOIE	CaRB	0.539	0.440	0.472	0.535	0.306	0.377
OIE4	CaRB	**0.606**	**0.446**	**0.512**	**0.606**	**0.311**	**0.408**
LSOIE	CaRB	0.539	0.344	0.415	0.539	0.252	0.337
IMoJIE	CaRB	0.539	0.414	0.446	0.536	0.300	0.354
SpanOIE	LSOIE	0.470	0.561	0.501	0.470	0.516	0.479
OIE4	LSOIE	0.505	0.558	0.529	0.505	0.512	0.505
LSOIE	LSOIE	**0.658**	**0.676**	**0.659**	**0.658**	**0.622**	**0.629**
IMoJIE	LSOIE	0.441	0.492	0.444	0.441	0.460	0.431

Table 6: Performance of different training and test sets averaged across models.

Sentence	According to the 2010 census, the population of the town is 2,310.
Multi^{2}OIE	(the population of the town; is; 2,310)
CIGL-OIE	(the population of the town; is; According to the 2010 census, 2,310)

Table 7: A demonstration that CIGL-OIE tends to extract longer objects than Multi^{2}OIE. Both are trained on SpanOIE. The sentence is from the CaRB test set.

N-ary training sets. Therefore **if an application benefits from *N-ary*, then the best OpenIE system should include either a *N-ary* model, *N-ary* training set, or both**, with both being preferred.

For **R2**, we conclude that (1) *IN* models are better than non-*IN* models when there is either a *IN* training and *IN* test set, or a non-*IN* training and non-*IN* test set; (2) *IN* training sets are better than non-*IN* training sets when there is an *IN* model and *IN* test set. Therefore **if an application benefits from *IN*, then the chosen training set and model should either both be *IN* or both be non-*IN*.**

For **R3**, we compare the efficiency of the sole generative model, IMoJIE, to the efficiency of every other model. We observe that every other model is faster than IMoJIE and the difference is statistically significant. This matches expectations, since it has been previously shown that IMoJIE is slower than other OpenIE models (Kolluru et al., 2020a).Therefore **if an application is concerned about efficiency, then the chosen OpenIE model should not be a generative model.**

8 A Case Study: Complex QA

To verify our recommendations, we perform a case study using QUEST (Lu et al., 2019), a Complex QA method that uses OpenIE to extract entities and predicates from the question and from documents to generate knowledge graphs. The nodes are entities derived from subjects and objects, while the edges are predicates. The knowledge graph is matched to the entities in the question and traversed to find potential answers. Because more extractions result in a larger knowledge graph, QUEST benefits from *HR* which the authors use their own rule-based OpenIE method to achieve.

8.1 Experimental Setup

To test our recommendations, we replace the OpenIE method used by the authors with Multi^{2}OIE trained on SpanOIE, CIGL-OIE trained on OIE4, and OpenIE6 trained on OIE4. We chose these models and training sets because they have the highest overall CaRB recall and F1 scores.

One caveat is that in order for QUEST to connect entities from multiple sentences, they must have the same surface form. Because OpenIE methods often extract long subjects and objects that include adjectives and modifiers, if the subject or object of an extraction contains entities extracted by QUEST,

OpenIE	Questions	Documents	MRR	P@1	Hit@5
QUEST	CQ-W	Top 10	0.132	0.080	0.167
CIGL-OIE	CQ-W	Top 10	**0.111**	**0.060**	**0.167**
OpenIE6	CQ-W	Top 10	0.104	0.060	0.147
Multi2OIE	CQ-W	Top 10	0.094	0.053	0.140

Table 8: Performance of QUEST using different OpenIE methods on the CQ-W dataset using the Top 10 Google documents.

we add additional relations using those entities. For example, in the sentence "Hector Elizondo was nominated for a Golden Globe for his role in Pretty Woman," QUEST may extract the entities "Hector Elizondo," "Golden Globe," and "Pretty Woman." If an OpenIE method were to extract the triple *("Hector Elizondo", "was nominated", "for a Golden Globe for his role in Pretty Woman")*, we would add the additional extractions *("Hector Elizondo", "was nominated", "Golden Globe")* and *("Hector Elizondo", "was nominated", "Pretty Woman")*. QUEST also replaces pronouns with the entities they refer to because nodes in the knowledge graph can not be made using pronouns.We replace pronouns using the same method QUEST does before running any OpenIE method.

We run QUEST using the CQ-W question set and search for answers in the Top-10 Google document set used in their paper. Because CIGL-OIE has the highest CaRB recall and OpenIE6 has the highest WiRe57 recall, we expect that using either of them will result in higher downstream performance than using Multi^{2}OIE.

8.2 Evaluation

We compare the Mean Reciprocal Rank (MRR), Precision@1 (P@1), and Hit@5 for each OpenIE model. The results of our case study are summarized in Table 8. We observe higher performance of CIGL-OIE and OpenIE6 than Multi^{2}OIE on QUEST, which matches our expectations based on the higher recall of CIGL-OIE and OpenIE6 and the desired property of *HR* but not *HP* for QA. Our case study demonstrates the applicability of our empirical study to the use of OpenIE methods in downstream applications.

An important note is that oftentimes a great deal of pre- and post-processing is necessary to adapt OpenIE for different downstream applications. Removing pronouns and adding additional entity-based extractions was necessary to achieve reasonable performance in QUEST. Even after modifying Multi^{2}OIE, CIGL-OIE, and OpenIE6 in this way, their performance is less than the original performance of QUEST. As a result, it is important to not just consider the performance and properties of OpenIE models, but also how to adapt models to their specific needs.

9 Challenges and Future Directions

Even with the introduction of neural models, OpenIE systems still have significant room for improvement. In Table 2 we state that canonicalizing extractions is desired by QA while extracting from imperative sentences is desired by both QA and summarization, but no existing model or dataset addresses these properties. In sections 3.1 and 3.2 we note the lack of consensus on how to label OpenIE and the issues with weak labeling. Existing metrics also have issues with semantic meaning as discussed in section 4, which is exacerbated by errors caused by weak labeling. The lack of consensus in how to label OpenIE relations results in a diverse set of models as we discuss in section 5.4. The different assumptions these models make are also largely constrained to English syntax, leaving future work in multilingual OpenIE open.

10 Conclusion

In this paper, we presented an application-focused empirical comparison of recent neural OpenIE models, training sets, and benchmarks. Our experiments showed that the different properties of OpenIE models and datasets affect the performance, meaning it is important to choose the appropriate system for a given application and not just choose whatever model is state-of-the-art. We hope that this survey helps users identify the best OpenIE system for their downstream applications and inspires new OpenIE research into addressing the properties desired by downstream applications.

Limitations

Although this work aims to be as comprehensive as possible, there are several limitations to this paper.

Our comparisons only consider neural OpenIE models despite rule-based methods being very popular among downstream applications. This is because of the lack of recent surveys on neural OpenIE methods and the difficulties we personally encountered when trying to determine which OpenIE method was state-of-the-art. We acknowledge that there are many cases where rule-based methods may be preferable to neural models due to being faster or more tailor-made for a specific application. However, we feel that focusing on neural OpenIE methods is not a detriment because we are interested in which methods work best "out of the box". Based on the results reported in these neural OpenIE papers, we believe they are currently the best out-of-the-box OpenIE models using the metrics we report in this paper on the test sets covered in this paper.

The corpora we chose are all limited to English. As a result, our results are not generalizable to any downstream task that relies on different languages.

In our experiments, we do not report results for the BenchIE test set or using the BenchIE metric. This is because the BenchIE test set uniquely can only be evaluated using the BenchIE metric, and the BenchIE metric can only be applied to the BenchIE test set. We do not feel that its exclusion hurts our final conclusions about the relative performance of OpenIE methods.

We perform a case study using Complex QA only, which we generalize to other applications.

For our case study, we were unable to replicate the results reported in the original QUEST paper (Lu et al., 2019). We have been in correspondence with the authors to address this issue, but we still feel that our results are valid given that we use the publicly available code and data and adapted it to use our OpenIE methods to the best of our ability.

Similarly, we report different results to the efficiency and performance of DetIE reported in the original paper (Vasilkovsky et al., 2022). We have been in contact with the original authors and differences in efficiency can be attributed to differing hardware while differences in performance can be attributed to different preprocessing of training and test sets. For instance, the authors of DetIE do not remove duplicate sentences when combining the Science and Wiki domains of LSOIE.

We do not make specific observations based on the different evaluation metrics, mainly focusing on CaRB and WiRe57 F1 score for our evaluation. We give our experimental results within appendix A so that future researchers can make observations and draw conclusions based on OIE2016.

Ethics Statement

We did not create any of the models, datasets, or applications covered in this paper. Any ethical issues with the preexisting OpenIE datasets we use in this paper will reflect on this work.

References

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 (Volume 1: Long Papers)*, pages 344–354.

Niranjan Balasubramanian, Stephen Soderland, Oren Etzioni, et al. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. Carb: A crowdsourced benchmark for open ie. 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 6262–6267.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yllias Chali, Shafiq R Joty, and Sadid A Hasan. 2009. Complex question answering: unsupervised learning approaches and experiments. *Journal of Artificial Intelligence Research*, 35:1–47.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Daniela Barreiro Claro, Marlo Souza, Clarissa Castellã Xavier, and Leandro Oliveira. 2019. Multilingual open information extraction: Challenges and opportunities. *Information*, 10(7):228.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. *arXiv preprint arXiv:1805.04270*.

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

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014a. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014b. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165.

Vikas Garg and Adam T Kalai. 2018. Supervising unsupervised learning. *Advances in Neural Information Processing Systems*, 31.

Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Goran Glavas, and Mathias Niepert. 2021. Benchie: Open information extraction evaluation based on facts, not tokens. *arXiv preprint arXiv:2109.06850*.

Rafael Glauber and Daniela Barreiro Claro. 2018. A systematic mapping study on open information extraction. *Expert Systems with Applications*, 112:372–387.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti, et al. 2020a. Openie6: Iterative grid labeling and coordination analysis for open information extraction. *arXiv preprint arXiv:2010.03147*.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. 2020b. Imojie: Iterative memory-based joint open information extraction. *arXiv preprint arXiv:2005.08178*.

Bhushan Kotnis, Kiril Gashteovski, Carolin Lawrence, Daniel Oñoro Rubio, Vanesa Rodriguez-Tembras, Makoto Takamoto, and Mathias Niepert. 2021. Integrating diverse extraction pathways using iterative predictions for multilingual open information extraction. *arXiv preprint arXiv:2110.08144*.

Hermann Kroll, Jan Pirklbauer, and Wolf-Tilo Balke. 2021. A toolbox for the nearly-unsupervised construction of digital library knowledge graphs. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in*.

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

Xiaolu Lu, Soumajit Pramanik, Rishiraj Saha Roy, Abdalghani Abujabal, Yafang Wang, and Gerhard Weikum. 2019. Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 105–114.

Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pages 4074–4077.

Iqra Muhammad, Anna Kearney, Carrol Gamble, Frans Coenen, and Paula Williamson. 2020. Open information extraction for knowledge graph construction. In *International Conference on Database and Expert Systems Applications*, pages 103–113. Springer.

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

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multi^{2}oie: Multilingual open information extraction based on multi-head attention with bert. *arXiv preprint arXiv:2009.08128*.

Ade Romadhony, Dwi H Widyantoro, and Ayu Purwarianti. 2019. Utilizing structured knowledge bases in open ie based event template extraction. *Applied Intelligence*, 49(1):206–219.

Hadeel Saadany and Constantin Orasan. 2021. Bleu, meteor, bertscore: Evaluation of metrics performance in assessing critical translation errors in sentiment-oriented text. *arXiv preprint arXiv:2109.14250*.

Sihem Sahnoun, Samir Elloumi, and Sadok Ben Yahia. 2020. Event detection based on open information extraction and ontology. *Journal of Information and Telecommunication*, 4(3):383–403.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S Weld. 2013. Open information extraction to kbp relations in 3 hours. In *TAC*.

Stephen Soderland, Natalie Hawkins, John Gilmer, and Daniel S Weld. 2015a. Combining open ie and distant supervision for kbp slot filling. In *TAC*.

Stephen Soderland, Natalie Hawkins, Gene L Kim, and Daniel S Weld. 2015b. University of washington system for 2015 kbp cold start slot filling. *Proceedings of TAC-KBP*, 2015.

Jacob Solawetz and Stefan Larson. 2021. Lsoie: A large-scale dataset for supervised open information extraction. *arXiv preprint arXiv:2101.11177*.

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*, pages 2300–2305.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895.

Michael Vasilkovsky, Anton Alekseev, Valentin Malykh, Ilya Shenbin, Elena Tutubalina, Dmitriy Salikhov, Mikhail Stepnov, Andrey Chertok, and Sergey Nikolenko. 2022. Detie: Multilingual open information extraction inspired by object detection. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*.

Pengcheng Yin, Nan Duan, Ben Kao, Junwei Bao, and Ming Zhou. 2015. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1301–1310.

Junlang Zhan and Hai Zhao. 2020. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530.

Mengli Zhang, Gang Zhou, Wanting Yu, and Wenfen Liu. 2021. Far-ass: Fact-aware reinforced abstractive sentence summarization. *Information Processing & Management*, 58(3):102478.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. 2022. A survey on neural open information extraction: Current status and future directions. *arXiv preprint arXiv:2205.11725*.

A Empirical Results

Model Performance In this section, we report the empirical results of training each model on a variety of training sets and evaluating them on a variety of test sets with different evaluation metrics. Sen./Sec. refers to the number of sentences that could be processed per second, which we use to compare the efficiency of different models. We report Precision (P), Recall (R), F1 Score (F1), and Area Under the Curve (AUC) for the OIE2016, WiRe57, and CaRB metrics. We make observations using these results in Section 7.

Table 9 shows the performance of different OpenIE models trained on different training sets on the OIE2016 benchmark.

Table 10 shows performance on WiRe57.

Table 11 shows performance on ReOIE2016.

Table 12 shows performance on CaRB.

Table 13 shows performance on LSOIE.

Research Questions We also report the empirical results of our student's t-tests comparing different OpenIE systems, which we use to answer the research questions we raise in section 6. For each research question, we report the number of statistical significance tests that had a t-score above or below 0 and had a p-value above or below 0.05. We use these results to answer those research questions in section 7.1.

Table 14 shows the results of the statistical significance tests used to answer R1 from section 6.

Table 15 shows results for R2.

Table 16 shows results for R3.

Model	Training set	Test set	Sen./Sec	OIE2016				WiRe57				CaRB			
				P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
SpanOIE	SpanOIE	OIE2016	16.65	0.704	0.792	0.745	0.675	0.576	0.376	0.455	0.296	0.576	0.459	0.511	0.362
IMoJIE	SpanOIE	OIE2016	2.61	0.755	0.851	0.8	0.614	0.575	0.389	0.464	0.212	0.575	0.466	0.515	0.253
Multi^{2}OIE	SpanOIE	OIE2016	28.21	0.724	0.915	0.809	0.719	0.558	0.439	0.491	0.29	0.566	0.521	0.542	0.348
IGL-OIE	SpanOIE	OIE2016	67.55	0.733	0.768	0.75	0.585	0.551	0.347	0.426	0.211	0.551	0.419	0.476	0.253
CIGL-OIE	SpanOIE	OIE2016	50.61	0.711	0.981	0.824	0.737	0.375	0.474	0.419	0.212	0.375	0.592	0.459	0.263
OpenIE6	SpanOIE	OIE2016	38.38	0.519	0.975	0.678	0.532	0.269	0.492	0.348	0.177	0.269	0.556	0.362	0.2
DetIE	SpanOIE	OIE2016	26.42	0.775	0.787	0.781	0.699	0.55	0.351	0.429	0.272	0.55	0.423	0.478	0.328
SpanOIE	OIE4	OIE2016	16.19	0.703	0.813	0.754	0.692	0.584	0.37	0.453	0.293	0.584	0.454	0.511	0.36
IMoJIE	OIE4	OIE2016	3.44	0.695	0.824	0.754	0.495	0.553	0.399	0.464	0.196	0.553	0.474	0.51	0.231
Multi^{2}OIE	OIE4	OIE2016	31.14	0.747	0.864	0.801	0.72	0.595	0.4	0.478	0.261	0.597	0.491	0.539	0.32
IGL-OIE	OIE4	OIE2016	70.02	0.718	0.84	0.774	0.661	0.544	0.39	0.455	0.257	0.544	0.48	0.51	0.313
CIGL-OIE	OIE4	OIE2016	49.26	0.718	0.92	0.806	0.726	0.529	0.436	0.478	0.289	0.529	0.537	0.533	0.356
OpenIE6	OIE4	OIE2016	24.20	0.557	0.922	0.694	0.615	0.413	0.467	0.438	0.278	0.415	0.523	0.463	0.314
DetIE	OIE4	OIE2016	26.29	0.787	0.855	0.82	0.764	0.563	0.366	0.443	0.286	0.563	0.453	0.502	0.354
SpanOIE	LSOIE	OIE2016	15.36	0.657	0.804	0.723	0.666	0.657	0.432	0.521	0.358	0.657	0.521	0.581	0.432
IMoJIE	LSOIE	OIE2016	1.00	0.852	0.766	0.807	0.577	0.719	0.339	0.461	0.216	0.719	0.411	0.523	0.261
Multi^{2}OIE	LSOIE	OIE2016	31.00	0.758	0.894	0.821	0.767	0.728	0.484	0.582	0.401	0.728	0.585	0.649	0.483
IGL-OIE	LSOIE	OIE2016	68.27	0.762	0.823	0.791	0.634	0.636	0.394	0.487	0.27	0.636	0.485	0.551	0.331
CIGL-OIE	LSOIE	OIE2016	52.40	0.74	0.947	0.831	0.738	0.568	0.494	0.528	0.314	0.568	0.618	0.592	0.391
OpenIE6	LSOIE	OIE2016	24.56	0.542	0.924	0.683	0.563	0.41	0.541	0.466	0.279	0.41	0.609	0.49	0.315
DetIE	LSOIE	OIE2016	26.16	0.857	0.879	0.868	0.816	0.687	0.419	0.521	0.354	0.687	0.528	0.597	0.445
SpanOIE	IMoJIE	OIE2016	7.16	0.188	0.975	0.316	0.579	0.084	0.394	0.138	0.213	0.084	0.428	0.14	0.232
IMoJIE	IMoJIE	OIE2016	1.68	0.779	0.905	0.837	0.607	0.551	0.381	0.451	0.191	0.551	0.451	0.496	0.225
Multi^{2}OIE	IMoJIE	OIE2016	31.58	0.764	0.842	0.801	0.739	0.596	0.378	0.463	0.252	0.599	0.453	0.516	0.302
IGL-OIE	IMoJIE	OIE2016	63.00	0.775	0.797	0.786	0.592	0.545	0.323	0.406	0.194	0.545	0.396	0.459	0.238
CIGL-OIE	IMoJIE	OIE2016	49.62	0.775	0.928	0.845	0.69	0.509	0.375	0.432	0.21	0.509	0.482	0.495	0.269
OpenIE6	IMoJIE	OIE2016	36.42	0.582	0.91	0.71	0.511	0.386	0.416	0.4	0.184	0.386	0.484	0.43	0.215
DetIE	IMoJIE	OIE2016	26.75	0.856	0.709	0.775	0.658	0.606	0.275	0.379	0.221	0.606	0.337	0.433	0.271

Table 9: A table that lists performance of different OpenIE systems on the OIE2016 benchmark.

Model	Training set	Test set	Sen./Sec	OIE2016				WiRe57				CaRB			
				P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
SpanOIE	SpanOIE	WiRe57	9.10	0.87	0.72	0.788	0.673	0.464	0.194	0.274	0.142	0.464	0.372	0.413	0.272
IMoJIE	SpanOIE	WiRe57	0.91	0.863	0.644	0.738	0.465	0.461	0.154	0.231	0.061	0.461	0.313	0.373	0.123
Multi^{2}OIE	SpanOIE	WiRe57	23.17	0.9	0.758	0.823	0.698	0.498	0.203	0.288	0.097	0.498	0.391	0.438	0.186
IGL-OIE	SpanOIE	WiRe57	9.34	0.916	0.638	0.753	0.604	0.482	0.167	0.248	0.097	0.482	0.333	0.394	0.189
CIGL-OIE	SpanOIE	WiRe57	7.75	0.889	0.84	0.864	0.77	0.281	0.195	0.231	0.069	0.283	0.406	0.333	0.145
OpenIE6	SpanOIE	WiRe57	4.93	0.74	0.831	0.783	0.641	0.304	0.28	0.291	0.127	0.28	0.408	0.332	0.167
DetIE	SpanOIE	WiRe57	27.16	0.948	0.743	0.833	0.724	0.47	0.197	0.278	0.145	0.47	0.381	0.421	0.28
SpanOIE	OIE4	WiRe57	9.07	0.895	0.743	0.812	0.704	0.526	0.217	0.307	0.166	0.526	0.397	0.453	0.303
IMoJIE	OIE4	WiRe57	1.19	0.823	0.665	0.735	0.433	0.414	0.189	0.26	0.059	0.414	0.35	0.379	0.109
Multi^{2}OIE	OIE4	WiRe57	19.65	0.921	0.717	0.807	0.67	0.537	0.197	0.289	0.104	0.537	0.37	0.439	0.194
IGL-OIE	OIE4	WiRe57	8.19	0.931	0.673	0.782	0.653	0.452	0.174	0.251	0.111	0.457	0.337	0.388	0.22
CIGL-OIE	OIE4	WiRe57	6.82	0.9	0.787	0.84	0.742	0.436	0.196	0.27	0.123	0.436	0.391	0.413	0.247
OpenIE6	OIE4	WiRe57	3.47	0.799	0.755	0.777	0.662	0.451	0.295	0.357	0.192	0.451	0.397	0.423	0.261
DetIE	OIE4	WiRe57	27.02	0.929	0.843	0.884	0.813	0.491	0.209	0.293	0.156	0.491	0.405	0.444	0.302
SpanOIE	LSOIE	WiRe57	8.52	0.759	0.534	0.627	0.469	0.357	0.135	0.196	0.092	0.357	0.209	0.263	0.142
IMoJIE	LSOIE	WiRe57	0.46	0.961	0.574	0.719	0.534	0.351	0.094	0.148	0.026	0.351	0.182	0.24	0.052
Multi^{2}OIE	LSOIE	WiRe57	18.31	0.851	0.534	0.656	0.485	0.44	0.128	0.198	0.067	0.44	0.202	0.276	0.106
IGL-OIE	LSOIE	WiRe57	9.54	0.92	0.571	0.705	0.549	0.32	0.099	0.151	0.034	0.32	0.183	0.233	0.063
CIGL-OIE	LSOIE	WiRe57	7.65	0.933	0.694	0.796	0.671	0.301	0.114	0.165	0.044	0.301	0.223	0.256	0.082
OpenIE6	LSOIE	WiRe57	3.81	0.766	0.688	0.725	0.554	0.311	0.194	0.239	0.086	0.311	0.247	0.275	0.114
DetIE	LSOIE	WiRe57	27.02	0.916	0.571	0.704	0.547	0.403	0.124	0.19	0.087	0.403	0.223	0.287	0.157
SpanOIE	IMoJIE	WiRe57	7.33	0.303	0.898	0.454	0.585	0.087	0.274	0.133	0.149	0.087	0.364	0.141	0.198
IMoJIE	IMoJIE	WiRe57	1.17	0.911	0.778	0.84	0.622	0.517	0.224	0.313	0.116	0.517	0.404	0.454	0.207
Multi^{2}OIE	IMoJIE	WiRe57	24.83	0.9	0.706	0.791	0.692	0.539	0.195	0.287	0.12	0.539	0.373	0.44	0.228
IGL-OIE	IMoJIE	WiRe57	10.36	0.934	0.7	0.8	0.65	0.48	0.157	0.236	0.08	0.485	0.291	0.364	0.144
CIGL-OIE	IMoJIE	WiRe57	7.83	0.926	0.799	0.858	0.744	0.44	0.196	0.271	0.099	0.44	0.395	0.417	0.197
OpenIE6	IMoJIE	WiRe57	5.76	0.802	0.781	0.792	0.648	0.452	0.292	0.355	0.144	0.459	0.393	0.424	0.2
DetIE	IMoJIE	WiRe57	27.71	0.965	0.65	0.777	0.639	0.526	0.165	0.251	0.126	0.526	0.328	0.404	0.25

Table 10: A table that lists performance of different OpenIE systems on the WiRe57 benchmark.

Model	Training set	Test set	Sen./Sec	OIE2016				WiRe57				CaRB			
				P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
SpanOIE	SpanOIE	ReOIE2016	16.87	0.741	0.842	0.788	0.733	0.772	0.595	0.672	0.527	0.772	0.61	0.681	0.54
IMoJIE	SpanOIE	ReOIE2016	2.71	0.773	0.84	0.805	0.627	0.785	0.601	0.681	0.456	0.785	0.607	0.684	0.46
Multi^{2}OIE	SpanOIE	ReOIE2016	27.70	0.737	0.932	0.823	0.753	0.749	0.688	0.717	0.586	0.749	0.698	0.723	0.596
IGL-OIE	SpanOIE	ReOIE2016	67.16	0.762	0.784	0.773	0.653	0.756	0.557	0.641	0.455	0.756	0.569	0.649	0.465
CIGL-OIE	SpanOIE	ReOIE2016	49.33	0.688	0.991	0.812	0.733	0.437	0.663	0.527	0.35	0.437	0.69	0.535	0.365
OpenIE6	SpanOIE	ReOIE2016	37.80	0.498	0.988	0.662	0.532	0.314	0.628	0.419	0.268	0.314	0.636	0.42	0.272
DetIE	SpanOIE	ReOIE2016	26.63	0.802	0.801	0.802	0.722	0.734	0.55	0.629	0.477	0.734	0.562	0.636	0.487
SpanOIE	OIE4	ReOIE2016	16.72	0.729	0.839	0.78	0.726	0.815	0.604	0.694	0.548	0.815	0.617	0.702	0.56
IMoJIE	OIE4	ReOIE2016	3.00	0.75	0.155	0.257	0.095	0.756	0.119	0.205	0.075	0.756	0.119	0.206	0.075
Multi^{2}OIE	OIE4	ReOIE2016	27.74	0.773	0.869	0.818	0.746	0.813	0.635	0.713	0.55	0.813	0.647	0.72	0.561
IGL-OIE	OIE4	ReOIE2016	64.23	0.751	0.877	0.809	0.72	0.732	0.615	0.668	0.52	0.732	0.629	0.677	0.531
CIGL-OIE	OIE4	ReOIE2016	51.78	0.74	0.948	0.831	0.776	0.698	0.675	0.686	0.564	0.698	0.697	0.698	0.582
OpenIE6	OIE4	ReOIE2016	23.30	0.559	0.938	0.701	0.642	0.506	0.671	0.577	0.467	0.506	0.679	0.58	0.472
DetIE	OIE4	ReOIE2016	26.36	0.798	0.858	0.827	0.771	0.757	0.569	0.65	0.5	0.757	0.587	0.662	0.516
SpanOIE	LSOIE	ReOIE2016	16.33	0.65	0.814	0.723	0.672	0.69	0.53	0.6	0.448	0.69	0.536	0.603	0.453
IMoJIE	LSOIE	ReOIE2016	1.03	0.836	0.726	0.778	0.525	0.747	0.409	0.529	0.279	0.747	0.414	0.533	0.283
Multi^{2}OIE	LSOIE	ReOIE2016	31.24	0.759	0.845	0.8	0.736	0.746	0.582	0.654	0.49	0.746	0.586	0.657	0.495
IGL-OIE	LSOIE	ReOIE2016	69.48	0.742	0.786	0.763	0.602	0.626	0.453	0.525	0.312	0.626	0.472	0.538	0.325
CIGL-OIE	LSOIE	ReOIE2016	53.49	0.715	0.93	0.808	0.716	0.548	0.559	0.553	0.351	0.548	0.582	0.564	0.365
OpenIE6	LSOIE	ReOIE2016	24.94	0.518	0.924	0.664	0.53	0.374	0.562	0.45	0.275	0.374	0.574	0.453	0.281
DetIE	LSOIE	ReOIE2016	27.39	0.847	0.85	0.848	0.785	0.692	0.493	0.575	0.417	0.692	0.513	0.589	0.434
SpanOIE	IMoJIE	ReOIE2016	7.36	0.175	0.993	0.298	0.584	0.099	0.527	0.166	0.289	0.099	0.535	0.167	0.294
IMoJIE	IMoJIE	ReOIE2016	1.84	0.802	0.947	0.868	0.65	0.713	0.592	0.647	0.388	0.713	0.603	0.653	0.395
Multi^{2}OIE	IMoJIE	ReOIE2016	30.72	0.794	0.863	0.827	0.793	0.812	0.606	0.694	0.534	0.817	0.614	0.701	0.542
IGL-OIE	IMoJIE	ReOIE2016	68.80	0.799	0.817	0.808	0.644	0.728	0.508	0.599	0.403	0.728	0.53	0.614	0.42
CIGL-OIE	IMoJIE	ReOIE2016	49.48	0.796	0.919	0.853	0.723	0.671	0.579	0.621	0.431	0.674	0.622	0.647	0.464
OpenIE6	IMoJIE	ReOIE2016	40.95	0.584	0.925	0.716	0.514	0.483	0.601	0.535	0.33	0.483	0.623	0.544	0.342
DetIE	IMoJIE	ReOIE2016	26.83	0.905	0.717	0.8	0.683	0.829	0.442	0.577	0.404	0.829	0.46	0.592	0.421

Table 11: A table that lists performance of different OpenIE systems on the ReOIE2016 benchmark.

Model	Training set	Test set	Sen./Sec	OIE2016				WiRe57				CaRB			
				P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
SpanOIE	SpanOIE	CaRB	17.14	0.81	0.778	0.794	0.704	0.609	0.273	0.377	0.219	0.609	0.403	0.485	0.324
IMoJIE	SpanOIE	CaRB	3.12	0.836	0.794	0.814	0.639	0.629	0.283	0.39	0.17	0.629	0.416	0.5	0.25
Multi^{2}OIE	SpanOIE	CaRB	22.39	0.826	0.878	0.851	0.793	0.59	0.315	0.411	0.22	0.609	0.458	0.523	0.326
IGL-OIE	SpanOIE	CaRB	69.67	0.831	0.771	0.8	0.672	0.611	0.267	0.371	0.184	0.611	0.399	0.483	0.275
CIGL-OIE	SpanOIE	CaRB	52.62	0.789	0.986	0.876	0.818	0.379	0.331	0.354	0.148	0.379	0.508	0.434	0.228
OpenIE6	SpanOIE	CaRB	41.02	0.643	0.981	0.777	0.671	0.335	0.406	0.367	0.181	0.338	0.489	0.399	0.223
DetIE	SpanOIE	CaRB	25.79	0.866	0.788	0.825	0.735	0.595	0.266	0.368	0.212	0.595	0.406	0.483	0.324
SpanOIE	OIE4	CaRB	16.92	0.804	0.777	0.79	0.701	0.646	0.28	0.39	0.23	0.646	0.413	0.503	0.339
IMoJIE	OIE4	CaRB	3.83	0.804	0.816	0.81	0.572	0.624	0.304	0.408	0.17	0.624	0.442	0.517	0.247
Multi^{2}OIE	OIE4	CaRB	33.37	0.838	0.831	0.835	0.761	0.647	0.298	0.408	0.213	0.647	0.442	0.525	0.317
IGL-OIE	OIE4	CaRB	72.82	0.82	0.834	0.827	0.734	0.607	0.298	0.399	0.219	0.607	0.438	0.509	0.323
CIGL-OIE	OIE4	CaRB	58.49	0.814	0.908	0.858	0.796	0.584	0.326	0.418	0.237	0.584	0.479	0.526	0.35
OpenIE6	OIE4	CaRB	24.93	0.685	0.903	0.779	0.716	0.518	0.395	0.448	0.281	0.518	0.482	0.499	0.346
DetIE	OIE4	CaRB	26.28	0.862	0.843	0.852	0.785	0.614	0.277	0.382	0.223	0.614	0.425	0.502	0.343
SpanOIE	LSOIE	CaRB	16.59	0.741	0.731	0.736	0.636	0.561	0.244	0.34	0.191	0.561	0.334	0.418	0.26
IMoJIE	LSOIE	CaRB	1.05	0.896	0.702	0.788	0.569	0.615	0.195	0.296	0.109	0.615	0.281	0.386	0.157
Multi^{2}OIE	LSOIE	CaRB	33.89	0.818	0.81	0.814	0.738	0.611	0.267	0.372	0.189	0.611	0.369	0.461	0.262
IGL-OIE	LSOIE	CaRB	67.65	0.825	0.743	0.782	0.616	0.529	0.215	0.305	0.127	0.529	0.304	0.386	0.178
CIGL-OIE	LSOIE	CaRB	49.70	0.814	0.897	0.853	0.753	0.475	0.273	0.346	0.149	0.475	0.386	0.426	0.21
OpenIE6	LSOIE	CaRB	28.14	0.667	0.898	0.766	0.627	0.403	0.333	0.365	0.168	0.403	0.389	0.396	0.198
DetIE	LSOIE	CaRB	26.27	0.904	0.8	0.849	0.762	0.578	0.234	0.334	0.185	0.578	0.343	0.43	0.27
SpanOIE	IMoJIE	CaRB	7.41	0.265	0.979	0.417	0.619	0.131	0.4	0.198	0.226	0.131	0.438	0.202	0.248
IMoJIE	IMoJIE	CaRB	1.77	0.863	0.914	0.888	0.696	0.633	0.306	0.413	0.179	0.633	0.457	0.531	0.266
Multi^{2}OIE	IMoJIE	CaRB	31.22	0.848	0.813	0.83	0.771	0.645	0.28	0.39	0.201	0.648	0.418	0.508	0.301
IGL-OIE	IMoJIE	CaRB	73.88	0.865	0.803	0.833	0.681	0.615	0.252	0.357	0.165	0.615	0.384	0.473	0.252
CIGL-OIE	IMoJIE	CaRB	55.01	0.855	0.909	0.881	0.768	0.563	0.286	0.379	0.178	0.574	0.437	0.496	0.274
OpenIE6	IMoJIE	CaRB	37.82	0.715	0.898	0.796	0.633	0.498	0.365	0.421	0.204	0.503	0.44	0.47	0.252
DetIE	IMoJIE	CaRB	27.16	0.932	0.69	0.793	0.667	0.67	0.21	0.32	0.175	0.67	0.327	0.439	0.273

Table 12: A table that lists performance of different OpenIE systems on the CaRB benchmark.

Model	Training set	Test set	Sen./Sec	OIE2016				WiRe57				CaRB			
				P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
SpanOIE	SpanOIE	LSOIE	18.56	0.745	0.851	0.794	0.742	0.537	0.388	0.451	0.298	0.537	0.551	0.544	0.423
IMoJIE	SpanOIE	LSOIE	2.92	0.631	0.866	0.73	0.499	0.53	0.516	0.523	0.244	0.53	0.537	0.534	0.253
Multi^{2}OIE	SpanOIE	LSOIE	27.55	0.618	0.909	0.736	0.646	0.525	0.596	0.558	0.364	0.525	0.628	0.571	0.383
IGL-OIE	SpanOIE	LSOIE	205.07	0.636	0.815	0.714	0.582	0.529	0.484	0.505	0.295	0.529	0.506	0.517	0.308
CIGL-OIE	SpanOIE	LSOIE	159.43	0.634	0.975	0.769	0.653	0.379	0.601	0.464	0.284	0.379	0.633	0.474	0.3
OpenIE6	SpanOIE	LSOIE	123.55	0.458	0.965	0.622	0.468	0.268	0.562	0.363	0.215	0.268	0.58	0.366	0.222
DetIE	SpanOIE	LSOIE	29.52	0.664	0.806	0.728	0.671	0.519	0.466	0.491	0.354	0.519	0.489	0.503	0.371
SpanOIE	OIE4	LSOIE	19.48	0.737	0.848	0.788	0.736	0.541	0.382	0.447	0.294	0.541	0.541	0.541	0.416
IMoJIE	OIE4	LSOIE	3.62	0.61	0.89	0.724	0.442	0.52	0.541	0.53	0.239	0.52	0.564	0.541	0.248
Multi^{2}OIE	OIE4	LSOIE	39.12	0.642	0.877	0.742	0.637	0.547	0.517	0.532	0.309	0.547	0.547	0.547	0.327
IGL-OIE	OIE4	LSOIE	196.72	0.628	0.896	0.738	0.659	0.521	0.54	0.53	0.361	0.521	0.566	0.543	0.378
CIGL-OIE	OIE4	LSOIE	191.90	0.617	0.945	0.747	0.692	0.505	0.587	0.543	0.392	0.505	0.621	0.557	0.414
OpenIE6	OIE4	LSOIE	64.24	0.47	0.924	0.623	0.587	0.394	0.537	0.455	0.342	0.394	0.557	0.462	0.354
DetIE	OIE4	LSOIE	30.26	0.667	0.854	0.749	0.712	0.51	0.482	0.496	0.364	0.51	0.509	0.51	0.385
SpanOIE	LSOIE	LSOIE	18.09	0.715	0.888	0.792	0.762	0.666	0.474	0.554	0.394	0.666	0.65	0.658	0.541
IMoJIE	LSOIE	LSOIE	1.09	0.741	0.891	0.809	0.563	0.748	0.571	0.648	0.379	0.748	0.597	0.664	0.395
Multi^{2}OIE	LSOIE	LSOIE	37.98	0.662	0.935	0.775	0.707	0.745	0.676	0.709	0.557	0.745	0.703	0.723	0.579
IGL-OIE	LSOIE	LSOIE	201.64	0.679	0.891	0.771	0.651	0.697	0.611	0.652	0.485	0.697	0.65	0.673	0.515
CIGL-OIE	LSOIE	LSOIE	183.46	0.643	0.978	0.776	0.705	0.621	0.717	0.666	0.529	0.621	0.767	0.686	0.566
OpenIE6	LSOIE	LSOIE	65.63	0.473	0.954	0.633	0.529	0.438	0.723	0.546	0.428	0.438	0.75	0.553	0.447
DetIE	LSOIE	LSOIE	28.19	0.739	0.893	0.809	0.776	0.694	0.579	0.631	0.49	0.694	0.618	0.654	0.523
SpanOIE	IMoJIE	LSOIE	7.19	0.226	0.996	0.368	0.61	0.085	0.389	0.139	0.211	0.085	0.439	0.142	0.238
IMoJIE	IMoJIE	LSOIE	2.98	0.681	0.945	0.792	0.532	0.517	0.497	0.507	0.225	0.517	0.523	0.52	0.236
Multi^{2}OIE	IMoJIE	LSOIE	33.67	0.651	0.882	0.749	0.703	0.554	0.502	0.527	0.333	0.554	0.527	0.54	0.348
IGL-OIE	IMoJIE	LSOIE	218.05	0.691	0.863	0.767	0.567	0.517	0.443	0.477	0.241	0.517	0.472	0.493	0.256
CIGL-OIE	IMoJIE	LSOIE	189.39	0.678	0.934	0.785	0.6	0.489	0.503	0.496	0.262	0.489	0.551	0.518	0.286
OpenIE6	IMoJIE	LSOIE	124.62	0.502	0.924	0.651	0.452	0.353	0.506	0.416	0.207	0.353	0.534	0.425	0.219
DetIE	IMoJIE	LSOIE	30.12	0.742	0.755	0.748	0.657	0.569	0.377	0.454	0.296	0.569	0.4	0.47	0.314

Table 13: A table that lists performance of different OpenIE systems on the LSOIE benchmark.

Independent Var.	Constants	p-value ≤ 0.05		p-value > 0.05	
		t-score > 0	t-score < 0	t-score > 0	t-score < 0
non-N-ary model vs. N-ary model	non-N-ary train, N-ary test	2	5	3	5
	N-ary train, N-ary test	3	5	1	6
non-N-ary train vs. N-ary train	non-N-ary model, N-ary test	0	11	0	4
	N-ary model, N-ary test	4	9	0	2

Table 14: Statistical significance tests to answer R1. Each number represents the number of test set and evaluation metric combinations with the corresponding t-score and p-value. When t-score is greater than 0, non-N-ary outperforms N-ary, and when t-score is less than 0, N-ary outperforms non-N-ary.

Independent Var.	Constants	p-value ≤ 0.05		p-value > 0.05	
		t-score > 0	t-score < 0	t-score > 0	t-score < 0
non-IN model vs. IN model	non-IN train, IN test	9	0	2	1
	IN train, IN test	0	4	7	1
	non-IN train, non-IN test	0	1	2	0
	IN train, non-IN test	3	0	0	0
non-IN train vs. IN train	non-IN model, IN test	6	6	0	0
	IN model, IN test	2	7	0	3
	non-IN model, non-IN test	2	1	0	0
	IN model, non-IN test	2	0	1	0

Table 15: Statistical significance tests to answer R2. Each number represents the number of test set and evaluation metric combinations with the corresponding t-score and p-value. When t-score is greater than 0, non-IN outperforms IN, and when t-score is less than 0, IN outperforms non-IN.

Configuration 1		Configuration 2		t-Score	p-value
Model	Sen./Sec	Model	Sen./Sec		
IMoJIE	2.070	Multi^{2}OIE	29.225	-21.621	1.50E-15
IMoJIE	2.070	IGL-OIE	84.072	-5.501	2.63E-05
IMoJIE	2.070	CIGL-OIE	68.800	-4.929	9.31E-05
IMoJIE	2.070	OpenIE6	28.357	-5.813	1.31E-05

Table 16: Statistical significance tests to answer R3 with *Generative Model vs. Non-generative Model* independent variable . Sentences per second is averaged over all training and test sets.

B Hyperparameter Sensitivity Study

In this section, we report the empirical results of training Multi2OIE on a variety of hyperparameters. For each combination of training and test set, we start with the original hyperparameters used by Ro et al. (2020), then modify one. The different hyperparameter values we test are values the authors test in their hyperparameter search. The hyperparameters the authors change are the number of epochs used for training, the dropout rate for the multi-head attention blocks, the dropout rate for the argument classifier, the batch size, the learning rate, the number of multi-head attention heads, the number of multi-head attention blocks, and the number of dimensions for the position embeddings. The original hyperparameter values Ro et al. (2020) use are in table 17.

Table 18 shows the CaRB score of Multi^{2}OIE trained with different hyperparameters, averaged over all training and test sets.

Table 19 shows the CaRB score averaged over all training sets on the OIE2016 test set.

Table 20 shows the CaRB score averaged over all training sets on the WiRe57 test set.

Table 21 shows the CaRB score averaged over all training sets on the ReOIE2016 test set.

Table 22 shows the CaRB score averaged over all training sets on the CaRB test set.

Table 23 shows the CaRB score averaged over all training sets on the LSOIE test set.

The largest difference in CaRB F1 score from the original model hyperparameters was for Multi^{2}OIE tested on WiRe57. However, it should be noted that WiRe57 only consists of 57 sentences with 343 relations. An incorrect prediction on a single sentence may lead to a significant F1 difference overall. Therefore, we feel that this difference is not due to sensitivity to hyperparameters, but rather due to the sensitivity of WiRe57. For other test sets, we observe much smaller effects of different hyperparameters on the CaRB score.

Hyperparameter	**Value**
Epochs	1
Multi-head Attention Dropout	0.2
Argument Classifier Dropout	0.2
Batch Size	128
Learning Rate	3e-5
Multi-head Attention Heads	8
Multi-head Attention Blocks	4
Position Embedding Dimensions	64

Table 17: The original hyperparameters used by Multi^{2}OIE.

Hyperparameter Changed		Average Difference from Original Hyperparameters			Max CaRB F1 Increase	Max CaRB F1 Decrease
		CaRB P	CaRB R	CaRB F1		
Epochs	2	0.0027	-0.0028	-0.0007	0.0200	-0.0130
	3	0.0028	-0.0025	-0.0003	0.0160	-0.0090
Multi-head Attention Dropout	0.0	0.0028	-0.0039	-0.0023	0.0020	-0.0150
	0.1	0.0006	-0.0027	-0.0015	0.0030	-0.0120
Argument Classifier Dropout	0.0	0.0003	-0.0013	-0.0011	0.0040	-0.0110
	0.1	-0.0005	0.0002	-0.0003	0.0050	-0.0110
Batch Size	64	0.0005	-0.0001	-0.0004	0.0040	-0.0050
Learning Rate	2e-5	-0.0010	0.0029	0.0012	0.0070	-0.0050
	5e-5	0.0031	-0.0061	-0.0033	0.0090	-0.0160
Multi-head Attention Heads	4	-0.0008	0.0013	0.0008	0.0150	-0.0150
Multi-head Attention Blocks	2	0.0011	-0.0009	-0.0006	0.0040	-0.0100
Position Embedding Dimensions	128	-0.0007	-0.0044	-0.0033	0.0030	-0.0130
	256	-0.0019	0.0023	0.0010	0.0140	-0.0110

Table 18: CaRB scores averaged over all training and test set combinations when using Multi^{2}OIE. Each row represents a change of a single hyperparameter from the final hyperparameters used by Ro et al. (2020). The different hyperparameter values tested are the same ones tested by Ro et al. (2020).

Test Set	Hyperparameter Changed		Average Difference from Original Hyperparameters			Max CaRB F1 Increase	Max CaRB F1 Decrease
			CaRB P	CaRB R	CaRB F1		
OIE2016	Epochs	2	0.0017	-0.0067	-0.0040	-0.0010	-0.0100
		3	0.0027	-0.0020	-0.0003	0.0040	-0.0050
OIE2016	Multi-head Attention Dropout	0.0	0.0013	-0.0020	-0.0010	0.0020	-0.0030
		0.1	0.0020	-0.0020	-0.0007	0.0020	-0.0050
OIE2016	Argument Classifier Dropout	0.0	0.0020	-0.0020	-0.0003	0.0000	-0.0010
		0.1	0.0040	0.0017	0.0023	0.0050	-0.0020
OIE2016	Batch Size	64	0.0007	0.0017	0.0010	0.0040	-0.0020
OIE2016	Learning Rate	2e-5	0.0003	0.0007	0.0010	0.0070	-0.0050
		5e-5	0.0043	-0.0073	-0.0033	0.0050	-0.0110
OIE2016	Multi-head Attention Heads	4	0.0030	0.0017	0.0020	0.0070	-0.0010
OIE2016	Multi-head Attention Blocks	2	0.0003	-0.0013	-0.0010	0.0040	-0.0040
OIE2016	Position Embedding Dimensions	128	0.0007	-0.0080	-0.0050	-0.0010	-0.0110
		256	-0.0017	-0.0027	-0.0023	0.0030	-0.0110

Table 19: CaRB scores averaged over all training sets on the OIE2016 test set when using Multi^{2}OIE.

Test Set	Hyperparameter Changed		Average Difference from Original Hyperparameters			Max CaRB F1 Increase	Max CaRB F1 Decrease
			CaRB P	CaRB R	CaRB F1		
WiRe57	Epochs	2	0.0047	0.0013	0.0037	0.0200	-0.0130
		3	0.0087	0.0030	0.0063	0.0160	-0.0030
WiRe57	Multi-head Attention Dropout	0.0	0.0077	-0.0097	-0.0070	-0.0020	-0.0150
		0.1	0.0050	-0.0057	-0.0023	0.0030	-0.0120
WiRe57	Argument Classifier Dropout	0.0	0.0017	-0.0060	-0.0047	0.0040	-0.0110
		0.1	-0.0007	-0.0047	-0.0033	0.0010	-0.0110
WiRe57	Batch Size	64	0.0067	-0.0033	-0.0017	0.0020	-0.0050
WiRe57	Learning Rate	2e-5	0.0043	0.0000	0.0010	0.0070	-0.0030
		5e-5	0.0063	-0.0080	-0.0053	0.0090	-0.0160
WiRe57	Multi-head Attention Heads	4	-0.0020	0.0020	0.0020	0.0150	-0.0150
WiRe57	Multi-head Attention Blocks	2	0.0013	-0.0020	-0.0013	0.0030	-0.0100
WiRe57	Position Embedding Dimensions	128	0.0000	-0.0080	-0.0060	0.0030	-0.0130
		256	-0.0007	0.0033	0.0037	0.0140	-0.0060

Table 20: CaRB scores averaged over all training sets on the WiRe57 test set when using Multi^{2}OIE.

Test Set	Hyperparameter Changed		Average Difference from Original Hyperparameters			Max CaRB F1 Increase	Max CaRB F1 Decrease
			CaRB P	CaRB R	CaRB F1		
ReOIE2016	Epochs	2	-0.0023	-0.0043	-0.0037	-0.0010	-0.0090
		3	-0.0030	-0.0070	-0.0060	-0.0040	-0.0090
ReOIE2016	Multi-head Attention Dropout	0.0	0.0010	-0.0040	-0.0017	0.0000	-0.0040
		0.1	-0.0017	-0.0040	-0.0030	-0.0020	-0.0040
ReOIE2016	Argument Classifier Dropout	0.0	-0.0020	0.0007	-0.0003	0.0020	-0.0020
		0.1	-0.0060	0.0020	-0.0013	0.0000	-0.0020
ReOIE2016	Batch Size	64	-0.0050	0.0017	-0.0010	0.0000	-0.0020
ReOIE2016	Learning Rate	2e-5	-0.0037	0.0047	0.0017	0.0060	-0.0010
		5e-5	-0.0037	-0.0060	-0.0050	-0.0030	-0.0080
ReOIE2016	Multi-head Attention Heads	4	-0.0037	0.0023	0.0003	0.0040	-0.0050
ReOIE2016	Multi-head Attention Blocks	2	0.0013	-0.0007	-0.0003	0.0000	-0.0010
ReOIE2016	Position Embedding Dimensions	128	-0.0043	-0.0027	-0.0033	0.0000	-0.0060
		256	-0.0043	0.0043	0.0010	0.0060	-0.0050

Table 21: CaRB scores averaged over all training sets on the ReOIE2016 test set when using Multi^{2}OIE.

Test Set	Hyperparameter Changed		Average Difference from Original Hyperparameters			Max CaRB F1 Increase	Max CaRB F1 Decrease
			CaRB P	CaRB R	CaRB F1		
CaRB	Epochs	2	0.0070	-0.0030	-0.0003	0.0020	-0.0030
		3	0.0027	-0.0033	-0.0017	0.0010	-0.0040
CaRB	Multi-head Attention Dropout	0.0	0.0040	-0.0040	-0.0020	0.0000	-0.0030
		0.1	-0.0023	-0.0020	-0.0020	0.0000	-0.0030
CaRB	Argument Classifier Dropout	0.0	0.0003	-0.0003	-0.0003	0.0010	-0.0030
		0.1	0.0010	-0.0003	-0.0003	0.0000	-0.0010
CaRB	Batch Size	64	0.0007	-0.0003	-0.0003	0.0010	-0.0010
CaRB	Learning Rate	2e-5	-0.0017	0.0020	0.0007	0.0010	0.0000
		5e-5	0.0053	-0.0047	-0.0020	0.0010	-0.0060
CaRB	Multi-head Attention Heads	4	-0.0010	-0.0007	-0.0010	0.0030	-0.0030
CaRB	Multi-head Attention Blocks	2	0.0043	-0.0023	-0.0003	0.0010	-0.0010
CaRB	Position Embedding Dimensions	128	0.0017	-0.0027	-0.0017	0.0000	-0.0040
		256	-0.0007	0.0000	0.0000	0.0020	-0.0030

Table 22: CaRB scores averaged over all training sets on the CaRB test set when using Multi^{2}OIE.

Test Set	Hyperparameter Changed		Average Difference from Original Hyperparameters			Max CaRB F1 Increase	Max CaRB F1 Decrease
			CaRB P	CaRB R	CaRB F1		
LSOIE	Epochs	2	0.0027	-0.0013	0.0007	0.0080	-0.0040
		3	0.0030	-0.0030	0.0003	0.0080	-0.0040
LSOIE	Multi-head Attention Dropout	0.0	0.0000	0.0003	0.0003	0.0010	0.0000
		0.1	0.0000	0.0003	0.0003	0.0010	-0.0010
LSOIE	Argument Classifier Dropout	0.0	-0.0007	0.0010	0.0003	0.0010	0.0000
		0.1	-0.0007	0.0023	0.0010	0.0020	0.0000
LSOIE	Batch Size	64	-0.0003	-0.0003	0.0000	0.0020	-0.0020
LSOIE	Learning Rate	2e-5	-0.0043	0.0073	0.0017	0.0050	-0.0030
		5e-5	0.0030	-0.0047	-0.0007	0.0040	-0.0040
LSOIE	Multi-head Attention Heads	4	-0.0003	0.0013	0.0007	0.0020	-0.0010
LSOIE	Multi-head Attention Blocks	2	-0.0017	0.0017	0.0000	0.0010	-0.0010
LSOIE	Position Embedding Dimensions	128	-0.0013	-0.0007	-0.0007	0.0000	-0.0010
		256	-0.0020	0.0067	0.0027	0.0050	0.0000

Table 23: CaRB scores averaged over all training sets on the LSOIE test set when using Multi^{2}OIE.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 10

☒ A2. Did you discuss any potential risks of your work?
We do not believe our observations can be used for adversarial attacks or have malicious effects. We train models that are already publicly available on data that is also already publicly available.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Sections 3-7

☑ B1. Did you cite the creators of artifacts you used?
Sections 3-7, links to the code and datasets used are in the code and data files attached to the submission

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We did not plan to use the artifacts for any commercial applications because we were writing a survey paper.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We did not plan to use the artifacts for any commercial applications because we were writing a survey paper. We were using them purely for research purposes.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we use are relations in sentences. We do not believe these data may lead to a violation of privacy. The source for the sentences were scientific articles, news articles, and Wikipedia, which we believe do not contain offensive content.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Sections 5, 7

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We did not believe the models we used were large enough to warrant this discussion, and we ran all models on a single GPU.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5.1, experimental setup

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6, Appendix B, we did not include error bars but we describe how we obtained our results and how we averaged them to reach our conclusions.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5.1, experimental setup

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.