A Survey on Neural Open Information Extraction: Current Status and Future Directions

Shaowen Zhou[1,2] , **Bowen Yu**[3] , **Aixin Sun**[1] , **Cheng Long**[1] , **Jingyang Li**[3] and **Jian Sun**[3]
[1]Nanyang Technological University
[2]Alibaba-NTU Singapore Joint Research Institute
[3]Alibaba Group, China
s200061@e.ntu.edu.sg, yubowen.ybw@alibaba-inc.com,
{axsun, c.long}@ntu.edu.sg, {qiwei.ljy, jian.sun}@alibaba-inc.com

Abstract

Open Information Extraction (OpenIE) facilitates domain-independent discovery of relational facts from large corpora. The technique well suits many open-world natural language understanding scenarios, such as automatic knowledge base construction, open-domain question answering, and explicit reasoning. Thanks to the rapid development in deep learning technologies, numerous neural OpenIE architectures have been proposed and achieve considerable performance improvement. In this survey, we provide an extensive overview of the state-of-the-art neural OpenIE models, their key design decisions, strengths and weakness. Then, we discuss limitations of current solutions and the open issues in OpenIE problem itself. Finally we list recent trends that could help expand its scope and applicability, setting up promising directions for future research in OpenIE. To our best knowledge, this paper is the first review on neural OpenIE.

Deep learning is a class of ML algorithms that uses multiple layers to extract features from the raw input.

(Deep learning; **is a class of**; ML algorithms)
(Deep learning; **uses**; multiple layers)
(Deep learning; **extracts**; features; from the raw input)

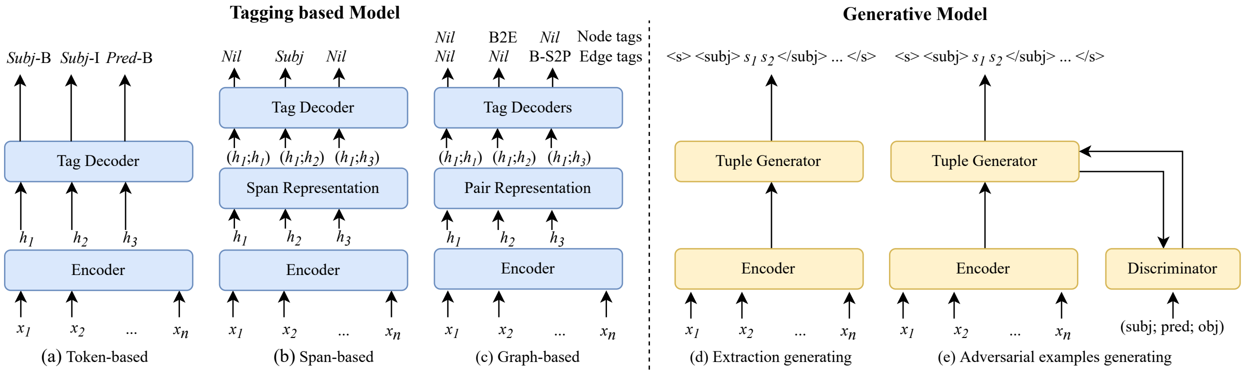
Figure 1: OpenIE tuples extracted from an example sentence (found in Wikipedia). A tuple consists of a predicate (in bold) and several arguments, representing a fact extracted from the sentence.

1 Introduction

Open Information Extraction (OpenIE) extracts facts in the form of n-ary relation tuples, *i.e.,* ($\texttt{arg}_1$, $\texttt{predicate}$, $\texttt{arg}_2$, ..., $\texttt{arg}_n$), from unstructured text, without relying on predefined ontology schema [Niklaus *et al.*, 2018]. Figure 1 shows example OpenIE tuples extracted from a given sentence. Compared to traditional (or closed) IE systems that request predefined relations, OpenIE relieves human labor on designing sophisticated and domain-dependent relation schema. Hence, it has the potential to handle heterogeneous corpora with minimal human intervention. With OpenIE, Web-scale unconstrained IE systems can be developed to acquire large quantities of knowledge. The gathered knowledge can then be integrated and used in a wide range of natural language processing (NLP) applications, such as textual entailment [Berant *et al.*, 2011], summarization [Stanovsky *et al.*, 2015], question answering [Fader *et al.*, 2014; Mausam, 2016], and explicit reasoning [Fu *et al.*, 2019].

Before deep learning, traditional OpenIE systems are either statistical or rule-based, and heavily rely on the analysis of syntactic patterns [Niklaus *et al.*, 2018]. Recently, neural OpenIE solutions become popular, thanks to the large-scale OIE benchmarks (*e.g.,* OIE2016 [Stanovsky and Dagan, 2016], CaRB [Bhardwaj *et al.*, 2019]), and the great success of neural-based models on various NLP tasks (*e.g.,* NER [Li *et al.*, 2022], machine translation [Yang *et al.*, 2020]). Starting with Stanovsky *et al.* 2018 and Cui *et al.* 2018, neural-based approaches dominate OpenIE research for their promising extraction quality on multiple OpenIE benchmarks. Neural solutions mainly formulate OpenIE as a sequence tagging problem or a sequence generation problem. Tagging-based methods tag a token or a span in a sentence as an argument or a predicate [Stanovsky *et al.*, 2018; Kolluru *et al.*, 2020a; Zhan and Zhao, 2020]. Generative methods generate extractions from sentence input with an auto-regressive neural architecture [Cui *et al.*, 2018; Kolluru *et al.*, 2020b]. Some recent work focuses on neural model parameter calibration by introducing a new loss [Jiang *et al.*, 2019], or a new objective to achieve syntactically sound and semantically consistent extraction [Tang *et al.*, 2020].

In this paper, we systematically review neural OpenIE systems. Existing OpenIE reviews [Niklaus *et al.*, 2018; Glauber and Claro, 2018; Claro *et al.*, 2019] focus on traditional solutions and do not well cover the recent neural-based methods. Due to the paradigm change, potential avenues for future research opportunities of OpenIE need to be reconsidered as well. In this survey, we summarise recent research developments, categorise existing neural OpenIE approaches, identify remaining issues, and discuss open problems and future directions. The notable contributions are summarized as follows: **1)** We propose a taxonomy of neural OpenIE models based on their task formulation. We then discuss their strengths and weaknesses; **2)** We provide an informative discussion on the background and evaluation methods for Ope-



(a) Token-based (b) Span-based (c) Graph-based (d) Extraction generating (e) Adversarial examples generating

Figure 2: A taxonomy of neural OpenIE model architectures

nIE. We also offer a detailed comparison of current SOTA methods; **3**) We discuss three challenges that restrict the development of OpenIE: evaluation, annotation, and application. Based on them, we highlight future directions: more open, more focused and more unified.

2 Neural OpenIE Solutions

Formally, given a sentence as a sequence of tokens/words $S = \langle w_1, w_2, \dots, w_n \rangle$, OpenIE outputs a list of tuples $T = T_1, T_2, \dots, T_p$ with the i-th tuple $T_i = \langle a_{i1}, p_i, a_{i2}, ..., a_{iq} \rangle$ representing a fact in the source sequence. Here, p_i denotes the predicate in T_i, and a_{ij} is p_i's j-th argument. The first argument in a tuple is considered as the subject. The maximum number of arguments m per tuple is pre-defined: $m = 2$ for binary and $m \geq 3$ for n-ary relation extraction.

Based on task formulation, we categorize neural OpenIE models into *tagging-based models* and *generative models*, see Figure 2. Next, we review architectures in the two categories, and brief solutions that focus on parameter calibration.

2.1 Tagging-based Models

Tagging-based models formulate OpenIE as a sequence tagging task. Given a set of tags each of which indicates a role (*e.g.,* argument, predicate) of a token or a span of tokens, the model learns the probability distribution of the tag of each token or span conditioned on sentence. Then, the OpenIE system outputs tuples based on the predicted tags.

Tagging-based OpenIE models share a similar architecture to other neural models for sequence tagging tasks in NLP (*e.g.,* NER [Li *et al.*, 2022]). A model usually contains three modules: an *embedding layer* to produce distributed representation of tokens, an *encoder* to generate context-aware token representations, and a *tag decoder* to predict the tag based on token representation and tagging scheme. The embedding layer often concatenates word embeddings with syntactic feature embeddings to better capture syntactic information in sentence. Recently, pre-trained language models (PLMs) have showed superior performance across various NLP tasks [Devlin *et al.*, 2019]. Because PLMs produce context-aware token representations, they can be used either to produce token embedding or as encoders.

Based on tagging schemes, we categorize the models into token-based, span-based, and graph-based models.

Token-based Models

Token-based models predict whether a token is (or a part of) an argument or a predicate. A common tagging scheme is BIO for **B**eginning, **I**nside, and **O**ut of a role *i.e.,* argument and predicate. Figure 2(a) gives an example of a two-token subject and one-token predicate. A token is tagged with 'O' if it is not part of an argument or predicate.

RnnOIE [Stanovsky *et al.*, 2018] considers part-of-speech (POS) feature, and uses Bi-directional LSTM (BiLSTM) [Srivastava *et al.*, 2015] to capture sentence context. It applies fully connected network with softmax layer on the output of the encoder, to produce probability distributions over all tags for each token. SenseOIE [Roy *et al.*, 2019] follows RnnOIE's model structure and introduces one-hop neighbours of a token in dependency tree as syntactic features. Instead of predicting all tags in a single task, Multi^{2}OIE [Ro *et al.*, 2020] designs two sub-tasks. One predicts predicate and the other predicts the arguments that are associated to the predicted predicate. Representation of predicate tokens are used as a feature to predict arguments. The model is also the first using PLM as sentence context encoder. OpenIE6 [Kolluru *et al.*, 2020a] implements an iterative grid labeling (IGL) system that organizes tag sequences in a 2D grid. Each sequence corresponds to an extraction. It uses a PLM to obtain contextualized token embedding, then feeds them to a transformer-based network [Vaswani *et al.*, 2017]. The latter decodes multiple sequences of tags iteratively based on sentence input and embedding of the labels obtained in the previous step.

Token-based model is straightforward. However, arguments and predicates are often token spans. Models which predict tags for individual tokens may not well capture the higher-level relationship among arguments and predicates.

Span-based Models

Span-based models directly predict whether a *token span* is an argument or a predicate. Figure 2(b) gives an example span $(h_1; h_2)$ which is identified as a subject from input. Typically, all possible token spans are enumerated from input sentence. Each token span is then assigned a tag indicating its role of predicate, an argument, or otherwise not to be extracted. Enumerated token spans may overlap with others. In general, a token span representing an argument should not overlap with the one representing a predicate. This case can be handled during inference using hand-crafted constraints. For model design, SpanOIE [Zhan and Zhao, 2020] considers POS and dependency relation between a token and its syntactic parent as syntactic features, and uses BiLSTM to produce contextualized token representation. Representation of a span is derived from the representation of its first and last tokens. Tag decoder then decodes tag from span representation.

Span-based methods consider a token span as the basic unit when deciding argument or predicate labels. This may help the model capture relationship among arguments and predicates. However, too many candidate spans that are neither argument nor predicate are generated, and it is time-consuming to enumerate all spans. Existing methods often set a maximum span length. Span-based methods also have difficulty in extracting tuple elements with discontinuous tokens, *e.g., "geography books"* is an argument with discontinuous tokens in sentence *"Alice likes geography and history books"*.

Graph-based Models

Graph-based models build a graph on token spans to identify triplets. MacroIE [Yu *et al.*, 2021a] constructs a graph with nodes being token spans, and edges indicating the connected nodes belonging to the same fact. It extracts tuples by finding maximal cliques in the graph. To construct nodes, it assigns a binary indicator (*i.e.,* $B2E$ tag shown in Figure 2(c)) to each token span; if the indicator is true, then the token span is a node. To construct edges, it assigns tags to a boundary token pair. Each token in the pair is from one token span. The assigned tag consists of two parts. The first part indicates whether the two boundary tokens are both at the beginning or at the end of the two corresponding token spans. The second part indicates the role of the two token spans. For example, $B\text{-}S2P$ tag shown in Figure 2(c) means that token x_1 and x_3 are at the start of a subject and a predicate spans respectively. The model learns node and edge representations using the same architecture. It uses a BERT-based encoder to learn contextualized token representation. The model then derives span's representation from token representations, and predicts labels with a simple tag decoder.

Graph-based methods model association between tuple elements, instead of directly predicting tuples. They can extract all tuples in a single run, and better handle overlapping and discontinuous arguments or predicates. However, the current design assigns labels to all token pairs, leading to a large number of NULL labels. The imbalanced label distribution may also harm the model's performance.

2.2 Generative Models

Generative models formulate OpenIE as a sequence generation problem that reads a sentence and outputs a sequence of extractions. Figure 2(d) gives an example of the generated sequence. Formally, given a sequence of tokens S and the expected extraction sequence $Y = \langle y_1, y_2, \ldots, y_m \rangle$, the model maximises the conditional probability $P(Y|S) = \prod_{i=1}^{m} p(y_i|y_1, y_2, \ldots, y_{i-1}; S)$. There is also work which generates adversarial tuples with the goal of making it difficult for a classifier to distinguish them from golden tuples.

Generate Extractions. The generative model architecture typically consists of: *an encoder* to give a distributed representation of the sentence context, and *a decoder* to generate tuples sequentially, based on sentence context and the sequence generated so far. NOIE [Cui *et al.*, 2018] uses a 3-layer stacked LSTM as both encoder and decoder. To handle out of vocabulary (OOV) issues and retain information in source sentence, it applies a simplified copy mechanism [Gu *et al.*, 2016] to copy words from the source sentence to the generated sequences. It also applies attention mechanism [Bahdanau *et al.*, 2015] for the RNN-based decoder to refer to the whole input sequence, instead of relying solely on the context representation produced by the encoder. Logician [Sun *et al.*, 2018] uses bi-directional GRU [Cho *et al.*, 2014] as both encoder and decoder. It reduces the vocabulary size to include only predefined keywords, so that more words will be copied from the source sentence. It also implements the coverage mechanism [Tu *et al.*, 2016] and explores encoding dependency parse features in the alignment model. The purpose is to reduce redundant extractions and to improve prediction accuracy. IMoJIE [Kolluru *et al.*, 2020b] uses BERT as encoder, and LSTM as decoder. Focusing on the redundant extraction issue in generative OpenIE models, it proposes an iterative tuple generation mechanism. This mechanism appends all tuples generated previously to the source sentence as the input, to produce the next tuple. It allows the decoder accessing all previous extractions directly, but seriously slows down the extraction speed.

Generate Adversarial Examples. Adversarial-OIE [Han and Wang, 2021] is based on Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014]. The model aims to obtain a generator which can generate tuples so similar to the gold annotations that a discriminator cannot distinguish them. The architecture consists of a transformer-based tuple generator, a Convolutional Neural Network (CNN) based discriminator, and policy gradient method REINFORCE [Williams, 1992] for optimizing the generator in an adversarial manner.

2.3 Model Comparison

Compared with generative models, most tagging-based models are non-autoregressive. This fundamental difference leads to four typical model differences: 1) **Extraction dependency.** Auto-regressive models predict next tuples based on previous predictions, leading to unnecessary sequential dependency among tuples. This dependency may cause error propagation among multiple steps. At the same time, such dependency may also leverage correlation between facts, to realize reasoning for better extraction. 2) **Extraction flexibility.** Tagging-based models are not as flexible as generative models. They assign labels to tokens and extract tokens without modification; thus the extracted tuples may be incoherent. Consider an example sentence "*Born in 1879, Albert Einstein is one of the most influential scientist of the 20th century.*" The predicate of an extraction may be "*born in*", but a more coherent predicate is "*was born in*". Though OpenIE6 partially solves this problem by introducing supplementary words such as *"is"*, *"of"* and *"for"*, the cases such as predicate needs adjustment according to syntactic rules remain unsolved. 3) **Extraction faithfulness.** On the other hand, the flexibility of generative models also brings in the risk of unfaithful extraction: meaningless facts that are not expressed in the original text may be generated. 4) **Extraction speed**:

	OIE16		**OIE16(S)**		**CaRB(OIE16)**		**CaRB(1-1)**		**CaRB**	
OpenIE System	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
Rule-based										
ClausIE [Del Corro and Gemulla, 2013]	59	38	-	-	61.0	38.0	40.2	17.7	45.0	22.0
OpenIE4 [Mausam, 2016]	60	42	-	-	54.3	37.1	40.5	20.1	51.6	29.5
Tagging-based										
RnnOIE [Stanovsky *et al.*, 2018]	62	48	20.4	5.0	56.0	32.0	39.3	18.3	49.0	26.1
SenseOIE [Roy *et al.*, 2019]	-	-	-	-	31.1	-	23.9	-	28.2	-
SpanOIE [Zhan and Zhao, 2020]	**69.4**	**49.1**	-	-	54.0	-	37.9	-	48.5	-
Multi^{2}OIE [Ro *et al.*, 2020]	-	-	-	-	-	-	-	-	52.3	32.6
OpenIE6 [Kolluru *et al.*, 2020a]	-	-	-	-	**65.6**	**48.4**	41.0	22.9	52.7	33.7
MacroIE [Yu *et al.*, 2021a]	-	-	-	-	-	-	**43.5**	**25.0**	**54.8**	**36.3**
Generative										
NOIE [Cui *et al.*, 2018]	-	47.3	-	-	53.5	37.0	38.3	19.8	51.1	32.8
IMoJIE [Kolluru *et al.*, 2020b]	-	-	-	-	56.8	39.6	41.2	22.2	53.3	33.3
Calibrating RnnOIE Model										
[Jiang *et al.*, 2019]	-	-	31.5	12.5	-	-	-	-	-	-
[Tang *et al.*, 2020]	-	-	**32.2**	**15.9**	-	-	-	-	-	-

Table 1: The performance of neural OpenIE systems on two popular benchmarks OIE2016 and CaRB, each with multiple partial matching strategies. The best results under each evaluation setting (based on the available scores) are in boldface, and the second best are underlined. The results missing in the literature are marked as "-". Since Logician is only evaluated on a Chinese benchmark, and Adversarial-OIE only gives precision-recall curve without AUC score on OIE2016, these two systems are not listed here. For comprehensiveness, we also include scores of two popular rule-based systems *i.e.,* ClausIE and OpenIE4.

Autoregressive models output results step by step. Being non-autoregressive, tagging-based methods can output results simultaneously by taking advantage of GPU parallelism. For example, the inference speed of the SOTA tagging model MacroIE [Yu *et al.*, 2021a] is about 35 times faster than generative model model IMoJIE [Kolluru *et al.*, 2020b].

2.4 Calibrating Neural OpenIE Models

Some work focuses on calibrating parameters of existing neural models to improve extraction quality. [Jiang *et al.*, 2019] adds a new optimization goal to a tagging based model. The basic idea is to normalize confidence scores of the extractions, so that they are comparable across sentences. The optimization goal minimizes the binary classification loss which distinguishes correct extractions from wrong ones across different sentences. In addition, the authors also propose an iterative learning mechanism which incrementally includes extractions that participate in the computation of binary classification loss. This mechanism calibrates model parameters, and improves training examples for binary classification at the same time, leading to improved performance. [Tang *et al.*, 2020] proposes a syntactic and semantic-driven reinforcement learning method to enhance supervised OpenIE models (*e.g.,* RnnOIE). It also improves the confidence score by incorporating an extra semantic consistency score.

3 Performance Evaluation

In OpenIE, the input sentence is not restricted to any domain, and the extraction process does not rely on any predefined ontology schema. Hence, it becomes a challenge to derive a unified standard to judge the quality of extractions.

Since neural-based solutions are evaluated on benchmark datasets [Stanovsky and Dagan, 2016; Bhardwaj *et al.*, 2019], we first introduce their annotation standards. Common annotation standards include completeness, correctness, and minimality. Completeness requires an OpenIE system to extract all information in a sentence. Correctness requires the extracted tuples to be implied from the sentence, and to have meaningful interpretation. Minimality requires the elements of a tuple to be indivisible units. Consider an example sentence *"Jeff Bezos founded Amazon and Blue Origin"*. *"Amazon and Blue Origin"* should be two arguments *"Amazon"* and *"Blue Origin"*, *i.e.,* two extractions are formed.

3.1 Evaluation Setting

OpenIE systems are typically evaluated by comparing the extractions with the gold set. Commonly used measures are F1 and PR-AUC scores. Table 1 lists the results collected from literature on two English benchmarks: OIE2016 [Stanovsky and Dagan, 2016] and CaRB [Bhardwaj *et al.*, 2019].

OIE2016 is the first large-scale OpenIE benchmark. It is created by automatic conversion from QA-SRL [He *et al.*, 2015], a semantic role labeling dataset. The sentences are from news (*e.g.,* WSJ) and encyclopedia (*e.g.,* WIKI) domains. Since there are no restrictions on the elements of OpenIE extractions, partial-matching criteria instead of exact-matching is typically used. Hence, the evaluation script can tolerate the extractions that are slightly different from the gold annotation. OIE2016 proposes to follow the matching criteria introduced in [He *et al.*, 2015], and considers two tuples a match if both share the same grammatical head of all of the elements. However, [Jiang *et al.*, 2019] noted that the evaluation metric implemented in the public code of OIE2016 uses a more lenient lexical overlap instead. [Jiang *et al.*, 2019] and [Tang *et al.*, 2020] follow syntactic-head matching metric and report much lower scores than those in the OIE2016 original paper. In Table 1, columns "OIE16" and "OIE16(S)" list the results of using OIE2016 data evaluated by lexical-match and syntactic-head matching criteria respectively.

CaRB [Bhardwaj *et al.*, 2019] is developed by re-annotating the dev and test splits of OIE2016 via crowd-sourcing. Besides improving annotation quality, CaRB also provides a new matching scorer. CaRB scorer uses token level match and it matches relation with relation, arguments with arguments. The authors also design an extraction-gold pair match table which records the similarity scores of extraction-gold pairs for a sentence. During precision computation, each extraction is matched exclusively to one gold tuple. The extraction having the highest matching score with a gold tuple form the first exclusive match. Then the matched gold tuple is removed from the subsequent matching. The next extraction having the highest matching score with one of the remaining gold tuples forms the next exclusive match. The same matching process applies to all of the remaining extractions. Precision is the average matching scores of all extraction matches. During recall computation, CaRB scorer allows one extraction being matched by multiple gold tuples, to avoid penalizing an extraction which covers the information conveyed in multiple gold tuples. [Kolluru *et al.*, 2020a; Yu *et al.*, 2021a] also conduct experiments with other matching criteria, such as OIE2016 which is introduced earlier. They also experiment one-to-one match, which is to replace multi-to-one mapping during recall computation with one-to-one mapping. In Table 1, the columns “CaRB(OIE2016)”, “CaRB(1-1)” and “CaRB” list the results of using CaRB data evaluated by lexical-match, one-to-one, and the original CaRB matching criteria, respectively.

3.2 Discussion

Bootstrapping of training data. Training deep neural models typically requires large volume of annotated data. To obtain sufficient “annotated data”, most neural-based OpenIE systems bootstrap training data by using existing systems (*e.g.,* rule-based systems). For example, NOIE [Cui *et al.*, 2018] bootstraps training set by applying OpenIE4 [Mausam, 2016] to a Wikipedia dump. Some work explores mixing training samples that are produced by multiple OpenIE systems to increase sample diversity. SenseOIE [Roy *et al.*, 2019] combines extractions from three OpenIE systems including Stanford OIE [Angeli *et al.*, 2015], OpenIE5 [Saha and Mausam, 2018] and UKG. IMoJIE [Kolluru *et al.*, 2020b] further improves the mixture quality by introducing a score-and-filter framework to denoise the extractions from multiple systems. IMoJIE reports a small increase of F1 score when compared to training using the best performing single source data. Likely, using more data sources or more advanced data argumentation techniques may further improve neural OpenIE performance. On the other hand, as the annotations are from existing systems, quality of the pseudo labels puts a limit to neural OpenIE models.

Common errors of neural OpenIE extractions. Neural OpenIE systems suffer from same common errors found in traditional systems [Schneider *et al.*, 2017]. Besides, the limitation of neural methods also magnifies some issues. As discussed in [Kolluru *et al.*, 2020b], generative models (*e.g.,* NOIE) suffer from redundant extractions. Due to cascading error, it is also difficult for generative models to extract all tuples when a sentence contains many gold tuples. Extractions produced by tagging-based methods are more likely to lack auxiliary words and implied propositions. Such extractions are marked partially correct in evaluation.

Which model performs the best? We first compare results of tagging-based and generative neural OpenIE systems in Table 1. SpanOIE performs significantly better than RnnOIE on OIE16 benchmark. However, it performs slightly worse than RnnOIE on CaRB benchmark, even using the same partial-matching scorer as OIE16. This means how gold annotation is derived greatly affect the results. Without high quality benchmarks for OpenIE, it is inconclusive to state which model performs the best in general. We expect the OpenIE community to produce more benchmarks across more domains (besides news and encyclopedia), under unified annotation standard. Another question is whether neural OpenIE systems always give higher quality extractions. On OIE2016 benchmark, neural-based OpenIE systems achieve better F1 and AUC score than rule-based systems. However, on CaRB, rule-based OpenIE4 outperforms many neural-based OpenIE systems. Though recent neural OpenIE systems (*e.g.,* IMoJIE, OpenIE6, and MacroIE) perform better than rule-based ones on CaRB, the improvement is not significant. To the best of our knowledge, there is no study systematically comparing neural and rule-based OpenIE systems. Note that, accuracy of current neural OpenIE systems may be limited by the low quality training data bootstrapped from rule-based systems.

4 Challenges and Future Directions

Neural OpenIE systems learn high-level features automatically from training data. This new paradigm imposes new challenges and also opens up new research opportunities.

4.1 Challenges

Evaluation. Large-scale high-quality training data remain lacking for neural OpenIE. For the same reason, neural OpenIE systems usually bootstrap training examples. However, extractions generated by existing OpenIE systems themselves are noisy, therefore limit the performance neural OpenIE models. Creating high-quality training data is time consuming and expensive. Moreover, determining annotation specifications is difficult for OpenIE. Compared to closed IE which relies on predefined ontology schema in predictable domains, OpenIE imposes very few restrictions on their extractions. Thus different annotators may expect different facts to be extracted. Due to various language phenomena in open domain, it is difficult to design a detailed and comprehensive annotation manual. Conceptually, as long as the extracted facts are comprehensible and semantically consistent with the source text, they are considered valid extractions. Though recent OpenIE benchmarks provide annotation guidelines of completeness, correctness, and minimality, more detailed specifications are much expected [Léchelle *et al.*, 2019].

Definition. OpenIE is defined for open domain information extraction. However, most existing studies evaluate their solutions on news, encyclopedia, or web pages. Groth *et* *al.* 2018 compare performance of traditional OpenIE systems on science, medical and general audience corpus. They find that systems perform much worse on science or medical corpus. Performance of neural OpenIE systems in domains other than news or encyclopedia is unknown, due to the lack of such benchmarks. It is also unknown how OpenIE systems perform on informal user-generated contents like tweets. Hence benchmarks covering more domains are necessary. It is also questionable whether an ominous OpenIE system that performs well on corpus in any domain is achievable. Word and grammatical patterns may vary largely in different domains.

Application. Compared to closed IE, the extractions from OpenIE are more difficult to use. There is possibility of multiple predicates referring to the same semantic relation, or arguments referring to the same entity. For example, we consider two extractions (*Einstein*; *was born in*; *Ulm*), (*Ulm*; *is the birthplace of*; *Einstein*). These tuples are extracted from two sentences which give the same fact. If an ontology schema is given, we may obtain a unified relation, *e.g.,* (*Einstein*; *schema:birthplace*; *Ulm*). Moreover, recent OpenIE benchmarks (*e.g.,* CaRB) tend to keep as much relevant information as possible in gold tuples. Neural OpenIE systems optimized for these benchmarks likely output tuples with long arguments. To remedy, recent work [Wu *et al.*, 2018] [Vashishth *et al.*, 2018] [Pal *et al.*, 2020] proposes to canonicalize the extracted relation tuples through clustering. [Gashteovski *et al.*, 2020] explores aligning OpenIE tuples to reference knowledge bases. However, such complex remedial measures have not been fully studied. New training data, new models and new evaluation are needed, which is exactly a "whack-a-mole" situation.

4.2 Future Directions

More open. Most existing neural OpenIE solutions follow the traditional settings that extract binary or n-ary tuples at sentence-level from English texts. Recently, some work explores new extraction sources either to extend the system's capability or to improve the extraction quality. New sources can be document-level texts, multilingual corpus, or multimodal data. 1) **Beyond sentence:** DocOIE [Dong *et al.*, 2021] explores using document-level context to solve syntactic ambiguities when extracting facts at sentence-level. To facilitate further research on this topic, the authors contribute an OpenIE dataset with document-level context. Unlike the document-level relation extraction task [Yao *et al.*, 2019], document-level OpenIE does not consider extracting the facts that must be inferred from more than one sentences (*e.g.,* cross-sentence co-reference). New directions may consider extracting the facts that are inferred from multiple sentences; 2) **Beyond English:** Existing neural OpenIE systems mainly focus on English corpus. The lack of multilingual OpenIE benchmarks makes it difficult to evaluate a multilingual OpenIE system's performance. To overcome this issue, a recent work [Ro *et al.*, 2020] attempts to use machine translation tools to create multi-language corpus, from existing English benchmarks. However, the performance of back translation is difficult to guarantee, which may lead to biased evaluation. We expect high-quality human annotated benchmark to trigger more research on multilingual OpenIE. 3) **Beyond text:** Supporting extracting information from semi-structured or multi-modal data extends the capability of any extraction system including OpenIE. Openceres [Lockard *et al.*, 2019] defines an OpenIE task on semi-structured websites. It utilizes the structure information to determine predicate and argument in table-like sources, though the method is not neural-based. It is also common that many web documents include images to clarify some concepts. Image itself may also contain relations. We expect future OpenIE research to explore structure and layout information in semi-structured documents, and multi-modal data.

More focused. Classic OpenIE definition requires extracting all facts from the source text. However, in many scenarios, we are only interested in the facts that are related to certain topics/entities. The latter can be predefined. For example, in the task of question answering, we focus more on the facts that are related to the entities mentioned in questions, rather than all facts found in the context. Yu *et al.* 2021b propose the concept of semi-open information extraction which restricts the subject of extractions to some entities. This definition allows OpenIE systems to focus on the facts that are directly related to predefined entities of interests. Some other work introduces more restrictions on the extraction scope and application scenarios. Assertion-based question answering (ABQA) [Yan *et al.*, 2018] and NeurON [Bhutani *et al.*, 2019] extract facts from Question and Answering (QA) datasets. Here, the facts are restricted to those that answer a question. For the choices of application scenarios, ABQA targets passage-level QA data while NeurON targets conversational QA data. We expect that future work may evaluate the relatedness of extractions according to configured application scenarios, and keep those which are relevant to the application. As the result, the extractions are more focused and readily usable for downstream tasks.

More unified. OpenIE can be viewed as the most general IE task, because it includes almost all IE capabilities, such as entity recognition, relation understanding, element matching, and so on. However, we regret to see that the IE community has not made full use of OpenIE to build a bridge between IE tasks, for a unified super IE model. In our vision, OpenIE will become a basic pre-training objective for universal IE, leveraging its openness and generality to help a model understands what entities, relations, and facts are.

5 Conclusion

This survey aims to review recent progress in neural OpenIE solutions. To the best of our knowledge, we are the first to offer a comprehensive review of the neural OpenIE solutions. We divide the neural OpenIE models into two categories: tagging-based and generative models, based on their task formulation. After presenting and comparing solutions in the two categories, we brief work on calibrating neural OpenIE models. In addition, we discuss the challenges of neural OpenIE solutions and outline the future directions. We hope this survey can help new researchers build a comprehensive understanding of the existing neural OpenIE solutions, and inspire new development in this field.

Acknowledgments

This research is supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), and Nanyang Technological University, Singapore.

References

[Angeli *et al.*, 2015] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *ACL*, pages 344–354, 2015.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Berant *et al.*, 2011] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *ACL*, pages 610–619, 2011.

[Bhardwaj *et al.*, 2019] Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. CaRB: A crowdsourced benchmark for open IE. In *EMNLP-IJCNLP*, pages 6262–6267, 2019.

[Bhutani *et al.*, 2019] Nikita Bhutani, Yoshihiko Suhara, Wang-Chiew Tan, Alon Halevy, and H. V. Jagadish. Open information extraction from question-answer pairs. In *NAACL*, pages 2294–2305, 2019.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

[Claro *et al.*, 2019] Daniela Barreiro Claro, Marlo Souza, Clarissa Castellã Xavier, and Leandro Souza de Oliveira. Multilingual open information extraction: Challenges and opportunities. *Inf.*, 10(7):228, 2019.

[Cui *et al.*, 2018] Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. In *ACL*, pages 407–413, 2018.

[Del Corro and Gemulla, 2013] Luciano Del Corro and Rainer Gemulla. Clausie: Clause-based open information extraction. In *WWW*, page 355–366, 2013.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[Dong *et al.*, 2021] Kuicai Dong, Yilin Zhao, Aixin Sun, Jung-Jae Kim, and Xiaoli Li. Docoie: A document-level context-aware dataset for openie. In *ACL/IJCNLP*, pages 2377–2389, 2021.

[Fader *et al.*, 2014] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *KDD*, pages 1156–1165, 2014.

[Fu *et al.*, 2019] Cong Fu, Tong Chen, Meng Qu, Woojeong Jin, and Xiang Ren. Collaborative policy learning for open knowledge graph reasoning. In *EMNLP-IJCNLP*, pages 2672–2681, 2019.

[Gashteovski *et al.*, 2020] Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. On aligning OpenIE extractions with knowledge bases: A case study. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 143–154, 2020.

[Glauber and Claro, 2018] Rafael Glauber and Daniela Barreiro Claro. A systematic mapping study on open information extraction. *Expert Syst. Appl.*, 112:372–387, 2018.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[Groth *et al.*, 2018] Paul Groth, Michael Lauruhn, Antony Scerri, and Ron Daniel Jr. Open information extraction on scientific text: An evaluation. In *COLING*, pages 3414–3423, 2018.

[Gu *et al.*, 2016] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, pages 1631–1640, 2016.

[Han and Wang, 2021] Jiabao Han and Hongzhi Wang. Generative adversarial networks for open information extraction. *Advances in Computational Intelligence*, 1(4):6, 2021.

[He *et al.*, 2015] Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*, pages 643–653, 2015.

[Jiang *et al.*, 2019] Zhengbao Jiang, Pengcheng Yin, and Graham Neubig. Improving open information extraction via iterative rank-aware learning. In *ACL*, pages 5295–5300, 2019.

[Kolluru *et al.*, 2020a] Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *EMNLP*, pages 3748–3761, 2020.

[Kolluru *et al.*, 2020b] Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. IMoJIE: Iterative memory-based joint open information extraction. In *ACL*, pages 5871–5886, 2020.

[Léchelle *et al.*, 2019] William Léchelle, Fabrizio Gotti, and Philippe Langlais. Wire57 : A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15, 2019.

[Li *et al.*, 2022] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.*, 34(1):50–70, 2022.

[Lockard *et al.*, 2019] Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. OpenCeres: When open information extraction meets the semi-structured web. In *NAACL*, pages 3047–3056, 2019.

[Mausam, 2016] Mausam. Open information extraction systems and downstream applications. In *IJCAI*, pages 4074–4077, 2016.

[Niklaus *et al.*, 2018] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. In *COLING*, pages 3866–3878, 2018.

[Pal *et al.*, 2020] Koninika Pal, Vinh Thinh Ho, and Gerhard Weikum. Co-clustering triples from open information extraction. In *ACM IKDD*, pages 190–194, 2020.

[Ro *et al.*, 2020] Youngbin Ro, Yukyung Lee, and Pilsung Kang. Multi^{2}OIE: Multilingual open information extraction based on multi-head attention with BERT. In *EMNLP*, pages 1107–1117, 2020.

[Roy *et al.*, 2019] Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. Supervising unsupervised open information extraction models. In *EMNLP-IJCNLP*, pages 728–737, 2019.

[Saha and Mausam, 2018] Swarnadeep Saha and Mausam. Open information extraction from conjunctive sentences. In *COLING*, pages 2288–2299, 2018.

[Schneider *et al.*, 2017] Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. Analysing errors of open information extraction systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 11–18, 2017.

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NIPS*, pages 2377–2385, 2015.

[Stanovsky and Dagan, 2016] Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *EMNLP*, pages 2300–2305, 2016.

[Stanovsky *et al.*, 2015] Gabriel Stanovsky, Ido Dagan, and Mausam. Open IE as an intermediate structure for semantic tasks. In *ACL*, pages 303–308, 2015.

[Stanovsky *et al.*, 2018] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *NAACL*, pages 885–895, 2018.

[Sun *et al.*, 2018] Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. Logician: A unified end-to-end neural approach for open-domain information extraction. In *WSDM*, pages 556–564, 2018.

[Tang *et al.*, 2020] Jialong Tang, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, Xinyan Xiao, and Hua Wu. Syntactic and semantic-driven learning for open information extraction. In *EMNLP*, pages 782–792, 2020.

[Tu *et al.*, 2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *ACL*, pages 76–85, 2016.

[Vashishth *et al.*, 2018] Shikhar Vashishth, Prince Jain, and Partha P. Talukdar. CESI: canonicalizing open knowledge bases using embeddings and side information. In *WWW*, pages 1317–1327, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Williams, 1992] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992.

[Wu *et al.*, 2018] Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, and Pengcheng Yin. Towards practical open knowledge base canonicalization. In *CIKM*, pages 883–892, 2018.

[Yan *et al.*, 2018] Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. Assertion-based QA with question-aware open information extraction. In *AAAI*, pages 6021–6028, 2018.

[Yang *et al.*, 2020] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation. *CoRR*, abs/2002.07526, 2020.

[Yao *et al.*, 2019] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In *ACL*, pages 764–777, 2019.

[Yu *et al.*, 2021a] Bowen Yu, Yucheng Wang, Tingwen Liu, Hongsong Zhu, Limin Sun, and Bin Wang. Maximal clique based non-autoregressive open information extraction. In *EMNLP*, pages 9696–9706, 2021.

[Yu *et al.*, 2021b] Bowen Yu, Zhenyu Zhang, Jiawei Sheng, Tingwen Liu, Yubin Wang, Yucheng Wang, and Bin Wang. Semi-open information extraction. In *WWW*, pages 1661–1672, 2021.

[Zhan and Zhao, 2020] Junlang Zhan and Hai Zhao. Span model for open information extraction on accurate corpus. In *AAAI*, pages 9523–9530, 2020.