BenchIE: A Framework for Multi-Faceted Fact-Based Open Information Extraction Evaluation

Kiril Gashteovski[1], **Mingying Yu**[1,2], **Bhushan Kotnis**[1], **Carolin Lawrence**[1], **Mathias Niepert**[1,4], **Goran Glavaš**[2,3]
[1]NEC Laboratories Europe GmbH, Heidelberg, Germany
[2]University of Mannheim, [3]LMU Munich, Germany
[4]University of Stuttgart, Germany
firstname.lastname@neclab.eu
goran@informatik.uni-mannheim.de

Abstract

Intrinsic evaluations of OIE systems are carried out either manually—with human evaluators judging the correctness of extractions—or automatically, on standardized benchmarks. The latter, while much more cost-effective, is less reliable, primarily because of the *incompleteness* of the existing OIE benchmarks: the ground truth extractions do not include all acceptable variants of the same fact, leading to unreliable assessment of the models' performance. Moreover, the existing OIE benchmarks are available for English only. In this work, we introduce BenchIE: a benchmark and evaluation framework for comprehensive evaluation of OIE systems for English, Chinese, and German. In contrast to existing OIE benchmarks, BenchIE is *fact-based*, i.e., it takes into account informational equivalence of extractions: our gold standard consists of *fact synsets*, clusters in which we exhaustively list all acceptable surface forms of the same fact. Moreover, having in mind common downstream applications for OIE, we make BenchIE *multi-faceted*; i.e., we create benchmark variants that focus on different facets of OIE evaluation, e.g., compactness or minimality of extractions. We benchmark several state-of-the-art OIE systems using BenchIE and demonstrate that these systems are significantly less effective than indicated by existing OIE benchmarks. We make BenchIE (data and evaluation code) publicly available.[1]

1 Introduction

Open Information Extraction (OIE) is the task of extracting relations and their arguments from natural language text in a schema-free manner (Banko et al., 2007). Consider the sentence *"Sen. Mitchell, who is from Maine, is a lawyer."*; an OIE system is expected to extract the triples *("Sen. Mitchell"; "is from"; "Maine")* and *("Sen. Mitchell"; "is"; "a lawyer")* from the sentence. OIE systems are used in many downstream tasks, including knowledge graph (KG) population (Gashteovski et al., 2020), open link prediction (Broscheit et al., 2020), and question answering (Yan et al., 2018). These downstream tasks lend themselves as natural setups for extrinsic OIE evaluation (Mausam, 2016). While valuable in concrete applications, such extrinsic evaluations do not measure the *intrinsic correctness* of the extracted facts: for that purpose, several benchmarks for intrinsic OIE evaluation have been proposed (Stanovsky and Dagan, 2016; Lechelle et al., 2019; Bhardwaj et al., 2019).

Automated benchmark evaluations are more feasible (i.e., faster and cheaper) than manual OIE evaluations (Hohenecker et al., 2020). The current benchmarks, however, use scoring functions that are based on approximate (token-level) matching of system extractions against ground truth facts, which seems to be substantially less reliable than human judgments of extraction correctness (Zhan and Zhao, 2020). This primarily stems from the *incompleteness* of existing OIE benchmarks: the gold standard extractions do not include *all* acceptable surface realizations of the *same fact*. Consider, for example, a sentence from the recent evaluation framework CaRB (Bhardwaj et al., 2019): *"Sen. Mitchell is confident he has sufficient votes to block such a measure with procedural actions"*; with the gold triple extraction (*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient votes to . . . procedural actions"*). Intuitively, a system extraction with a more concise object—(*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient votes"*)—could also be accepted, as it still captures the same core piece of knowledge, and would arguably be valuable in most downstream tasks.

To account for this, existing benchmarks credit system extractions for per-slot lexical overlap with gold extractions. Such scoring is overly lenient and overestimates the systems' ability to extract correct *knowledge facts*. Consider, e.g., a system

[1]https://github.com/gkiril/benchie

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4472 - 4490

extraction (*"Sen. Mitchell"*; *"is confident he has"*; *"procedural actions"*) for the above-mentioned sentence. From the *factual* perspective, this extraction is clearly incorrect (*Sen. Mitchell* has *votes*, not *actions*). However, the popular CaRB benchmark with its token-level metrics would judge the extraction as having (1) perfect precision, since all extracted tokens can be found in corresponding slots of a gold extraction and (2) high recall, as all of the gold subject and predicate tokens as well as two gold object tokens (*"procedural"* and *"actions"*) are found within corresponding slots of the system extraction (Table 1). Moreover, by providing a single ground truth extraction per fact, existing OIE benchmarks fail to acknowledge that different downstream applications focus on different facets (i.e., aspects) of OIE extractions: e.g., for text summarization, one may prefer *minimal* extractions (Ponza et al., 2018), whereas knowledge base population benefits from strict correctness of entities in subject and object slots (Lin et al., 2020).

In this work, we depart from lenient OIE evaluations based on per-slot token overlaps and propose BenchIE, a novel *fact-centric* and *multi-faceted* OIE evaluation framework and benchmark at the core of which is the following question:

Does the system extraction express the same fact (i.e., the same unit of knowledge) as any of the ground truth extractions (and vice versa) w.r.t. the specific aspect of the OIE extraction that is of interest for one or more downstream applications?

Contributions. BenchIE advances the state of the art in OIE evaluation in the following: **(1)** it is the first **fact-centered** approach to OIE evaluation: to reliably answer the above question, we exhaustively list all correct extractions of the same fact. In contrast to existing benchmarks, BenchIE specifies *complete* sets of fact-equivalent extractions (dubbed *fact synsets*), allowing us to avoid error-prone evaluation based on token overlap measures; **(2)** BenchIE is the first **multi-faceted** OIE benchmark, allowing to test systems for different aspects of OIE extractions that may be relevant in concrete downstream applications; **(3)** BenchIE is a **multilingual** benchmark, covering English, Chinese, and German, and to the best of our knowledge the first with manually annotated (i.e., gold standard) extractions in all languages;[2] **(4)** finally, as a fact-based and multi-faceted benchmark, BenchIE allows us to perform what we believe to be the most comprehensive **profiling and comparative evaluation** of OIE systems. BenchIE portrays fact extraction abilities of six state-of-the-art OIE models much less favorably and points to their limitations that cannot be detected with existing benchmarks.

2 Matching Facts, Not Tokens

Most OIE systems extract *(subject, predicate, object)* triples, with concepts as subjects and objects and verb phrases (VPs) as predicates (Banko et al., 2007; Stanovsky et al., 2018; Lauscher et al., 2019; Gashteovski et al., 2017, 2019), though systems producing n-ary (Akbik and Löser, 2012), nested (Bhutani et al., 2016), and noun-mediated extractions (Yahya et al., 2014) also exist. Here we follow the most common practice and focus on VP-mediated facts. Our novel fact-based benchmark and evaluation paradigm can, however, equally be applied to other types of extractions (e.g., Friedrich et al. (2022) used this fact-based concept for OIE to create gold annotations for *NE-Centric OIE triples*; i.e., triples where each argument is a named entity and the relations could be either verb phrases or noun phrases).

2.1 Fact Synsets

We introduce the general concept of a *fact synset*: a set of *all* possible extractions (i.e., different surface forms) for a given fact type (e.g., VP-mediated facts) that are instances of the same fact. E.g., given the input sentence from Table 2, the extractions (*"Sen. Mitchell"*; *"has sufficient votes to block"*; *"such a measure"*) and (*"Sen. Mitchell"*; *"has sufficient votes to block"*; *"measure"*) capture the same fact and thus belong to the same fact synset.

Existing benchmarks fail to exhaustively list all acceptable extractions for the same fact. This is precisely why, in order to avoid penalizing systems for correct extractions that are not exactly the same as the gold triples, they resort to lenient token-based performance measures prone to two types of errors: (1) they punish correct fact extractions that have limited lexical overlap with the gold extraction of the same fact, e.g., (*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient votes"*) vs. (*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient votes to . . . procedural actions"*)

[2] Ro et al. (2020) introduce a multilingual version of the CaRB dataset by machine translating both sentences and extractions. However, automated translation seems to be highly unreliable for OIE – as shown by Kotnis et al. (2022), up to 70% of sentence or extraction translations obtained this way were incorrect.

Input sentence: *"Sen. Mitchell is confident he has sufficient votes to block such a measure with procedural actions."* **CaRB golden extraction:** *("Sen. Mitchell"; "is confident he has"; "sufficient votes to block ...procedural actions")*					
	OIE extraction		CaRB (P / R)		BenchIE
t_1	*("Sen. Mitchell"; "is confident he has";*	*"sufficient")*	1.00	0.44	0
t_2	*("Sen. Mitchell"; "is confident he has";*	*"sufficient actions")*	1.00	0.50	0
t_3	*("Sen. Mitchell"; "is confident he has";*	*"sufficient procedural actions")*	1.00	0.56	0
t_4	*("Sen. Mitchell"; "is confident he has";*	*"sufficient votes")*	1.00	0.50	1

Table 1: Difference in scores between CaRB and BenchIE. For the input sentence, CaRB provides only one extraction which covers all the words in the sentence. Then, for each input OIE extraction (from t_1 to t_4) it calculates token-wise precision and recall scores w.r.t. the golden annotation. In contrast, BenchIE provides 46 gold extractions for the same sentence and recognizes OIE extractions as valid if they exactly match any of them.

Input sentence: *"Sen. Mitchell is confident he has sufficient votes to block such a measure with procedural actions."*			
f_1	*("Sen. Mitchell" | "he";*	*"is";*	*"confident [he has sufficient ... actions]")*
f_2	*("Sen. Mitchell" | "he";*	*"is confident he has";*	*"sufficient votes")*
	("Sen. Mitchell" | "he";	*"is confident he has";*	*"suff. votes to block [such] [a] measure")*
f_3	*("Sen. Mitchell" | "he";*	*"is confident he has sufficient votes to block"*	*"[such] [a] measure")*
	("Sen. Mitchell" | "he";	*"is confident he has ... to block [such]";*	*"[a] measure")*
	("Sen. Mitchell" | "he";	*"is confident he has ... to block [such] [a]";*	*"measure")*
f_4	*("Sen. Mitchell" | "he";*	*"is confident he has ... [such] [a] measure with";*	*"procedural actions")*
	("Sen. Mitchell" | "he";	*"is confident he has ... [such] [a] measure";*	*"with procedural actions")*

Table 2: An example sentence with four BenchIE fact synsets (f_1–f_4). BenchIE accounts for entity coreference and accepts triples with both *"Sen. Mitchell"* and *"he"* as subjects: the delimiter "|" is just a shorthand notation for different extractions. Similarly, the square brackets ([]) represent a shorthand notation for multiple extractions: triples both with and without the expression(s) in the brackets are considered correct.

and (2) they reward incorrect extractions that have high lexical overlap with a gold extraction, e.g., (*"Sen. Mitchell"*; *"is confident he has"*; *"procedural actions"*) vs. (*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient votes to block... with procedural actions"*).

To prevent this, BenchIE relies on *exact matching* of system extractions against the gold fact synsets. Further, some OIE systems (over)generate extractions of the same fact; e.g., (*"Sen. Mitchell"*; *"has sufficient votes to block"*; *"such a measure"*) and (*"Sen. Mitchell"; "has sufficient votes to block"; "measure"*). Existing evaluation procedures do not acknowledge the *fact equivalence* of extractions and consequently reward OIE systems for multiply extracting the same fact. Our evaluation based on fact synsets directly remedies these shortcomings of existing OIE benchmarks.

2.2 Annotation Process

English Benchmark. To make BenchIE comparable to previous benchmarks, we annotate fact synsets on a subset of sentences from CaRB (Bhardwaj et al., 2019). Because exhaustive annotation of fact synsets is time consuming, we carried it on 300 (out of 1,200) randomly sampled CaRB sentences. To collect truly exhaustive fact synsets, two expert annotators independently labeled the selected 300 sentences in three rounds. **(1)** Each annotator first (independently) manually denoted every extraction in which a VP-predicate connects two concepts. The annotator then grouped the fact-equivalent triples into fact synsets.[3] To speed the annotation process up, we developed a dedicated web-based annotation tool AnnIE that facilitates the extraction of VP-mediated triples (e.g., we color-code verbs to indicate possible predicate heads) and their clustering into fact synsets;[4] **(2)** The annotators then carefully examined all gold extractions from the original CaRB dataset and added those judged to be correct, yet missing from the manually labeled fact synsets from the previous step; **(3)** Finally, each annotator compared the extractions of all OIE systems in evaluation (see §4) against the BenchIE's fact synsets (i.e., the result of the first two steps). Any system extraction not found in BenchIE was carefully examined and—if judged to be correct—added to the appropriate fact synset.[5] Finally, the

[3] We provide the annotation guidelines in Appendix A.1.

[4] We show AnnIE's interface in Appendix B. For further details about the tool, see Friedrich et al. (2022).

[5] Very few extractions were actually added in steps (2) and (3); i.e., there were very few correct extractions (from CaRB gold standard and output of OIE systems) that the annotators missed during manual annotation of fact synsets.

two annotators merged their independently created annotations by discussing and jointly resolving the disagreements. The overall annotation effort for the English dataset amounted to 80 hours per annotator. English BenchIE contains 136,357 unique gold extractions, grouped into 1,350 fact synsets. For comparison, CaRB (Bhardwaj et al., 2019) lists mere 783 gold triples for the same 300 sentences. Table 2 shows fact synsets for an example sentence.

Inter-Annotator Agreement (IAA). To validate BenchIE's annotations, we measure the inter-annotator agreement (IAA) between our two expert annotators. To this end, we quantify the agreement via *recall at the fact level* (see §2.3 for further details): for each annotator, we compute their fact-level recall as the percentage of fact synsets of the other annotator they *cover* with their extractions.[6] We average the fact-level recalls of the two annotators as the IAA score. We observed a high IAA score of 0.79. Upon manual inspection, we found that the annotators mostly agree on fact-synset level; most of the the disagreements are on extractions level (particularly, from marking the optional tokens within an extraction; see Appendix A.1.3 for details about the optional tokens).

Chinese and German Benchmarks. Two bilingual expert annotators – native in the target language and fluent in English (EN) – translated the original 300 English sentences to Chinese (ZH) and German (DE), respectively. Then, to collect exhaustive fact synsets in ZH and DE, they followed the same three annotation rounds described for §2.2. Due to substantial (primarily syntactic) differences compared to EN, we adjusted the annotation guidelines for these languages (see the Appendix A.2 and A.3 for more details). The statistics (number of fact synsets and extractions) of the ZH and DE benchmarks are given in Table 3. Compared to EN BenchIE, the ZH benchmark contains significantly fewer fact synsets (994 compared to 1,350) and more than two orders of magnitude fewer extractions. The drastically smaller number of extractions is primarily due to the lack of determiners and articles in Chinese. Their frequent occurrence in English combined with their neutrality w.r.t. extractions' correctness results in many mutually different yet fact-equivalent extractions. The numbers for German are, expectedly, much closer to those for English.

[6] An extraction *covers* a fact synset if it exactly matches any of the synset's (fact-equivalent) gold triples.

	#Extractions	#Synsets	#Extr. / Synset
CaRB	783	/	/
BenchIE EN	136,357	1,350	101.0
BenchIE DE	82,260	1,086	75.7
BenchIE ZH	5,318	994	5.4

Table 3: Multilingual BenchIE: Extraction statistics.

2.3 Evaluation Measure

We assume that BenchIE is (1) *complete*, i.e., that it contains (a) *all* VP-mediated facts expressed in input sentences and (b) for each fact, its every acceptable extraction as well; and (2) *sound*, i.e., that it does not contain any incorrect extraction that would capture a fact not stated in the sentence. Such a complete OIE gold standard enables not only a more reliable evaluation of OIE systems by means of exact matching, but also an evaluation at the more meaningful level of knowledge facts, rather than at the level of individual triples.

Concretely, we consider a system extraction to be correct if and only if it exactly matches some gold extraction from some fact synset. The number of *true positives (TPs)* is the number of fact synsets (i.e., different facts) *covered* by (at least one of the) system extractions. This way, a system that extracts N different triples of the same fact, will be rewarded only once for the correct extraction of the fact. BenchIE's false negatives (FNs) are then, intuitively, fact synsets not covered by any of the system extractions. Finally, each system extraction that does not exactly match any gold triple (from any synset) counts as a false positive (FP). We then compute *Precision*, *Recall*, and F_1 score (as the final score) from TP, FP, and FN in standard fashion.

3 Multi-Faceted OIE Benchmark

Different downstream applications care about different aspects of OIE extractions. For IE-based text summarization and simplification (Ponza et al., 2018; Štajner and Glavaš, 2017), e.g., triples should be minimal overall, across all slots (i.e., without unnecessary tokens), but the exact token placement across the slots (e.g., if a preposition is in the predicate or object) does not matter. For entity linking and knowledge base population (Lin et al., 2020), in contrast, the token placement between slots is critical: a token that is not part of an entity, should not be placed into subject or object. Acknowledging this, we create three additional variants of the English BenchIE, referred to as *facets*, each

Input sentence: *"Sen. Mitchell is confident he has sufficient votes to block such a measure with procedural actions."*	
BenchIE-E	*("Sen. Mitchell" | "he"; "is confident he has ... [such] [a] measure with"; "procedural actions")*
BenchIE-C	*"(Sen. Mitchell | he) is confident he has sufficient votes to block [such] [a] measure with procedural actions"*
BenchIE-M	*("Sen. Mitchell" | "he"; "is confident he has sufficient votes to block measure with"; "procedural actions")* *("Sen. Mitchell" | "he"; "is confident he has sufficient votes to block measure"; "with procedural actions")*

Table 4: Illustration of BenchIE's *facets* for one fact synset (f_4 from Table 2): all *acceptable* surface realizations under each facet are shown. "|" and square brackets have the same shorthand notation purpose as in Table 2.

corresponding to one aspect that is relevant in common OIE applications. This effort addresses recent calls for multi-dimensional analysis of NLP systems (Ethayarajh and Jurafsky, 2020; Narayan et al., 2021) and is well-aligned with recent efforts that create multi-faceted benchmarks for other NLP tasks (Liu et al., 2021; Väth et al., 2021) and datasets (Xiao et al., 2022).

3.1 BenchIE-E

The default, general-purpose BenchIE facet from the previous section was designed to be somewhat tolerant to token distribution accross slots (see Appendix A.1.2 for details): some tokens may be placed in either the predicate or object (e.g., the preposition *with* in the synset f_4 in Table 2). This enables a more flexible comparison of OIE systems that are designed for different purposes (i.e., systems that produce slightly different token placements are not punished) and is in line with prior work on intrinsic OIE evaluation, both automatic (Stanovsky and Dagan, 2016; Bhardwaj et al., 2019) and manual (Fader et al., 2011; Del Corro and Gemulla, 2013; Gashteovski et al., 2017). Such extraction flexibility, however, may not be desirable in tasks like automated KG construction (Wolfe et al., 2017; Jiang et al., 2019) or entity linking (Lin et al., 2020, 2021). Angeli et al. (2015) show empirically that extractions with wholesome entities and without additional tokens yield benefits in KG construction.

Since OIE is predominantly used for KG-related tasks (Weikum et al., 2020), it is paramount to have an evaluation facet that imposes strict(er) token boundaries on entity slots – subjects and objects. We thus create the *entity facet* of the benchmark (BenchIE-E) with this additional constraint of wholesomeness of subject and object concepts. BenchIE-E was constructed by one of our annotators (see §2.2) by removing from EN BenchIE's fact synsets the extractions in which subject and/or object was not a wholesome concept (see Table 4).

3.2 BenchIE-C

The default BenchIE facet (§2) compares OIE extractions against gold triples from fact synsets at the slot level: to be judged correct, an extraction must exactly match some gold triple in all slots. This criterion, however, is overly strict if extractions are to be used in applications like summarization or simplification (Ponza et al., 2018; Štajner and Glavaš, 2017), which commonly concatenate the content of the slots. In this case, it does not matter if a sequence of tokens occurs at the end of the subject or beginning of the predicate (analogously for predicate and object). To reflect this, we introduce the *concatenation facet*, BenchIE-C: for each gold BenchIE triple, we create the gold BenchIE-C utterance by simply concatenating the content of the triple's slots (see Table 4).

3.3 BenchIE-M

Our third additional evaluation facet addresses the aspect of *minimality* of OIE extractions (Gashteovski et al., 2017). More compact extractions can benefit both text generation (Ponza et al., 2018; Štajner and Glavaš, 2017) and KG-related tasks (Lin et al., 2020, 2021). If two triples t_1 and t_2 capture the same fact (i.e., are in the same fact synset), t_1 is considered *more compact* than t_2 if tokens of each t_1 slot make a (non-strict) subsequence of tokens in the corresponding t_2 slot (Gashteovski, 2020).[7] To allow for evaluation of minimality, BenchIE-M triples contain only the non-optional tokens (denoted in square brackets in Table 2) from the corresponding BenchIE triple. Consequently, BenchIE-M fact synsets on average contain many fewer extractions than the original BenchIE synsets.[8]

4 Fact-Level Evaluation

We first compare BenchIE's fact-level evaluation (i.e., default facet, §2) against CaRB's token-level

[7] At least one t_1 slot has to be a strict subsequence of the respective t_2 slot; t_1 and t_2 would be the same otherwise.

[8] This does not imply that each fact synset in BenchIE-M contains only one (i.e., minimal) triple (see Table 4).

		EN							ZH	DE
		Naive OIE	ClausIE	MinIE	Stanford	ROIE	OpenIE6	M^{2}OIE	M^{2}OIE	M^{2}OIE
P	CaRB	0.24	0.58	0.45	0.17	0.44	0.48	**0.60**	/	/
	BenchIE	0.03	**0.50**	0.43	0.11	0.20	0.31	0.39	0.18	0.09
	Δ	+0.21	+0.08	+0.02	+0.06	**+0.24**	+0.17	+0.21	/	/
R	CaRB	**0.70**	0.53	0.44	0.29	0.60	0.67	0.61	/	/
	BenchIE	0.02	0.26	**0.28**	0.16	0.09	0.21	0.16	0.10	0.03
	Δ	**+0.68**	+0.27	+0.16	+0.13	+0.51	+0.46	+0.45	/	/
F_1	CaRB	0.36	0.56	0.44	0.22	0.51	0.56	**0.61**	/	/
	BenchIE	0.03	**0.34**	**0.34**	0.13	0.13	0.25	0.23	0.13	0.04
	Δ	+0.33	+0.22	+0.10	+0.09	**+0.38**	+0.31	**+0.38**	/	/

Table 5: Comparison of performance of OIE systems on BenchIE and CaRB benchmarks for precision (P), recall (R) and F_1 score (F_1). The row Δ indicates the difference between CaRB score and BenchIE score (i.e., $\Delta = CaRB - BenchIE$). **Bold numbers** indicate highest score per row (i.e., highest score for P / R / F_1 per benchmark) or highest score difference per row (i.e., highest Δ for P / R / F_1).

scoring (Bhardwaj et al., 2019).[9] Our quantitative results confirm our intuitions and observations (see Table 1): CaRB systematically and substantially overestimates OIE systems' performance. BenchIE, we argue, portrays the fact extraction abilities of OIE systems more realistically.

4.1 Experimental Setup

OIE Systems. We tested six widely used OIE systems that extract VP-mediated facts for EN, namely: ClausIE (Del Corro and Gemulla, 2013), Stanford OIE (Angeli et al., 2015), MinIE (Gashteovski et al., 2017), ROIE (Stanovsky et al., 2018), OpenIE 6 (Kolluru et al., 2020) and M^{2}OIE (Ro et al., 2020). We additionally implemented the following naive baseline (Naive OIE): each verb (detected using spaCy's POS-tagger (Honnibal and Montani, 2017)) becomes the predicate, its entire preceding sentence context becomes the subject and succeeding context the object. For ZH and DE, we evaluated a supervised M^{2}OIE (Ro et al., 2020) model based on the multilingual BERT (Devlin et al., 2019), trained on a large EN dataset (Zhan and Zhao, 2020) and transferred (zero-shot) to target languages by means of its multilingual encoder.

Implicit and N-ary Extractions. Some OIE systems produce implicit extractions containing tokens that do not occur in the sentence.[10] As BenchIE does not contain implicit annotations, we remove such extractions from the OIE systems' output, to avoid penalizing OIE systems for extracting fact types not covered by the benchmark. To make CaRB directly comparable, we automatically remove all its implicit extractions too. ROIE and M^{2}OIE produce N-ary extractions (i.e., more than three slots), whereas BenchIE contains only triples. We follow standard practice (Del Corro and Gemulla, 2013) and convert those extractions into triples by concatenating the third and subsequent slots into a single object.

4.2 Results and Discussion

Table 5 summarizes results of OIE systems on BenchIE and CaRB. Across the board, BenchIE's fact-level precision and recall are significantly lower than CaRB's respective precision and recall computed on token level. On average, CaRB scores the OIE systems higher than BenchIE by 14 percentage points for precision, 38 percentage points for recall and 26 percentage points for the F_1 score.

Precision. System's precision on BenchIE is lower (albeit not so drastically lower as recall) than on CaRB because BenchIE, as a complete benchmark, punishes *incorrect facts*, i.e., extractions that cannot be found in BenchIE's fact synsets. CaRB, on the other hand, rewards any token overlap that the incorrectly extracted fact has against its gold triple(s) – in many cases such overlap is substantial and CaRB consequently rewards the incorrect fact with high precision. Consider, for example, the sentence from Table 1 and an incorrect fact extraction (*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient actions"*); on BenchIE, this extraction is a false positive because it does not exist in any of the four fact synsets it lists for the sentence. CaRB, in contrast, rewards the extraction with perfect precision because all its tokens are accounted for in the corresponding slots of its gold triple (*"Sen. Mitchell"*; *"is confident he has"*; *"sufficient votes to . . . actions"*).

[9] CaRB is an improved version of the widely-adopted OIE2016 benchmark (Stanovsky and Dagan, 2016); our findings for CaRB are thus likely to hold for OIE2016 as well.

[10] E.g., the triple *("Biden"; "be"; "President")* extracted from the phrase *"President Biden ..."*

In an attempt to quantify how much CaRB overestimates fact-level precision with its token overlap metric, we evaluated our Naive OIE baseline on both CaRB and BenchIE. While BenchIE reflects the poor quality of naive extractions with the near-zero performance, CaRB estimates its precision to be non-negligible (0.24) and even higher than that of the Stanford's OIE system (0.17). In contrast, BenchIE assigns much lower score to this baseline: precision of 0.03—8 times less than CaRB's score.

Recall. While CaRB somewhat overestimates fact-level precision of OIE systems, its overestimation of their recall is much more drastic: all tokens of its gold extractions that can be found in respective slots of a factually incorrect extraction of an OIE system contribute to the system's recall. The overestimation of CaRB's recall scores is best illustrated by the fact that our naive baseline (Naive OIE) obtains a score of 0.7, better than any of the six OIE systems under evaluation. In terms of recall, CaRB obviously rewards long extractions – the longer the system extraction is, the more likely it is to cover more tokens from gold standard extractions. Neural extractors OpenIE6, ROIE, and M^{2}OIE on average produce much longer extractions than rule-based systems like MinIE or Stanford (e.g., on average, a ROIE extraction has 16 tokens, whereas Stanford extraction has 7.7 tokens): accordingly, CaRB rewards the neural systems with much higher recall scores. BenchIE, on the other hand, credits only the OIE extractions that cover its fact synsets (and only once per fact synset). Our Naive OIE is, intuitively, highly unlikely to match gold extractions from fact synsets and BenchIE reflects this with a fact-level recall of only 2%. Similarly, BenchIE's recall scores reveal that the long extractions of neural OIE systems very rarely correspond to any acceptable variant of an expressed fact (e.g., ROIE's fact-level recall is only 9%).

Multilingual OIE. We evaluated M^{2}OIE (as the only multilingual model in our evaluation) on the Chinese and German versions of BenchIE. Quite expectedly, the performance for Chinese and German in target languages is below the source English performance. However, the drop due to the zero-shot language transfer is, at first glance – surprisingly, much larger for German than for Chinese: this goes against findings from other tasks, where transfer performance correlates with linguistic proximity between the source and target language (Lauscher et al., 2020). M^{2}OIE's Chinese performance is encouraging, as it surpasses the English performance of some of the other OIE models (e.g., its recall score is better than ROIE, and its precision score is better than Stanford's). We believe this is because (a) OIE is a highly syntactic task; and (b) Chinese language is syntactically simple and has the same word order as English (SVO). German language, on the other hand, despite overall linguistic proximity to English, has a different word order (SOV; from generative perspective), with the main verb often appearing at the very end of the sentence – this, we believe, is the main cause of poor OIE transfer between English and German. We believe BenchIE is a good starting point for multilingual OIE evaluation: we subsequently created additional data for Arabic, Galician, and Japanese: see Kotnis et al. (2022) and Friedrich et al. (2022) for details and further analyses.

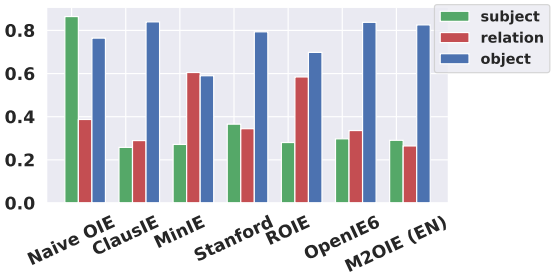

Figure 1: Relative proportion of errors per slot for OIE systems. Note that (1) fractions do not add up to 1 as extraction can be erroneous in more than one slot; and (2) the figure does not indicate systems' absolute error rates (for performance comparison, see Table 5).

5 Profiling OIE Systems with BenchIE

Token-based evaluation of existing OIE benchmarks (with real per-extraction scores in the range $[0, 1]$) makes pinpointing of extraction error source difficult. This limits their usability in automatic error analysis and system profiling. The fact that previous work performed OIE error analyses manually (Fader et al., 2011; Schneider et al., 2017) confirms this. BenchIE, in contrast, lists all acceptable extractions and thus naturally lends itself to reliable automatic error analysis and profiling.

5.1 Slot Errors

We carry out the analysis of errors per slots on the default BenchIE facet (§2), because it is application-agnostic, unlike the additional facets from §3. We observed that most of the errors in all OIE systems stem from extracting the objects (see Figure 1). For an SVO language like English,

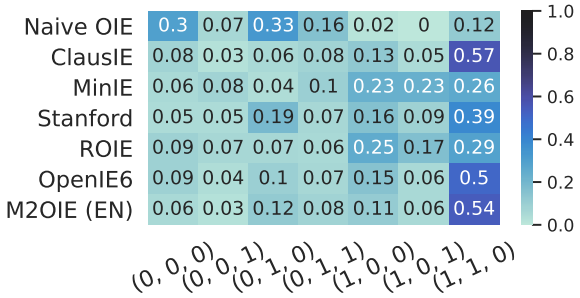

Figure 2: Distribution of incorrect extractions of OIE systems across different slot-error combinations.

correctly extracting subjects and predicates seems substantially easier than correctly extracting objects. MinIE (rule-based) and ROIE (neural) have higher shares of predicate mis-extractions. MinIE post-processes ClausIE's triples by moving words from objects to predicates. Since ClausIE most frequently makes object errors, this effectively redistributes those errors between predicates and objects of MinIE's extractions.

Figure 1, however, does not tell the whole story, as many extractions are erroneous in multiple slots. For more detailed insights, we assign each incorrect extraction to one of seven error buckets: each error bucket indicates one combination of extraction errors across the three slots. For example, the bucket $(1, 1, 0)$ contains extractions that match their *closest* gold triple in the subject and predicate, but not object. The closest gold triple is the one that matches the extraction in most slots.[11] The error-bucket analysis, summarized in Figure 2, reveals that, across all systems, most extractions with object errors actually have correct subjects and predicates (bucket $(1, 1, 0)$). MinIE deviates from this pattern and produces also many extractions with both incorrect object and predicate (bucket $(1, 0, 0)$) or only bad predicate (bucket $(1, 0, 1)$). Expectedly, most extractions of our naive baseline most often get only the predicate right (bucket $(0, 1, 0)$) or all three slots wrong (bucket $(0, 0, 0)$). This further emphasizes how misleading current token-based benchmarks can be – CaRB rewards this baseline with very high recall (see §4).

5.2 Bucketized Error Analysis

To understand where OIE systems fail systematically, we split the input sentences into buckets and measured the performance of OIE systems per bucket. Based on preliminary qualitative error analysis, we chose bucketization according to some linguistic properties of the sentences that produced erroneous triples. In particular, we examine the performance of OIE systems for sentence length, presence of conjunctions and case markers, since these appeared to be the most common reasons for failure. Note that BenchIE instances can be "bucketized" according to an arbitrary dimension interest, lending itself to diverse future fine-grained evaluations and analyses of OIE systems' behaviour. In general, we found that OIE systems exhibit weakest performance on long sentences (with more than 30 tokens) as well as those that contain conjunctions or have more than two case markers (Figure 3). For a more detailed discussion, see Appendix C.

5.3 Multi-Faceted Evaluation

Finally, we profile the OIE systems on our three special benchmark facets (§3): BenchIE-E, -C and -M. Figure 4 summarizes the performance of OIE systems on these three facets.

BenchIE-C. Ignoring slot boundaries, this facet is more lenient to OIE systems than the default facet – BenchIE-C yields higher scores than the regular BenchIE facet for *all* systems. The gap between the system's performance on BenchIE-C and BenchIE effectively quantifies how often the system misplaces tokens between adjacent slots. This gap is very small for Stanford OIE and MinIE – this means that, for extractions with correct overall token span, they also distribute the tokens between the slots correctly. For downstream tasks like text summarization, BenchIE-C results point to ClausIE as the best choice. Interestingly, we observed that CaRB's Precision for some systems (ClausIE and MinIE) effectively matches their Precision on BenchIE-C (see Figure 4), which is another indication that CaRB scores, in effect, neglect precise token distributions across slots.

BenchIE-E. This facet is stricter than the default BenchIE facet – it allows fewer token placement variants in subject and object. For all OIE systems the F_1 BenchIE-E score is thus lower than the corresponding BenchIE score. MinIE and Stanford OIE obtain very similar performance on BenchIE-C, BenchIE (default), and BenchIE-E: this means that their extraction (when correct in overall token span) most often have clean concepts in subject and object. All neural systems and ClausIE exhibit huge performance drops on BenchIE-E – this

[11] An incorrect extraction may have several "closest" gold triples that correspond to different error buckets. In this case, we increase the count for all competing buckets.

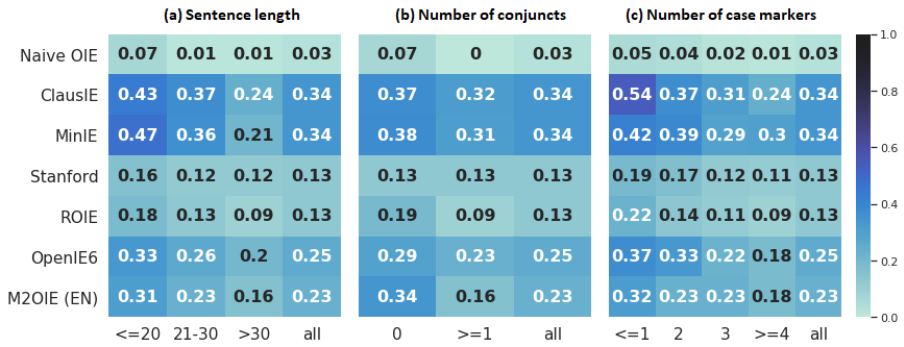

Figure 3: Bucketized experiments: F1 score according to different bucketizations of the input sentences: sentence length (a); number of conjunctions (b); number of case markers (c).

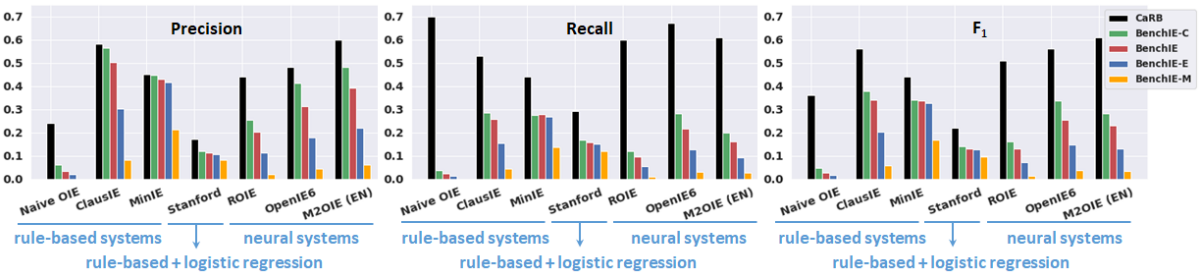

Figure 4: Multi-faceted evaluation of OIE systems.

means that their subject and object concept extractions are not clean, which makes these systems less suitable for tasks like KG population and entity linking. Out of the systems we evaluate, MinIE is the best fit for such downstream tasks.

BenchIE-M. This facet yields the lowest performance for all systems, as it punishes extractions with any unnecessary tokens. Expectedly, MinIE – a system tailored to produce minimal extractions – yields the best performance on this facet. But even MinIE "loses" half of its performance when minimality is enforced (BenchIE vs. BenchIE-M). This calls for more work on minimizing OIE extractions. Stanford OIE outperforms all systems except MinIE, which renders it a good pick when extraction minimality is beneficial for a downstream task.

Neural vs. Rule-Based Systems. Neural systems underperform their rule-based counterparts on most facets. This gap is most pronounced on BenchIE-E, whereas it is much smaller on BenchIE-C: these observations strongly indicate that neural systems struggle the most with correct distribution of tokens across the (adjacent) extraction slots. They also do not attempt to remove the optional (i.e., unnecessary) tokens, as indicated by extremely low performance on BenchIE-M. On CaRB, however, these same neural systems yield the best performance. Being trained and validated on datasets with extractions similar to CaRB's, neural extractors seem to overfit to CaRB evaluation. Our fact-based multi-faceted evaluation, however, reveals that their extractions are far less likely to be useful down the stream.

6 Conclusion

We introduced BenchIE: a benchmark for more reliable fact-level evaluation of OIE systems for English, Chinese and German. Unlike existing benchmarks, BenchIE takes into account fact-level equivalence of extractions: it consists of *fact synsets* that contain *all* acceptable surface forms of the same fact. Further, EN BenchIE is multi-faceted – it allows to evaluate OIE extractions w.r.t. several aspects relevant in common downstream tasks. Our experiments show that current benchmarks, with incomplete gold standard and approximate token-level matching, drastically overestimate fact extraction abilities of OIE systems. Currently, the limits of BenchIE are its relatively small size (300 sentences v.s. CaRB's 1,200) and its time-consuming annotation process. A promising research direction is the investigation of trade-off between the manual effort and completeness of different OIE annotation strategies. In this scenario, BenchIE is an ideal point of reference: it can precisely quantify the completeness of some larger (non-exhaustive) OIE dataset created with limited or no manual effort.

References

Alan Akbik and Alexander Löser. 2012. KrakeN: N-ary Facts in Open Information Extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX@NAACL-HLT)*, pages 52–56.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Probing Linguistic Features of Sentence-level Representations in Relation Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1534–1545.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 344–354.

Ravneet Arora, Chen-Tse Tsai, and Daniel Preoţiuc-Pietro. 2021. Identifying Named Entities as they are Typed. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 976–988.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 2670–2676.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. CaRB: A Crowdsourced Benchmark for Open IE. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6263–6268.

Nikita Bhutani, H V Jagadish, and Dragomir Radev. 2016. Nested Propositions in Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 55–64.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. Can We Predict New Facts with Open Knowledge Graph Embeddings? A Benchmark for Open Link Prediction. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, pages 2296–2308.

Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the International World Wide Web Conferences (WWW)*, pages 355–366.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Kuicai Dong, Yilin Zhao, Aixin Sun, Jung-Jae Kim, and Xiaoli Li. 2021. DocOIE: A Document-Level Context-Aware Dataset for OpenIE. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2377—-2389.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the Eye of the User: A Critique of NLP Leaderboards. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545.

Niklas Friedrich, Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, and Goran Glavaš. 2022. AnnIE: An Annotation Platform for Constructing Complete Open Information Extraction Benchmark. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL): System Demonstrations*.

Kiril Gashteovski. 2020. Compact Open Information Extraction: Methods, Corpora, Analysis. *PhD thesis*.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. MinIE: Minimizing Facts in Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2630–2640.

Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. 2020. On Aligning Openie Extractions with Knowledge Bases: A Case Study. In *Proceedings of the Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP@EMNLP)*, pages 143–154.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2019. OPIEC: An Open Information Extraction Corpus. In *Proceedings of the Conference on Automated Knowledge Base Construction (AKBC)*.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 643–653.

Patrick Hohenecker, Frank Mtumbuka, Vid Kocijan, and Thomas Lukasiewicz. 2020. Systematic Comparison of Neural Architectures and Training Approaches for Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8554–8565.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.

Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2019. The Role of "Condition": A Novel Scientific Knowledge Graph Representation and Construction Model. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1634–1642.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti, et al. 2020. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761.

Bhushan Kotnis, Kiril Gashteovski, Daniel Oñoro Rubio, Ammar Shaker, Vanesa Rodriguez-Tembras, Makoto Takamoto, Mathias Niepert, and Carolin Lawrence. 2022. MILLIE: Modular & Iterative Multilingual Open Information Extraction. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499.

Anne Lauscher, Yide Song, and Kiril Gashteovski. 2019. MinScIE: Citation-centered Open Information Extraction. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 386–387. IEEE.

William Lechelle, Fabrizio Gotti, and Phillippe Langlais. 2019. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. In *Proceedings of the Linguistic Annotation Workshop (LAW@ACL)*, pages 6–15.

Xueling Lin, Lei Chen, and Chaorui Zhang. 2021. TENET: Joint Entity and Relation Linking with Coherence Relaxation. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 1142–1155.

Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. 2020. KBPearl: A Knowledge Base Population System Supported by Joint Entity and Relation Linking. In *Proceedings of the Very Large Data Base Endowment (PVLDB)*, pages 1035–1049.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021. ExplainaBoard: An Explainable Leaderboard for NLP. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL): System Demonstrations*, pages 280—-289.

Mausam. 2016. Open Information Extraction Systems and Downstream Applications. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4074–4077.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open Language Learning for Information Extraction. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 523–534.

Avanika Narayan, Piero Molino, Karan Goel, Willie Neiswanger, and Christopher Re. 2021. Personalized Benchmarking with the Ludwig Benchmarking Toolkit. In *Conference on Neural Information Processing Systems (NeurIPS)*.

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

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multi^2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1107–1117, Online. Association for Computational Linguistics.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing Errors of Open Information Extraction Systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems@EMNLP*, pages 11–18.

Sanja Štajner and Goran Glavaš. 2017. Leveraging Event-based Semantics for Automated Text Simplification. *Expert systems with applications*, 82:383–395.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a Large Benchmark for Open Information Extraction. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2300–2305.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised Open Information Extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 885–895.

Dirk Väth, Pascal Tilli, and Ngoc Thang Vu. 2021. Beyond Accuracy: A Consolidated Tool for Visual Question Answering Benchmarking. *arXiv preprint arXiv:2110.05159*.

Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian Suchanek. 2020. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. *arXiv preprint arXiv:2009.11564*.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2017. Pocket Knowledge Base Population. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 305–310.

Yang Xiao, Jinlan Fu, Weizhe Yuan, Vijay Viswanathan, Zhoumianze Liu, Yixin Liu, Graham Neubig, and Pengfei Liu. 2022. DataLab: A Platform for Data Analysis and Intervention. *arXiv preprint arXiv:2202.12875*.

Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. 2014. ReNoun: Fact Extraction for Nominal Attributes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 325–335.

Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. 2018. Assertion-based QA with Question-Aware Open Information Extraction. In *Proceedings of the Conference of the American Association for Artificial Intelligence (AAAI)*, pages 6021–6028.

Junlang Zhan and Hai Zhao. 2020. Span Model for Open Information Extraction on Accurate Corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 9523–9530.

A Appendix: Annotation Guidelines

A.1 Annotation Guidelines for English

A.1.1 General Principle

The annotator should manually extract verb-mediated triples from a natural language sentence. Each triple should represent two entities or concepts, and the verb-mediated relation between them. For example, from the input sentence *"Michael Jordan, who is a former basketball player, was born in Brooklyn."*, there are three entities and concepts—*Michael Jordan, former basketball player* and *Brooklyn*—which are related as follows: *("Michael Jordan"; "is"; "former basketball player")* and *("Michael Jordan"; "was born in"; "Brooklyn")*.

Once the triple is manually extracted, it should be placed into the correct fact synset (see Section A.1.2).

A.1.2 Fact Synsets

Once a triple is manually extracted, the annotator should place the triple into its corresponding fact synset (for more details about the concept of fact synsets, refer to Section 2). In case there is no existing fact synset for the manually extracted triple, the annotator should create one and place the triple in that synset.

Coreference. The annotator should place extractions that refer to the same entity or concept under the same fact synset. Consider the following input sentence: *"His son , John Crozie, was an aviation pioneer."*; the following triples should be placed in the same fact synset:

- *("His son"; "was"; "[an]*[12] *aviation pioneer")*
- *("J. Crozie"; "was"; "[an] aviation pioneer")*

because *"His son"* and *"John Crozie"* refer to the same entity.

Token placements within the slots. The annotator should consider placing certain tokens in different slots, without damaging the meaning of the fact. Consider the input sentence *"Michael Jordan was born in Brooklyn."*. There is one fact synset (f_1) and its corresponding triples (t_1, t_2 and t_3):

f_1 t_1 : *("M. J."; "was born in"; "Brooklyn")*
t_2 : *("M. J."; "was born"; "in Brooklyn")*
t_3 : *("M. J."; "was"; "born in Brooklyn")*

[12] words in square brackets indicate optional tokens (see Section A.1.3)

In t_1, the preposition *"in"* is in the relation, while in t_2 it is in the object. Likewise, the annotator should allow for some flexibility w.r.t. the verbs. While the verbs and prepositions naturally belong to the relation, some OIE systems were designed with different goal in mind; e.g., to detect head verbs as relations for detecting clauses within the extractions (Del Corro and Gemulla, 2013) or to fit SRL frames for predicates (Stanovsky et al., 2018). We do not want to penalize the OIE systems for such design choices.

For BenchIE-E[13], however, this flexibility of token placements is not allowed. In particular, for f_1 the annotator is allowed to only extract t_1, while t_2 and t_3 should not be listed. Note that this is the only difference in the annotation guidelines between BenchIE-E and the standard BenchIE facet.

Passive voice. When possible, if an extraction is in passive voice, the annotator should place its active voice equivalent into the appropriate fact synset. For instance, consider the sentence *"The ball was kicked by John."*; then, the fact synset should contain the following triples:

- *("[The] ball"; "was kicked by"; "John")*
- *("John"; "kicked"; "[The] ball")*

Note that the opposite direction is not allowed. If the sentence was *"John kicked the ball."*, then the annotator is not allowed to manually extract the triple *("[The] ball"; "was kicked by"; "John")* because such extraction contains words that are not originally found in the input sentence (*"was"* and *"by"*). These are so-called implicit extractions and we do not consider them (for details, see Section A.1.8 of the appendix).

A.1.3 Optional Tokens

If possible, the annotator should label as *optional* all tokens that can be omitted in an extraction without damaging its semantics. Such tokens include determiners (e.g., *a, the, an*), honorifics (e.g., *[Prof.] Michael Jordan*) or certain quantities (e.g., *[some] major projects*. The optional tokens are marked with square brackets [ ]. In what follows, we show examples of considered optional token(s).

Determiners. Unless a determiner is a part of a named entity (e.g., *"The Times"*), it is considered as optional. For instance, the following triples are considered to be semantically equivalent:

[13] For details on BenchIE-E, see Section 3.1.

- *("Michael Jordan"; "took"; "the ball")*
- *("Michael Jordan"; "took"; "ball")*

The annotator, therefore, should annotate *("Michael Jordan"; "took"; "[the] ball")*, where the optional token is in square brackets.

Titles. Titles of people are considered optional; e.g., *("[Prof.] Michael Jordan"; "lives in"; "USA")*.

Adjectives. The annotator should label adjectives as optional if possible. For example, in the following triple, the adjective *"smart"* can be considered optional: *("Albert Einstein"; "was"; "[a] [smart] scientist")*. Note that the annotator should be careful not to label adjectives as optional if they are essential to the meaning of the triple. For instance, the adjective *"cold"* should not be labeled as optional in the triple *("Berlin Wall"; "is [infamous] symbol of"; "[the] cold war")*.

Quantities. Certain quantities that modify a noun phrase can be considered as optional; e.g., *("Mitsubishi"; "has control of"; "[some] major projects")*.

Words indicating some tenses. The annotator can treat certain verbs that indicate tense as optional. For instance, the word *"has"* in *("FDA"; "[has] approved"; "Proleukin")* can be considered as optional, since both VPs *"have approved"* and *"approved"* contain the same core meaning.

Verb phrases. It is allowed for the annotator to mark verb phrases as optional if possible; e.g. *("John"; "[continues to] reside in"; "Berlin")*.

A.1.4 Attribution Clauses

Extractions that indicate attribution of another core piece of information should be placed in separate fact synset, because they indicate a separate piece of information with separate predicate. For example, the core information of the sentence *"Conspiracy theorists say that Barack Obama was born in Kenya."* is that Barack Obama was born in Kenya. As indicated by Mausam et al. (2012), it is important for OIE systems to extract the context about the attribution of such information. Therefore, the annotator should extract the core information—the triple *("Barack Obama"; "[was] born in"; "Kenya")*—in one fact synset, and the triples indicating attribution—*("Conspiracy theorists"; "say that"; "Barack Obama was born in Kenya")*—in another.

A.1.5 Incomplete Clauses

The annotator should not extract incomplete clauses, i.e., triples that lack crucial piece of information. Suppose there is the input sentence *"He was honored by the river being named after him"*. The following triple should not be manually extracted: *("He"; "was honored by"; "[the] river")*, but the following triples should be: *("He"; "was honored by [the] river being named after"; "him")* and *("[the] river"; "being named after"; "him")*.

A.1.6 Overly Complex Extractions

The annotators should not manually extract overly specific triples, such that their arguments are complex clauses. For instance, for the input sentence *"Vaccinations against other viral diseases followed, including the successful rabies vaccination by Louis Pasteur in 1886."*, the following triple should not be extracted: *("Vaccinations against other viral diseases"; "followed"; "including the successful rabies vaccination by Louis Pasteur in 1886")* because the object is a complex clause which does not describe a single concept precisely, but rather it is composed of several concepts.

A.1.7 Conjunctions

The annotator should not allow for conjunctive phrases to form an argument (i.e., subject or object). Such arguments should be placed into separate extractions (and in separate fact synsets). Consider the sentence *"Michael Jordan and Scottie Pippen played for Chicago Bulls."*. The annotator should manually extract the following triples:

- *("M. Jordan"; "played for"; "Chicago Bulls")*
- *("S. Pippen"; "played for"; "Chicago Bulls")*

The annotator should not, however, extract *("M. J. and S. P."; "played for"; "Chicago Bulls")*.

A.1.8 Implicit Extractions

We focus on explicit extractions, which means that every word in the extracted triple must be present in the original input sentence. Therefore, implicit extractions—i.e., extractions that contain inferred information with words not found in the sentence—are not considered. One example implicit extraction is *("Michael Jordan"; "be"; "Prof.")* from the input sentence *"Prof. Michael Jordan lives in USA."*, where the triple infers that Michael Jordan is professor without being explicitly indicated in the sentence (i.e., the word *"be"* is not present in the input sentence, it is inferred).

A.2 Annotation Guidelines (Chinese)

The annotator should follow the same general principles as with the English annotation guidelines (Section A.1). Due to the language difference, we slightly adapted the annotation guidelines for the Chinese language. In what follows, we list those differences.

A.2.1 Articles

Chinese language does not contain articles (i.e., *"a", "an", "the"*). Therefore, in the manual translation of the sentences, there are no articles in the Chinese counterparts.

A.2.2 Prepositional Phrases within a Noun Phrase

Certain noun phrases with nested prepositional phrase cannot be translated directly into Chinese the same way as in English. For example, suppose we have the phrase *"Prime Minister of Australia"*. In Chinese, the literal translation of this phrase would be *"Australia's Prime Minister"*. For instance, in the English annotations the sentence *"He was the Prime Minister of Australia"* would have two fact synsets:

f_1 *("He"; "was [the] Pr. Min. of"; "Australia")*

f_2 *("He"; "was"; "[the] Pr. Min. [of Australia]")*

This is because the fact synset f_1 relates the concepts *"he"* and *"Australia"* with the relation *"was [the] Prime Minister of"*, while the second fact synset relates the concepts *"he"* and *"Prime Minister [of Australia]"* with the relation *"was"*.

In Chinese language, however, the construction of f_1 would not be possible, because the phrase *"Prime Minister of Australia"* cannot be separated into *"Prime Minister"* and *"Australia"*. Therefore, the golden annotation for this particular example in Chinese would be only one fact synset: *("He"; "was"; "[Australia's] Prime Minister")*, which is equivalent with f_2.

A.3 Annotation Guidelines (German)

In general, the annotators for German should follow the same guidelines described in Section A.1 for English. In what follows, we describe the differences which are specific for the German annotations.

A.3.1 Separable Verbs

Separable verbs (e.g., *"aufstehen"*) in German consist of a lexical core (a verb; *"stehen"*) and a separable particle (e.g., a preposition; *"auf"*). When used in a sentence, separable verbs in German are split in such manner that the separable particle goes to the end of the sentence. Consider the following sentence that contains the separable verb *"aufstehen"*: *"Ich stehe um 7 Uhr auf"*. To accommodate the verb-mediated relations, the annotator should extract the separable particle right after the separable core within the predicate: *("Ich"; "stehe auf um"; "7 Uhr")*

A.3.2 Modal Verbs

The modal verbs follow similar pattern as the separable verbs. Namely, the modal verb has the main predicate position within the sentence (directly followed by the subject), and the main verb that is modified by the modal verb is at the end of the sentence; e.g. sentence *"I must go to work"* and its German counterpart *"Ich muss zur Arbeit gehen"*. Following the same guidelines for verb-mediated predicates, the annotator should extract the modal verb together with the main verb: *("Ich"; "muss gehen zur"; "Arbeit")*.

A.3.3 Passive Voice

Consider the following English sentence written in passive voice *"The letters were sent through the messenger"* and its German counterpart *"Die Briefe wurden durch den Boten geschickt"*. Following the spirit of extractions with verb-mediated relations, the annotator should extract the following triple: *("[Die] Briefe"; "wurden geschickt durch"; "[den] Boten")*.

B Annotation Tool

To facilitate the annotation process, we developed a web-based annotation tool: AnnIE (Friedrich et al., 2022). First, the annotator is given the input sentence as a string along with its tokenized form (Figure 5). Then, the tool highlights the tokens of interest that are candidates for the slots. In particular, we highlight the verbs in one color (candidate predicates) and the nouns in another (candidate arguments).

Then, the annotator can select the tokens with a UI and place them into slots. This forms one annotated triple. Note that the annotator can also annotate for optional tokens and phrases with the

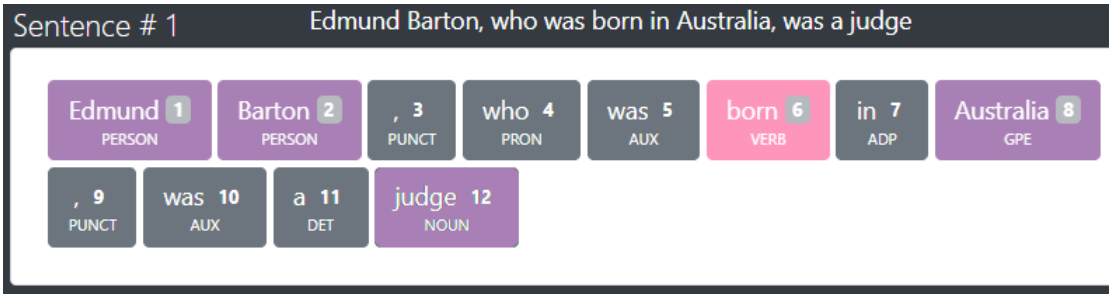

Figure 5: Highlighting tokens of interest: verbs (potential relations) and nouns (potential arguments).

Figure 6: Manual labeling of OIE triples. The user selects tokens from the tokenized input sentence and places them into the correct slot: subject (green), predicate (yellow) or object (blue). Then, the user adds the extracted triple either to an active fact cluster (i.e., fact synset) or to a new one. The user can also select which tokens are optional by clicking the "Optional" button on an active token selection.

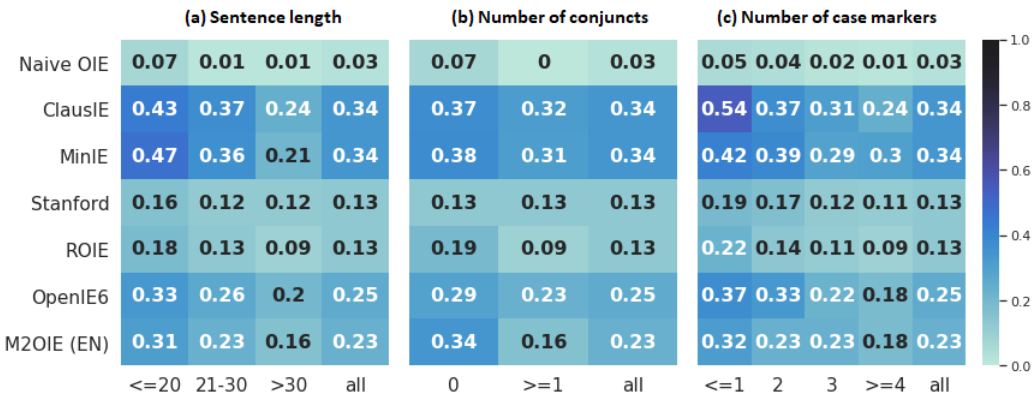

Figure 7: Bucketized experiments: F1 score according to different bucketizations of the input sentences: sentence length (a); number of conjunctions (b); number of case markers (c).

use of the mouse double click. Then, the annotator can place the newly annotated triple in either a new fact synset (cluster) or in an existing one (Figure 6). For more details on the annotation tool, see (Friedrich et al., 2022).

C Further Error Analysis

Based on preliminary qualitative error analysis, we chose bucketization according to some linguistic properties of the sentences that produced erroneous triples. In particular, we examine the performance of OIE systems for sentence length, presence of conjunctions and case markers, since these appeared to be the most common reasons for failure. Note that BenchIE allows for any type of bucketization, which can be used for diverse set of fine-grained evaluation for future research on OIE.

C.1 Sentence Length

Sentence length is a feature that can affect the performance of NLP systems for different tasks, including relation extraction (Alt et al., 2020) and named entity recognition (Arora et al., 2021). To evaluate how sentence length afffects performance of OIE systems as well, we split the sentences into three buckets: sentences shorter or equal than 20 tokens, between 21 and 30 tokens, and more than 30 tokens. The distribution of these buckets are 120, 113 and 67 sentences respectively.

We observed that shorter sentences usually yield the best performance for all OIE systems w.r.t. the F_1 score (Figure 7a). An extreme example is MinIE, which loses 26 percentage points from sentences shorter than 20 tokens to sentences longer than 30 tokens. Part of the reason why such sentences are harder to handle is because they contain more complex linguistic structures, such as conjunctions and case markers. Such sentences tend to to produce overly complex extractions that contain very complex structures in their arguments (see example extraction t_3 in Table 6).

C.2 Conjunctions

To examine the effect of the conjunctions on the performance of OIE systems, we bucketized the input sentences according to the dependency type `conj`, which relates two conjunct words in a sentence. In particular, we place the sentences with no conjuncts in one bucket, and the sentences with one or more conjuncts in another bucket. With such bucketization, half of the sentences are in the first bucket, and half in the other. We observed that the F1 score suffers when a sentence contains at least one pair of conjuncts (Figure 7b). This observation partially explains the observation from Section 5.1 that OIE systems have troubles identifying the objects correctly. In subsequent experiments, we observed that sentences with more than one conjuncts worsen the scores further compared to the sentences with one or no conjuncts. The triple t_5 in Table 6 is an example of such erroneous extraction.

Neural models seem to suffer the most due to the conjuncts. For instance, M^{2}OIE loses more than half of the F1 score points (from 0.34 down to 0.16) when at least one conjunct is found in the sentence. The exception for the neural systems is OpenIE 6, which is more stable (goes down from 0.29 to 0.23). The reason is because OpenIE 6 was specifically trained to handle conjunctions. Interestingly, ClausIE and MinIE—rule-based systems—lose approximately the same amount of F1 score points as the neural OpenIE 6. This indicates that neu-

Extraction ID	Extractions			BenchIE
Sentence s_1**:** *"A large gravestone was erected in 1866 , over 100 years after his death."*				
t_1	*("A large gravestone";*	*"was erected";*	*"in 1866 over 100 y. after his death")*	0
t_2	*("A large gravestone";*	*"was erected";*	*"in 1866")*	1
Sentence s_2**:** *"The brightest star in Serpens, Alpha Serpentis, or Unukalhai, is a red giant of spectral type K2III located approximately away which marks the snake's heart ."*				
t_3	*("The brightest star in Serpens, Alpha Serpentis , or Unukalhai"*	*"is"*	*"a red giant of sp. type K2III loc. app. away which marks the snake 's heart")*	0
t_4	*("brightest star in Serpens";*	*"is";*	*"red giant")*	1
Sentence s_3**:** *"Lugo and Lozano were released in 1993 and continue to reside in Venezuela.*				
t_5	*("Lugo and Lozano";*	*"released";*	*"in 1993")*	0
t_6	*("Lugo";*	*"were released";*	*"in 1993")*	1
t_7	*("Lozano";*	*"were released";*	*"in 1993")*	1

Table 6: Example extractions along with their score on BenchIE.

ral models can be trained to handle conjunctions similarly as rule-based systems, though there is still room for improvement. We observed similar behaviors for coordinated conjunctions.

C.3 Case Markers

In preliminary qualitative experiments, we found that the objects are often overly specific because they include phrases that should in principle not be part of the expressed concept. Such excessively specific phrases are usually prepositional phrases or case markers. Consider, for example, the triple t_1 in Table 6. The object in this triple is overly specific and, thus, incorrect.

To quantify the effect of such case markers, we bucketized the data according to the number of the typed dependencies `case` that are found in the input sentences. We observed that, as the number of `case` dependencies increases, the performance of OIE systems decreases (Figure 7c). We observed similar behavior for the number of prepositions in a sentence. The rule-based system ClausIE is very sensitive w.r.t. this property, while MinIE is more stable. MinIE was built on top of ClausIE and also focused on restructuring the output of ClausIE, which is the likely reason why MinIE is more robust w.r.t. the case markers. Neural systems (ROIE, OpenIE 6 and M^{2}OIE) are very sensitive to this property, since their performance is much lower when we compare the buckets of 0 or 1 case dependency and the buckets with more than 4 case dependencies.

D More Detailed Discussion on Related Work

D.1 OIE Benchmarks

The currently existing benchmarks are based on token-based scoring. The first attempt to create an OIE benchmark was OIE2016 (Stanovsky and Dagan, 2016). The authors used a dataset from another task—QA-SRL (He et al., 2015)—and automatically ported it to OIE. For scoring an OIE triple, they follow the original task's guidelines (He et al., 2015) and match only the grammatical heads of each slot from the OIE triple with the ones from the golden datasets. Such approach has many drawbacks (Zhan and Zhao, 2020), because (1) every error in the automatic porting transfers over to the evaluation dataset; (2) triples are incorrectly (and over-optimistically) scored because it only considers token-overlaps on grammatical heads, not the whole slots. Being crowdsourced, CaRB (Bhardwaj et al., 2019) improves over OIE2016 by aggregating per-slot token-level precision and recall scores between system and gold extractions across the three slots (subject, predicate, and object). However, such approach is overly-lenient, as it allows for incorrect extractions to be scored positively (see examples in Table 1). Subsequent work followed similar evaluation procedures. For instance, Dong et al. (2021) propose a dataset that evaluates document-level OIE which uses the same scoring procedures as CaRB.

D.2 Multi-faceted Evaluation

While having a reliable single-metric benchmark is crucial for the progress of NLP, recent research indicated that focusing on single metrics is some-

what limited, because it does not provide further insights that go beyond the averaged scores (Ethayarajh and Jurafsky, 2020; Narayan et al., 2021). In particular, Ethayarajh and Jurafsky (2020) argue that single-metric scores ignore certain properties of the evaluated NLP models. Such properties, however, could be relevant for practitioners or for certain downstream tasks. As a consequence, the final evaluation score is computed at the expense of other properties of the model. To allow such multi-faceted evaluations, Liu et al. (2021) proposed ExplainaBoard, which scores NLP systems from several tasks across different facets, and Väth et al. (2021) propose a multi-faceted benchmark for visual question answering.

Due to the incompleteness of current OIE benchmarks—and because of the peculiarity of the task—no such multi-faceted evaluation for OIE has been proposed. For each tested extraction, the state-of-the-art benchmarks provide scores that are in the interval of $[0, 1]$. Such design is employed because the benchmarks are incomplete, which, in turn, makes it difficult to do proper multi-faceted evaluation. To tackle this issue, we propose a multi-faceted evaluation that scores OIE systems across several facets that are important for downstream tasks (see details in Section 3).

D.3 Automatic Error Analysis

Producing automatic error analysis with current benchmarks is not trivial because they are not exhaustive and do not provide crisp scores. For instance, when there are scores within the interval of $[0, 1]$ for each slot—as in CaRB—, it is hard to say where exactly the error occurred. Previous work on OIE performed error analysis manually (Fader et al., 2011; Schneider et al., 2017), which is very time-consuming and inefficient. In contrast to prior work, BenchIE is exhaustive benchmark that provides crisp scores, which allows for automatic per-slot error analysis. We discuss BenchIE's automatic error analysis approach in Section 5.