BenchIE^{FL}, A Manually Re-Annotated Fact-Based Open Information Extraction Benchmark

Anonymous ACL submission

Abstract

Open Information Extraction (OIE) is a field of natural language processing that aims to present textual information in a format that allows it to be organized, analyzed and reflected upon. Numerous OIE systems are developed, claiming ever-increasing performance. In order to establish these performances and compare them with each other, it is necessary to use a reference. *BenchIE* is the latest reference we know of. Despite being very well thought, we noticed a number of issues we believe are limitative. Therefore, we propose *BenchIE*^{FL}, a new OIE benchmark which fully enforces principles of *BenchIE* while containing less errors, omissions and shortcomings when candidate facts are matched towards reference ones. *BenchIE*^{FL} allows to draw some interesting conclusions on the actual performance of extractors.

1 Introduction

Yates et al. (2007) formally introduced the Open Information Extraction task of extracting organized tuples containing information expressed in a sentence, identifying relations and their arguments. This task has numerous downstream applications ranging from Question Answering (Fader et al., 2014) to Text Comprehension (Stanovsky et al., 2015). The first method used to evaluate these systems was to examine their output and manually determine whether the extracted tuples were expressed in a given sentence. This method lacks the capacity to measure the recall of systems, or the proportion of the total sentence information that is captured by the extractors. This led to the creation of different OIE benchmarks, consisting of annotations of all possible tuples from a corpus and a matching function establishing the concordance between extracted and annotated facts. Our analysis of the most recent benchmark *BenchIE* (Gashteovski et al., 2022) shows that although very well thought, its results are noisy and prone to biases, making its conclusion less trustworthy.

Contributions Our main contribution in this work is the release of a new OIE benchmark, *BenchIE*^{FL}, that we created by re-annotating *BenchIE*, correcting frequent errors, inconsistencies and methodology limitations, resulting in more concise, precise and pertinent annotations. *BenchIE*^{FL} benefits as well of a new matching function that is more flexible, and which captures more valid extractions, thus producing — as we shall see — a fairer ranking of evaluated systems.

In doing so, we produce a number of useful resources,[1] including new guidelines for the OIE task, both for sentence annotation and tuple matching. This is necessary because of the lack of consensus on these subjects. We compare seven OIE extractors systems — neural and non neural — and show that believed state-of-the-art systems are not necessarily the best.

This paper is organized as follows. In section 2, we review some existing OIE systems and benchmarks. We introduce issues with *BenchIE* in section 3, leading us to define guidelines discussed in section 4. In section 5 we characterize *BenchIE*^{FL}, a new version of *BenchIE* we annotated by following our guidelines. Experiments are reported in section 6 and we conclude in section 7.

2 Related Work

2.1 OIE Systems

Open information extraction systems vary in their approaches to the task. Earlier systems are mostly rule-based. These systems, like *ReVerb* (Fader et al., 2011), *ClausIE* (Del Corro and Gemulla, 2013) or *MinIE* (Gashteovski et al., 2017) make

[1] We distribute 5 annotated references, annotation guidelines as well as the evaluation scripts we used for our various experiments (Will be distributed upon acceptation)

use of parts-of-speech tags, syntactical analysis and other grammatical characteristics to derive simple extraction rules. Newer systems are almost exclusively using neural approaches, and belonging to one of two distinct categories, either sequence-to-sequence or sequence tagging techniques. Sequence-to-sequence models, like *ImojIE* (Kolluru et al., 2020b) and *M2OIE* (Ro et al., 2020) output the original sentence one word at a time, as well as markers that delimit the arguments and relations, making use of copying and attention mechanisms to output the original sentence words. Neural tagging models like *OpenIE6* (Kolluru et al., 2020a) and *CompactIE* (Fatahi Bayat et al., 2022) use different approaches but mainly focus on identifying relations initially, and then tagging other words as arguments related to identified relations. Models from both categories are typically trained on bootstrapped data from previously released extractors selected with confidence scores and heuristics.

Further information on those extractors, including implementation details are provided in Appendix A.

2.2 OIE Benchmarks

The manual inspection of OIE systems output lacked the capacity to measure any kind of recall. This led to the creation of the first complete OIE benchmark, *OIE2016* (Stanovsky and Dagan, 2016), created by automatically transforming two Question Answer Driven Semantic Role Labeling (QA-SRL) datasets (He et al., 2015) into an OIE benchmark, comprising 3200 sentences (from the Wall Street Journal and Wikipedia) and 10359 extractions. The matching function used by *OIE2016* matches an extraction and an reference tuple if the grammatical head of both their arguments and their relation are the same.

WiRE57 (Lechelle et al., 2019) is the following OIE reference, identifying limitations both in the conversion involved in *OIE2016* and its matching function. The authors demonstrate that it is straightforward to attack this benchmark with a dummy extractor that delivers the best results. They release a new benchmark made up of 57 expertly annotated sentences from Wikipedia and Reuters into 343 tuples, and use **token-level** scoring, meaning that for each annotated tuple, the precision is measured as the proportion of extracted words present in the annotations, and the recall, the proportion of annotated words present in the extractions. This reference is purposely criticized for its small size of just 57 sentences.

The authors of *CaRB* (Bhardwaj et al., 2019) proposed crowd-sourcing as a solution to the high cost of manually annotating sentences. They release their reference of 2714 tuples from a subset of 1200 sentences from *OIE2016* annotated thanks to Amazon Mechanical Turk. They also introduce modifications to *WiRE57*'s scoring function eliminating the penalties for long extractions and those that combine information from multiple annotated tuples.

To the best of our knowledge, the most recently developed benchmark is *BenchIE* (Gashteovski et al., 2022) which is **fact-centered**, meaning that it does not partially reward systems for extracting some part of a fact, but instead evaluates whether extracted and annotated facts are semantically equivalent. Its authors criticize *CaRB* mainly in the way it scores system extractions; showing that token-level scoring allows for incorrect extractions to be scored highly both in precision and recall. To counter this, they propose to use a conservative **exact matching** function, meaning that extractions that are identical to an annotated tuple will count; allowing to measure the overall precision as the ratio of matched extractions to total extractions, and the recall as the ratio of matched annotations to total annotations. This notion of exact match works because of the *fact synset* principle that they introduce : instead of annotating only one formulation of a given fact, they aim to list all *possible* formulations of the fact in a single *synset* or *cluster*. Thus, if an extraction matches any of the formulations of a *synset*, is it said to match that *cluster*. They manually annotate 300 sentences of the original *OIE2016* dataset, resulting in 1354 *clusters*.

These benchmarks yield different conclusions regarding the best performing extractors. This is because their annotation principles, text corpora and matching functions are all different. Recently, Pei et al. (2023) made recommendations for assisting in deciding the best (neural) extractor for a given downstream task, and which benchmarks' characteristics better correlate with it. While we agree that ultimately, we should test extractors on specific tasks (which we also revisit in Section 6.2), there is a need for sound references that will help appreciating limitations of current extraction technology, hopefully leading to better extractors. In their study, Pei et al. (2023) rejected *BenchIE* in com-

Error type	Annotation
For example , when two such hydrophobic particles come very close , the clathrate-like baskets surrounding them merge	
Missing Fact	(when two such hydrophobic particles – come very close – [the] clathrate-like baskets surrounding them merge)
False Fact	(two [such] hydrophobic particles – come – [very] close)
They held the first Triangle workshop in 1982 for thirty sculptors and painters from the US , the UK and Canada at Pine Plains , New York .	
Irrelevant fact	(They – held [the] [first] Triangle workshop in – 1982)
Double annotation	Cluster 10 (They – held [the] [first] Triangle workshop at – [Pine Plains] New York) Cluster 12 (They – held [the] [first] Triangle workshop at – New York)
Double meaning	Cluster 3 (They – held [the] [first] Triangle workshop – for [thirty] painters from [the] US) (They – held [the] [first] Triangle workshop – for [thirty] painters from Canada)

Table 1: Illustration of error types found in a sample of *BenchIE*. Square brackets indicate words that are optional when matching facts.

paring benchmarks,[2] while our inspections make us believe it is the most well-though benchmark.[3]

3 Issues with *BenchIE*

By inspecting a (random) sample of 50 sentences of *BenchIE* - hereafter referred to as *B50* - we noticed a number of issues we think worth being taken care of for more meaningful comparisons of extractors.

3.1 Annotation problems

We identified five error types that are illustrated in Table 1.

Missing fact A fact is missing from the reference. This may be due to the lack of inference in *BenchIE* (see section 4.1), or from oversight or omission. The example shows the main information piece from the sentences and no formulation of that fact is present in the annotated clusters. This fact is not a case of inference, it is simply missing from the annotations.

False fact A fact in the reference is false, or not necessarily implied by the sentence. The example shows a fact that is false because it lacks the mention that *hydrophobic particles* can *come very close* (when *baskets surrounding them merge*), instead conveying the fact that they do.

Irrelevant fact A fact in the reference lacks relevance, because of missing context or other information. In the example, without the optional *the first*, the fact is not relevant because the year of the event is associated with it being *the first*.

Double annotation Two clusters have at least one formulation that expresses the same information. The example shows two clusters being exactly the same without the optional words. The optional words in cluster 10 should be *New York* and not *Pine Plains*.

Double meaning A single cluster has at least two formulations that express different information. The example shows two formulations of the cluster 3 conveying very distinct information.

Table 2 shows the number of sentences in *B50* for which a cluster or a formulation shows a given error type. We observe that more than half the sentences have one or more missing facts. In many cases, this is because of inference (see section 4.1), but some facts have simply been omitted. Both irrelevant or false facts are present in about a third of the sentences.These high frequency of issues in *BenchIE* show the need for a more thorough annotation of the original sentences. These new annotations need to be motivated and conducted by solid guidelines that are typically lacking in the OIE task.

3.2 Matching problem

Alongside the errors found in *BenchIE*'s annotations, we annotated facts output by *ReVerb*, *ImojIE*,

[2] On the grounding that it is a fact-centered benchmark, which we see as a positive characteristic.

[3] Our review of existing benchmarks is not exhaustive. In particular, *ReOIE2016* (Zhan and Zhao, 2019) and *LSOIE* (Solawetz and Larson, 2021) have been proposed as updated versions of *OIE2016*, but they both use the same matching function, which has been largely criticized.

OpenIE6, and *CompactIE* and found many cases were a system made an extraction that in our eye was valid and should match an annotated fact that was not matched by the exact match used in the *BenchIE* benchmark. This is because while Gashteovski et al. (2022) argue that they listed all possible valid formulations of a given fact in each cluster, we find that this is not the case (and we argue that this is in practice very hard to do).

In the 50 sentences of *B50*, we found 26 (52%) sentences with at least one fact from one of the extractors that was not matched when it should have. Because of this, we develop a new matching function aiming to capture more matches between extracted and annotated tuples.

Error type	Count	%
Missing Fact	26	52
False fact	15	30
Irrelevant fact	32	64
Double annotation	28	56
Double meaning	15	30

Table 2: Count of sentences with a given error type in the 50 sentences of *B50* in *BenchIE*.

4 Guidelines

4.1 Annotation Guidelines

Few principles are universally accepted by OIE benchmarks and systems authors. Here, we try to list crucial principles that make most sense for OIE output to be useful for downstream tasks, and aim for those to guide annotation of our and future references. Our full annotation guidelines, along with examples illustrating the principles can be found in Appendix H. Examples for the following principles are shown in Table 3.

Informativity Tuples should be informative and not contain generalities. This is a general principle, meaning that as long as a tuple contains an information expressed in the sentence and that it respects all other principles, it should be included in the annotations. In the negative example, the second argument *living* is not informative, thus that fact should not be annotated.

Minimality Each tuple should be minimal, meaning that it can not be separated into multiple distinct tuples. The faulty annotation in line 2 of Table 3 combines three facts, and should therefore not be annotated.

Exhaustivity The set of tuples for a given sentence should cover all pieces of information expressed in the sentence. At the cluster level, all of the possible formulations for which any of the arguments or the relation is different should be listed. In the example, all the types of *writing* that *He* has done should be annotated in separate clusters.

Relation Completness Relations are responsible for the information; as such they should be complete, meaning that the information in the arguments do not change the core meaning of the relation. In the example, the negative fact should not be annotated because its relation, *is*, is not complete, its meaning is modified by the word *over* in the second argument.

Inference We define inferred tuples as facts that are implied by the sentence (true if the sentence is true), but for which the relation linking the arguments is not present in the text. Inference should be carried out because it is useful in downstream tasks such as QA or knowledge base (KB) population (Gashteovski et al., 2020). However, limits should be set in regards to the information inferred. These nuances are explained in detail in the complete guidelines but simply put, tuples that can be inferred without needing complex reasoning or external knowledge should be annotated. In the last example of Table 3, the tuple is included because it is clearly implied by the sentence.

4.2 Matching Guidelines

To guide which extracted tuples should match with which annotated clusters, we need matching guidelines. Exactly identical extraction and annotation should obviously match but other nuances exist and should be made clear. The following principles aim to illustrate these nuances and examples are shown in Table 5. Our full matching guidelines can be found in Appendix I.

Relation specificity The extracted and annotated relations should be as specific, meaning that prepositions and linking words should not be arbitrarily placed in the arguments or in the relation. In the example, the relation *was thrown* is not as precise and complete as the annotated relations and its meaning is changed by the words *out of* in the second argument, thus the extraction should not be matched.

Word choice Certain extractions may contain syntax errors, misplaced words or other word choices that are different to those of the annotations.

Principle	Sentence	Annotations
Informativity	*Alex lived in Paris and is now living in Cologne.*	✓ (Alex – is now living in – Cologne) ✗ (Alex – is now – living)
Minimality	*The group was created in 2020 by three people.*	✓ (The group – was – created) ✓ (The group – was created in – 2020) ✓ (The group – was created by – three people) ✗ (The group – was created – in 2020 by three people)
Exhaustivity	*He has written several newspaper and magazine opinion pieces.*	✓ (He – has written – [several] pieces) ✓ (He – has written – [several] opinion pieces) ✓ (He – has written – [several] newspaper pieces) ✓ (He – has written – [several] magazine pieces)
Relation completeness	*Tokyo's population is over 13 millions.*	✓ (Tokyo's population – is over – 13 millions) ✗ (Tokyo's population – **is** – over 13 millions)
Inference	*'My classical way' was released in 2010 on Marc's own label, Frazzy Frog Music.*	✓ (Marc's [own] label – is – Frazzy Frog Music)

Table 3: Illustration of annotation principles. Facts in green (preceded by a check mark) should be included in the annotation, while facts in red (preceded by an cross mark) should not.

These are not inherently bad but if they affect the sense of the extraction, then the extraction should not match. The example shows the case were even if the extraction and the annotation do not use the same determiner (*an* and *the*), they both convey the same meaning, i.e. that *He* is older than his brother.

Level of detail Many system extractions carry information from more than one annotated tuple. We want to match these extractions only if the extraction combines information from no more than two clusters, otherwise, we consider it too noisy for a downstream task. The example shows extractions that combine information from two and from three clusters respectively. The negative example is noisier since its third argument is too long and lacks preciseness.

5 *BenchIE*^{FL}

Following our new annotation guidelines, the first author of this paper annotated *BenchIE*'s original sentences resulting in *BenchIE*^{FL}, a re-annotated corpus of 300 sentences. This annotation effort was made using the *AnnIE* annotation platform (Friedrich et al., 2022).

Different statistics of both benchmarks are reported in Table 4. First, we annotate more facts in total, that is, a higher average number of clusters per sentence. This is because we include inferred information and follow the minimality principle (meaning that we divide the information as much as possible). Second, we annotate (far) fewer formulations per cluster, both because relationship specificity is of great importance in our guidelines, and because we don't rely on an exact match function (see Section 5.2), so there is no need for *BenchIE*^{FL} to list all possible formulations of the same fact. Third, we note that on average, our annotations are shorter; again due to the minimality principle. On the other hand, the mean lengths of the relations are more or less equivalent, due to our desire to preserve the specificity of the relations. Appendix G shows an example highlighting differences in both annotations of the same sentence.

	BenchIE^{FL}	***BenchIE***
Total clusters	1798	1354
Avg cluster/sentence	6	4.5
Avg formulation/cluster	3	6
Avg formulation length	10.6	12.5
Avg relation length	3.9	4.0

Table 4: Annotation statistics of *BenchIE* and our re-annotated version: *BenchIE*^{FL}.

5.1 Manual evaluation

Since all of our annotations are carried out by a single annotator, we set out to validate them using another non-author annotator. We present the annotator with sentences of *B50* and their annotations from both sets of annotations (25 each), and then ask them to (blindly) annotate : **exhaustivity** by indicating if a set of annotations for a given sentence fail to covers all the facts expressed, **minimality** by indicating clusters that can be separated into several (smaller) clusters, and **relation completeness** by marking relations that are modified by their arguments or that do not hold on their own.

The results of the validation experiment are presented in Table 6. We observe that for minimality and relation completeness, the annotator found our

Principle	Sentence	Annotations
Relation specificity	*The party was thrown out of the government.*	(The party – **was** – thrown out of [the] government) (The party – **was thrown out of** – [the] government)
		✗ (The party – **was thrown** – out of [the] government)
Word choice	*He is the older brother of Alex.*	(He – is – **[an]** older brother)
		✓ (He – is – **the** older brother)
Level of detail	*Alex broadcasts a web series Music on a website.*	(Alex – broadcasts, [a] web series) (Alex – broadcasts, Music) (Alex – broadcasts Music on, a website)
		✓ (Alex – broadcasts – Music on a website) ✗ (Alex – broadcasts – a web series Music on a website)

Table 5: Examples of matching principles. Annotated facts are in black, facts in green (preceded by a check mark) should be matched to the annotations, while facts in red (preceded by a cross mark) should not.

annotations to follow the guidelines much more closely. For exhaustivity, to make a fair comparaison between both sets of annotations regarding inference, we removed tuples from *BenchIE*^{FL} that could not be worded using words from the sentence. Of the 9 sentences out of which the annotator found missing facts, 5 of them were inferred facts initially present in our annotation that we had removed. Considering this, we find that our annotation to have less than half the missing facts found in *BenchIE*. See Appendix C for more analysis.

	BenchIE^{FL}	*BenchIE*
Non-exhaustive sentences	9 (36%)	9 (36%)
Non-minimal sentences	3 (12%)	16 (64%)
Incomplete relation sentences	4 (16%)	14 (56%)

Table 6: Validation statistics. Non-minimal sentences are those with at least one non-minimal annotation, and incomplete sentences are sentences with at least one cluster whose relation is incomplete.

5.2 Matching function

We set out to develop a new matching function that captures more matches than the exact matching used in *BenchIE*. In order to evaluate this matching function, we annotate the extractions produced by the seven systems introduced in Section 2.1 for the 50 sentences of *B50*, and indicate for each extraction the index of the cluster it should match to according to our matching guidelines; an extraction that should not match is marked by 0. Examples of such annotations are provided in Table 7. The resulting resource, named *BenchIE*$^{FL}_{match}$ contains 9400 extraction-annotation pairs, 96.8% of which have no associated cluster.

We did similarly with the sentences of *Wire57* (Lechelle et al., 2019) in order to get an outdomain set of matches to eventually train a matching function on, leading to *WiRE57*$^{FL}_{match}$, a collection of 10973 extraction-annotation pairs (97.5% non-match). Note that this required to first transform the annotations of *Wire57* into the *BenchIE* format (since it is not annotated with clusters of facts) and insuring that the resulting annotations satisfy our annotation guidelines. Since *Wire57* is a benchmark which gained in popularity lately, we also share *WiRE57*^{FL}, the resulting resource.

While annotating *WiRE57*$^{FL}_{match}$, we gathered three simple types of heuristics that we found useful for accommodating extractions that merge information from several clusters.

Alternative formulations (AF) In two specific (yet frequent) situations, do we credit an extraction which does not match a reference cluster because it regroups information in two clusters: when its relation is reducible[4] to `is`, and when one of its argument contains a coordinate conjunction `and`. Implementation details are provided in Appendix B, but as an illustration, the tuple (He – is a – Canadian musician) might yield to two alternatives (He – is – Canadian) and (He – is a – musician), that will more likely (exact) match the reference.

Level of detail Matching (LoD) We match an extraction which linearization[5] is verbatim the one of a reference formulation and which one of its argument and its relation are also present in another cluster. A typical example is illustrated in Table 5 where the candidate tuple (Alex – broadcasts – Music on a website) is matched because it has the same linearization as the reference tuple (Alex – broadcasts Music on – a website) and *Alex*, and *broadcast* are found in the second cluster: (Alex – broadcast – Music) in the corresponding slots.

[4] We say r reduces to r', if removing optional words from r leads to r'.

[5] By linearization of a tuple, we mean the string obtained by concatenating in that order arg1, relation and arg2.

System extraction	Annotation cluster	Match
(He – served as first Prime Minister of – Australia)	(He – served as – [the][a] Prime Minister)	0
(He – became – founding justice of High Court of Australia)	(He – became – [a] founding justice)	1

Table 7: Examples of matching annotations. Index of clusters are not relevant here.

Punctuation (Punc) We carry out matching removing all punctuation characters and lowercasing strings. This is because we have not listed all possible combinations of capitalization and punctuation in our annotations, and consider we should not.

Matching function	W_m^{FL}	B_m^{FL}
Exact Match (EM)	0.85	0.83
EM + AF	0.87	0.87
EM + LoD	0.86	0.87
EM + AF + LoD	0.92	0.96
EM + AF + LoD + Punc	**0.93**	**0.97**
Perceptron	0.97	0.86
Random forest	0.90	0.85

Table 8: F1 of various matching functions. W_m^{FL} and B_m^{FL} stand for *WiRE57*$_{match}^{FL}$ and *BenchIE*$_{match}^{FL}$ respectively.

The F1 score of various matching functions on both *WiRE57*$_{match}^{FL}$ and *BenchIE*$_{match}^{FL}$ is reported in Table 8. We observe that the exact match function is outperformed by the heuristics we described, and that using them all leads to the best performance overall on both sets. We also train classifiers on *WiRE57*$_{match}^{FL}$, both multi-layer perceptrons and random forests using LightGBM on different sets of features, ranging from matching word counts and similarity metrics to specific word usage and token-level matching scores, with a total of over 50 features created. While the perceptron performs very well on the data set it has been trained on, it does not generalize much better than the random forest classifier on *BenchIE*$_{match}^{FL}$, and both classifiers significantly underperform the best heuristic-based matching functions. Therefore we selected (EM + AF + LoD + Punc) as our default matching function in the evaluation toolkit accompanying *BenchIE*^{FL}. We further verify in Appendix D, that the ranking of systems thanks to our scoring function correlates better than other scoring functions to the ranking obtained by evaluating systems based on the (manual) matching annotations we conducted.

6 Experiments

We use the default configuration for the 7 extractors mentioned in Section 2.1 without attempting to optimize them for specific benchmarks. Pei et al. (2023) observed that optimizing such systems do not lead to significative performance differences.

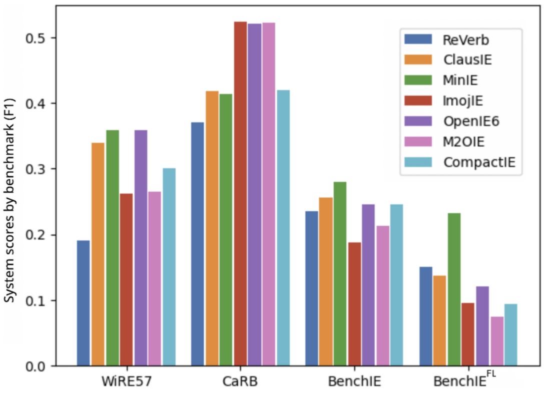

Figure 1: System performance by benchmarks, using the score provided by scoring function of each benchmark.

6.1 Benchmark comparison

We can compare benchmarks by the different system rankings they lead to. Figure 1 shows performances of tested systems on the four aforementioned benchmarks, using the evaluation toolkit coming along with each benchmark.

We observe that *WiRE57*, *BenchIE* and *BenchIE*^{FL} all have fairly similar rankings while *CaRB* identifies very different top performers (*IMojIE* and *M2OIE*). Certain similarities can be observed between the results obtained by *BenchIE* and our reference, both of which rank *MinIE* as the best-performing system. On the other hand, for virtually all other systems, there are (major) differences between their final rankings according to these two references. While no fine-tuning has been performed on these systems and we simply used their default configurations, it is still remarkable that both recent state-of-the art benchmarks, *BenchIE* and *CaRB* largely overestimate neural models compared to *BenchIE*^{FL}. In fact, we observe that neural networks by the virtue of the datasets they have been trained on, have a tendency

passage:	the internal frame backpack is a recent innovation , invented in 1967 by greg lowe, who went on to found lowe alpine and lowepro, companies specializing in backpacks and other forms of carrying bags for various equipment
question:	when was the backpack invented ?
WebAssertions	(a recent innovation – invented – in 1967 – by greg lowe)
WebAssertions^{FL}	([the] [internal frame] backpack – [was]/[be]/[is] invented in – 1967)

Table 9: Question-Passage-Answer triples from *WebAssertions* and *WebAssertions*^{FL}.

of copying large chunks of input texts,[6] that often lead to non informative tuples. See Appendix E.

6.2 ABQA Experiment

Ultimately, OIE extractions, and thus systems, are only useful in downstream applications. This means that a useful benchmark's rankings should follow closely the rankings obtained by evaluating OIE extractions usefulness in these downstream tasks. To test whether this is the case for our reference, we study a task that directly uses OIE: Assertion-Based Question Answering (Yan et al., 2018). In ABQA, the input is a passage of a few sentences and a question, and the output is the answer to that question identified in the passage. Yan et al. (2018) created an ABQA reference, *WebAssertions* by using *ClausIE* to extract all facts from the passages and then asking annotators to identify extractions answering the given question.

Table 9 shows an example of a triple from *WebAssertions*. We found many tuples in it that are too long or even fail to answer the question. Thus, we re-annotated the 100 first passage-question pairs of *WebAssertions*, following the *BenchIE* format and listing all clusters that can answer the question. We distribute the resulting resource named *WebAssertions*^{FL}, an example of which being in Table 9.

We run all tested systems on the passages of *WebAssertions*^{FL}, using our matching function to match extractions of systems to annotations, and use those matches to compute scores for the ABQA benchmark, giving systems one point for a given sentence if one of their extractions match any reference cluster. The scores of tested systems are reported in Table 10, along with their rank. Here again, *MinIE* outperform other extractors.

We compute Pearson product-moment correlation coefficients between this ranking and the one obtained using OIE benchmarks and observe (see Table 15 in Appendix F) that *BenchIE*^{FL} has the highest correlation for the scores on this specific downstream task, leading us to hypothesize that *BenchIE*^{FL} is a better indicator of real performances of extractors on downstream tasks.

System	Score	System	Score
ReVerb	0.23 (2)	*IMojIE*	0.17 (4)
ClausIE	0.18 (3)	*OpenIE6*	0.17 (5)
MinIE	**0.27** (**1**)	*M2OIE*	0.16 (7)
CompactIE	0.17 (6)		

Table 10: Scores (and rank) of systems on *WebAssertions*^{FL}.

7 Discussion

We propose 5 new annotated resources: 2 annotated OIE corpora (*BenchIE*^{FL} and *WiRE57*^{FL}) totalizing 357 annotated sentences, 2 sets of matching annotations (*BenchIE*$^{FL}_{match}$ and *WiRE57*$^{FL}_{match}$) for a total of 107 sentences and extractions produced by 7 systems, and one annotated ABQA reference *WebAssertions*^{FL}. We also propose new guidelines for the OIE task, both for the annotation of tuples and the matching of candidate extractions. We also deliver noticeable improvements on the exact matching function, while not compromising the principles behind *BenchIE*.

Thanks to those resources, we conduct a meaningful comparison of of-the-shelve extractors, showing that plain old technology is still competitive. Finally we conduct a study that shows on a downstream task that our benchmark better reflects the performance of OIE systems on this task.

There are a number of avenues worth pursuing along this work. We would like to measure how well *BenchIE*^{FL} correlates to other downstream tasks. Annotating more sentences is also of interest, especially in other languages.

[6] *IMojIE* and *M2OIE* produce tuples with an average number of words of 13.4 and 12.2 respectively, while for instance, *ReVerb* produces tuples of 7.5 words on average.

Limitations

BenchIE^{FL} aims to correct some of the shortcomings of *BenchIE*. However, it still has some limitations.

- While it contains a fair number of sentences to draw conclusions about which OIE systems to use, it lacks the needed size to be useful in training models. Annotation of sentences requires a lot of effort, but adding to our 357 total annotated sentences (*BenchIE*^{FL} + *WiRE57*^{FL}) would likely be useful.
- We believe our annotations to be more rigorous than previous ones. Still, it is likely that some mistakes were made and should be corrected. It would have been great (but out of the scope of this work) to enrol more annotators to measure their agreement while annotating according to our guidelines.
- Even if the matching function we propose is more performant than the exact match function of *BenchIE*, it still lacks some flexibility. We were not able to train a better classifier, but some different features or methods may outperform our custom function.
- Still, *BenchIE*^{FL}, like most of OIE benchmarks, is only available in English.

Ethics Statement

References

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW ’13, page 355–366, New York, NY, USA. Association for Computing Machinery.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’14, page 1156–1165, New York, NY, USA. Association for Computing Machinery.

Farima Fatahi Bayat, Nikita Bhutani, and H. Jagadish. 2022. CompactIE: Compact facts in open information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 900–910, Seattle, United States. Association for Computational Linguistics.

Niklas Friedrich, Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. 2022. AnnIE: An annotation platform for constructing complete open information extraction benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 44–60, Dublin, Ireland. Association for Computational Linguistics.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.

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 First Workshop on Evaluation and Comparison of NLP Systems*, pages 143–154, Online. Association for Computational Linguistics.

Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. 2022. BenchIE: A framework for multi-faceted fact-based open information extraction evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4472–4490, Dublin, Ireland. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

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

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020b. IMoJIE: Iterative memory-based joint open information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online. Association for Computational Linguistics.

William Lechelle, Fabrizio Gotti, and Phillippe Langlais. 2019. WiRe57 : A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15, Florence, Italy. Association for Computational Linguistics.

Kevin Pei, Ishan Jindal, Kevin Chen-Chuan Chang, ChengXiang Zhai, and Yunyao Li. 2023. When to use what: An in-depth comparative empirical analysis of OpenIE systems for downstream applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 929–949, Toronto, Canada. Association for Computational Linguistics.

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multiˆ2OIE: 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.

Jacob Solawetz and Stefan Larson. 2021. LSOIE: A large-scale dataset for supervised open information extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2595–2600, Online. Association for Computational Linguistics.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.

Gabriel Stanovsky, Ido Dagan, and Mausam. 2015. Open IE as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 303–308, Beijing, China. Association for Computational Linguistics.

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.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Junlang Zhan and Hai Zhao. 2019. Span model for open information extraction on accurate corpus.

A Implementation details

We used implementations of various systems and benchmarks in order to test and compare them to each other. Here is a brief description of the systems we used:

ReVerb This system uses regular expressions to extract candidate facts and a simple feature-based classifier to filter duplicates and non-useful tuples.[7]

ClausIE This system uses grammatical analysis to create *clauses*, or minimal sentences and extracts tuples based on those clauses and their *type*. We could not find a working implementation of *ClausIE* so we used a simple script to generate extraction from a web page providing an API to demonstrate the system's capabilities.[8]

MinIE This system is built on top of *ClausIE*, adding more patterns to identify clauses and modifying final extractions to be minimal and have more solid relations. We used a Python wrapper of the official implementation.[9]

ImojIE This system is a sequence to sequence model using a BERT encoder, trained on extractions from *OpenIE4*, *RnnOIE* and *ClausIE*.[10]

OpenIE6 This system is a succesor to *OllIE*, and it is a grid-labelling model built on BERT embeddings with syntaxic and grammatical constraints used at inference time trained on boot-strapped *ImojIE* extractions.[11]

M2OIE This system is one of the only multilingual system, only needing corresponding language BERT models, that works by first extracting relations and then extracting related arguments in a sequence-to-sequence architecture. [12]

CompactIE This systems works by extracting constituents (arguments and relations) and linking them using a neural classifier, and is trained to output *compact*, or minimal extractions by adding constraints on the extraction components.[13]

All systems were ran on an Apple M2 processor. Most systems had dependencies issues out of the box and significant effort was needed to make them work as expected. Since our datasets were quite small, *CaRB* being the largest with 1200 sentences, computation time was not an issue, although all neural systems needed far more time to extract facts than tule-based systems. *ImojIE* was the slowest of systems, needing more than four hours to run on *CaRB*'s test set.

Regarding the benchmarks we have been using:

WiRE57 : This benchmark is comprised of a small, expertly annotated corpus of 57 sentences and uses a token-level matching function.[14]

CaRB : This benchmark is a crow-sourced annotation of 1200 sentences, and also uses a token-level scoring function. We used the test set which is made up of a total of 640 sentences.[15]

BenchIE : This benchmark uses *clusters* or *synsets* to group all formulations of a single fact, allowing usage of an exact matching function. It is made up of 300 manually annotated sentences. We used the default facet.[16]

B Matching function

B.1 Alternative formulations

Here we explain the details of implementation of the alternative formulations we introduce in section 5.2. The aim of these alternative formulations is to match extractions to annotations even when an exact match does not show correspondence. We introduce the notion of rewriting pairs (A,B) where we authorize an argument of an extraction that contains both A and B to be rewritten by removing either A or B. We called these modified extractions *alternative formulations*

We identify two cases where alternative formulations can be generated safely: in extractions with the relation `is`, and in those where an argument contains the token `and`. In the first case, we collect from the reference all pairs (A,B) from formulations (A,r,B) where r reduces to `is`,[17] while in the second case, we collect pairs (A,B) whenever we have two tuples (E,rel,A) and (E,rel,B) in the reference. Table 11 shows an example of how these

[7] http://reverb.cs.washington.edu/

[8] https://gate.d5.mpi-inf.mpg.de/ClausIEGate/ClausIEGate/

[9] https://github.com/mmxgn/miniepy

[10] https://github.com/dair-iitd/imojie

[11] https://github.com/dair-iitd/openie6

[12] https://github.com/youngbin-ro/Multi2OIE

[13] https://github.com/FarimaFatahi/CompactIE

[14] https://github.com/rali-udem/WiRe57

[15] https://github.com/dair-iitd/CaRB

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

[17] We say r reduces to r', if removing optional words from r leads to r'.

pairs are generated from a sample annotation and extraction and how they can match annotations.

B.2 Scoring extractions

If a candidate extraction does not exact match a reference cluster, we consider alternative extractions obtained by rewriting pairs and give credit to the original extraction if one of those alternatives exact match the reference. We only consider the first matching alternative and its associated cluster for the computation of precision and recall as to not overly reward extractions for combining information and thus not respecting the minimality principle.

Chilly Gonzales is a Canadian musician who lived in Paris, France and in Cologne, Germany.
Annotations: (Chilly Gonzales – lived in – Paris) (Chilly Gonzales – lived in – Cologne) (Chilly Gonzales – is – Canadian) (Chilly Gonzales – is [a] – musician)
Rewriting pairs: `is` : (Canadian – Musician) `and` : (Paris – Cologne)
Extraction: (Chilly Gonzales – is a – Canadian musician) **Alternative formulations:** (Chilly Gonzales – is – Canadian) (Chilly Gonzales – is – musician)
Extraction: (Chilly Gonzales – lived in – Paris and Cologne) **Alternative formulations:** (Chilly Gonzales – lived in – Paris) (Chilly Gonzales – lived in – Cologne)

Table 11: Alternative formulation generation example

C About inferred clusters

Since our annotation and *BenchIE*'s differ largely in handling of inference, we present a modified version of our annotation set for which the original annotator transformed the inferred clusters to use only words present in the original sentence (light inference) and removed the inferred clusters that could not be modified as such (heavy inference). This modified annotation is what is presented to the annotator when comparing ours and *BenchIE*'s annotations in section 5.1, in order to give a fair comparison in regards to exhaustiveness given that *BenchIE*'s annotation does not contain inference, other than in some very rare cases. Statistics regarding this modification and the use of inference in both references are presented in Table 12. We observe that in our original annotation, we had 22 annotated clusters that were instances of heavy inference which we were not able to transform into light inference and had to be removed, while 22 of them could be transformed. We also observe that *BenchIE* has very few instances of inference. This is because they decide to only annotate facts for which the relation is verbatim in the text, although we did find a few instances of light inference.

	BenchIE^{FL}	**Modified**	***BenchIE***
Total clusters	139	117	119
Inferred clusters	47 (34%)	25 (21%)	3 (2.5%)
Heavy inference	44 (32%)	0	0
Light inference	3 (2%)	25 (21%)	3 (2.5%)

Table 12: Inference statistics

D Comparison of scoring functions

Using our manual matching annotations, we compute theoretical scores and rankings for tested systems. We then compute scores and rankings using the different matching scores and functions of previous benchmarks. We compare these matching methods by computing their Pearson product-moment correlation coefficients with the manual rankings on *BenchIE*$^{FL}_{match}$ (See Table 13). We observe that token-level matching has very low correlation and that while both *BenchIE*'s exact match and our *Custom Match* have similar scores, ours correlates to a greater degree.

Matching function	Correlation with manual rankings
WiRE57	0.219
BenchIE	0.961
Custom Match	**0.997**

Table 13: Pearson product-moment correlation between rankings obtained by matching functions and manual rankings on *BenchIE*$^{FL}_{match}$

E Neural system scores

Here we try to demonstrate why neural approaches get lower scores on our benchmark than on others. First we compare lengths of extractions by systems in Table 14, were we see that neural approaches all have longer mean extraction length compared to rule-based systems, except for *CompactIE* wich has been specifically trained to output compact extractions. We also see that our two best performing systems, *MinIE* and *ReVerb* have the shortest extractions. We hypothesize that they obtain the highest scores in part because they are most capable at precisely separating facts into minimal extractions, which is something that neural systems fail to do. This is illustrated in Table 16 were we show all extractions for all systems on a given sentence. We see that both *ImojIE* and *M2OIE*, neural systems, make extremely long and useless extractions, recopying almost the whole sentence. We observed that neural systems have a tendency to recopy large parts of the input sentences without being able to accurately separate facts. We hypothesize that this is because they have been trained on non-minimal and relation-complete extractions.

System	**Mean extraction length** (# words)
ReVerb	7.5
ClausIE	10.9
MinIE	7.9
ImojIE	13.4
OpenIE6	12.2
M2OIE	12.2
CompactIE	9.3

Table 14: Extraction length by system

F Downstream task comparison

This section is a complement to the ABQA experiment in section 6.2 where we evaluated the performance on OIE systems on a downstream task. We computed these ABQA rankings and can then compute Pearson product-moment correlation coefficients between them and the rankings obtained on different benchmarks, using their default evaluation tool-kits. Table 15 shows these correlations. We observe that the ABQA rankings correlate very strongly with the rankings obtained on *BenchIE*^{FL} compared to how it correlates with the rankings of other benchmarks. We can say that our reference is a better indicator of OIE systems performance on the ABQA task and can hope that this is the case for a majority of OIE's downstream tasks.

Benchmark	**Correlation with ABQA Scores**
WiRE57	0.044
CaRB	-0.649
BenchIE	0.616
BenchIE^{FL}	**0.940**

Table 15: Correlation between system scores on benchmarks and on *WebAssertions*^{FL}.

G Annotation Examples

Table 17 shows differences between our annotation and *BenchIE*'s original annotations for the same sentences. We see that our annotation has a lot more clusters, partly because of our inclusion of inference but also because of the minimality principle that guides us to annotate the fact (My Classical Way – was – released), which is not annotated in *BenchIE*. We also see that *BenchIE*'s annotation contain a lot of different formulations of the same fact, most of which we do not consider to be valid because of relation integrity.

His son , John Crozier , Jr. , was an early aviation pioneer who began building a human-powered flying machine in the 1890s , but was killed in a feud in Grainger County before he could complete it .	
System	**Extractions**
ReVerb	(John Crozier – was – an early aviation pioneer) (an early aviation pioneer – began building – a human-powered flying machine) (the 1890s – was killed in – a feud) (he – could complete – it)
ClausIE	(His – has – son) (His son – is – John Crozier) (John Crozier – is – Jr.) (His son – was – an early aviation pioneer) (an early aviation pioneer – began – building a human-powered flying machine in the 1890s) (an early aviation pioneer – began – building a human-powered flying machine) (an early aviation pioneer – was killed – in a feud in Grainger County before he could complete it) (an early aviation pioneer – was killed – in a feud in Grainger County) (he – could complete – it)
MinIE	(His – has – son) (son – is – John Crozier) (John Crozier – is – son) (John Crozier – is – Jr.) (son – was – early aviation pioneer) (early aviation pioneer – began building human-powered flying machine in – the 1890s) (early aviation pioneer – began – building human-powered flying machine) (early aviation pioneer – was killed in feud in – Grainger County) (he – complete – it)
ImojIE	**(His son – was – an early aviation pioneer who began building a human-powered flying machine in the 1890s , but was killed in a feud in Grainger County)** (he – could complete – it)
OpenIE6	(His son – was – an early aviation pioneer who began building a human-powered flying machine in the 1890s) (an early aviation pioneer – began – building a human-powered flying machine in the 1890s) (an early aviation pioneer – was killed – in a feud in Grainger County before he could complete it) (he – could complete – it) (an early aviation pioneer – began building – a human-powered flying machine in the 1890s)
M2OIE	**(His son , John Crozier , Jr. – was – an early aviation pioneer who began building a human-powered flying machine in the 1890s , but was killed in a feud in Grainger County)** (an early aviation pioneer – began building – a human-powered flying machine in the 1890s) (an early aviation pioneer – was killed – in a feud in Grainger County) (he – could complete – it)
CompactIE	(John Crozier – was – an early aviation pioneer) (an early aviation – was killed – in a feud in Grainger County) (an early aviation – could complete – it)

Table 16: Extraction examples

“ My Classical Way ” was released on 21 September 2010 on Marc ’s own label , Frazzy Frog Music .	
BenchIE’s annotatios	***BenchIE*^{FL}’s annotations**
Cluster 1: [‘ ‘] My Classical Way [’ ’] – was released on – 21 September 2010 [‘ ‘] My Classical Way [’ ’] – was – released on 21 September 2010 [‘ ‘] My Classical Way [’ ’] – was released – on 21 September 2010 [“] My Classical Way [”] – was released on – 21 September 2010 [“] My Classical Way [”] – was – released on 21 September 2010 [“] My Classical Way [”] – was released – on 21 September 2010 “My Classical Way” – was released on – 21 September 2010 “My Classical Way” – was – released on 21 September 2010 “My Classical Way” – was released – on 21 September 2010 Cluster 2: [‘ ‘] My Classical Way [’ ’] – was released on – Marc ’s [own] label [‘ ‘] My Classical Way [’ ’] – was – released on Marc ’s [own] label [‘ ‘] My Classical Way [’ ’] – was released – on Marc ’s [own] label [‘ ‘] My Classical Way [’ ’] – was released on – Frazzy Frog Music [‘ ‘] My Classical Way [’ ’] – was – released on Frazzy Frog Music [‘ ‘] My Classical Way [’ ’] – was released – on Frazzy Frog Music [“] My Classical Way [”] – was released on – Marc ’s [own] label [“] My Classical Way [”] – was – released on Marc ’s [own] label [“] My Classical Way [”] – was released – on Marc ’s [own] label [“] My Classical Way [”] – was released on – Frazzy Frog Music [“] My Classical Way [”] – was – released on Frazzy Frog Music [“] My Classical Way [”] – was released – on Frazzy Frog Music “My Classical Way” – was released on – Marc ’s [own] label “My Classical Way” – was – released on Marc ’s [own] label “My Classical Way” – was released – on Marc ’s [own] label “My Classical Way” – was released on – Frazzy Frog Music “My Classical Way” – was – released on Frazzy Frog Music “My Classical Way” – was released – on Frazzy Frog Music	Cluster 1: [‘] [‘] My Classical Way [”] – was – released Cluster 2: [‘] [‘] My Classical Way [”] – was released on – 21 September 2010 Cluster 3: [‘] [‘] My Classical Way [”] – was released on – Frazzy Frog Music Cluster 4: Frazzy Frog Music – is – Marc [’s] own label Frazzy Frog Music – is own label of – Marc Frazzy Frog Music – is owned by – Marc Marc [’s] own label – is – Frazzy Frog Music Cluster 5: Frazzy Frog Music – is [a] – label Frazzy Frog Music – is – [a] label Cluster 6: Marc – has [a] – label Marc – has – [a] label Marc – has – [own] label Marc – owns [a] – label Marc – owns – [a] label

Table 17: Annotaions examples from *BenchIE* and *BenchIE*^{FL}

H Annotation Guidelines

This appendix contains annotation guidelines for the open information extraction task and has been used in the annotation process for the *BenchIE*^{FL} reference. The various principles dictate which facts should and should not be annotated. The information is presented in the following format: Sentences are in the top cell and in the cell bellow, examples that should be annotated are in green and preceded by a check mark, ones that should not be included are in red and preceded by a cross mark.

H.1 Number of arguments

All tuples must contain between 1 and 2 arguments. Extractions with more than two arguments can be split into more compact extractions. The first example shows this principle, while the second example shows how some tuples only have a single argument (we write XXX in the second argument for convenience).

H.2 Informativeness

Annotated tuples must contain relevant information that is expressed in the sentence. Tuples must be informative and relevant. They must not contain generality or empty words that convey no information. In the example, the fact that *he has written* is not relevant since it is a generality, most people *have written* and it is not the information presented in the sentence.

Kyle left for school on Monday.
✗ (Kyle – left – for school – on Monday)
✓ (Kyle – left on – Monday)
✓ (Kyle – left for – school)

Number of arguments : First example

Gideon Rodan taught at the University of Connecticut School of Dental Medicine.
✓ (Gideon Rodan – taught – XXX)
✓ (Gideon Rodan – was – a teacher)

Number of arguments : Second example

He has written several newspaper and magazine opinion pieces in The Guardian, Vice, Billboard, and others.
✓ (He – has written – opinion pieces)
✗ (He – has – written)

Informativeness

H.3 Minimality

Annotated tuples must contain minimal information, which cannot be subdivided into smaller pieces of information. No argument should contain information about two different entities if this is true for both, and no tuple should contain more than one piece of information about an entity if these can be divided. In the first example, all the different minimal pieces of information (creators, time of creation) must be separated in minimal clusters and not grouped like in the example that should not be included. It is sometimes necessary to separate information, if and only if it is also true when separated. In the second example, the dog is neither *black* nor *brown*, but *black* **and** *brown*, whereas in the first example, *He has Cornish ancestors* **and** *He has Welsh ancestors*.

H.4 Exhaustivity

All the minimal information present in the sentence must be included in the annotations. Some argu-

The group was created in 2020 by three people
✓ (The group – was – created)
✓ (The group – was created in – 2020)
✓ (The group – was created by – three people)
✗ (The group – was – created in 2020 by three people)

Minimality : First example

He has Cornish as well as Welsh ancestry.
✓ (He – has – Cornish ancestry)
✓ (He – has – Welsh ancestry)

Minimality : Second example

He has written several newspaper and magazine opinion pieces in The Guardian, Vice, Billboard, and others.
✓ (He – has written – several newspaper opinion pieces)
✓ (He – has written – several magazine opinion pieces)
✓ (He – has written – several opinion pieces)

Exhaustivity

Tokyo's population is over 13 Millions
✓ (Tokyo's population – **is over** – 13 Millions)
✗ (Tokyo's population – **is** – over 13 Millions)

Relation completeness : First example

ments or relations may be affected by modifications but remain true without them, so it's necessary to list all possible formulations that respect the other principles. In the example, it is true that *he wrote opinion pieces, newspaper opinion pieces* and *magazine opinion pieces*, so all these facts must be listed in three separate clusters.

H.5 Relation completeness

Relations are the vehicles of information; arguments must not contain information that changes their meaning. Relations can be complicated but necessary, while they can sometimes be simplified. They must be simplified as much as possible to respect the principle of minimality, without losing their original meaning, expressed in the sentence. In the example, the second argument of the erroneous annotation contains the word *over*, which modifies the meaning of the relation *is*, whereas in the positive example, the second argument, *13 Millions*, is only the object of the relation.

Sometimes, relationships can be complicated but necessary, while sometimes they can be simplified, keeping the additional part optional only if it's made necessary by the lack of other tuples explaining that additional part, as in the second example where the part *from Hungary* in the second cluster is optional because the place of origin of their escape is present in the first cluster. In the third example, *in Paris* is not optional because without this information, the relation no longer holds.

The dog is black and brown.
✓ (The dog – is – black and brown)

Minimality : Third example

His parents are Ashkenazi Jews who had to flee from Hungary during World War II.
✓ (His parents – had to flee from – Hungary)
✓ (His parents – had to flee **[from Hungary]** during – World War II)

Relation completeness : Second example

H.6 Coreference resolution

No coreference resolution is performed outside sentences. Even if a given sentence comes from a document that allows us to resolve a coreference, as OIE is intended to be a task performed on isolated sentences, we only resolve the coreferences of entities included in sentences taken in isolation. Tuples using pronouns for which we can't identify the substitution element may seem meaningless, but coreference resolution must take place outside OIE, being a task in itself. In the example, we don't do coreference resolution for the pronoun *He*, as no information about it is available in the sentence. However, we include a formulation replacing *them* with *tax reductions* in the annotation.

Chilly Gonzales is a Grammy-winning Canadian musician who resided in Paris, France for several years, and now lives in Cologne, Germany.

✓ (Chilly Gonzales – resided **in Paris** for – several years)

Relation completeness : Third example

He did not go as far as he could have in tax reductions ; indeed he combined them with increases in indirect taxes .

✓ (He – combined **them** with – increases in indirect taxes)
(He – combined **tax reductions** with – increases in indirect taxes)

Coreference resolution

H.7 Inference

Inference is necessary: facts directly implied by the sentence, even if not expressed verbatim, are relevant pieces of information. A nuance is necessary here with regard to potential implicit facts. These are not necessarily implied by the sentence and should therefore be omitted. In the first example, it is necessarily implied by the sentence that *Paul Johanson* is *Monsanto's Director of Science*. On the other hand, the fact that *Monsanto's spray* is *gentle on the female organ* is not necessarily true, what is true that this information is said by *Paul Johanson*.

However , Paul Johanson , Monsanto 's director of plant sciences , said the company 's chemical spray overcomes these problems and is gentle onthe female organ .

✓ (Paul Johanson – is – Monsanto's director of science)

✓ (Paul Johanson – **says** – the company's chemical spray is gentle on the female organ)

✗ (the company's chemical spray – is gentle on – the female organ)

Inference : First example

It is then necessary to distinguish between light and heavy inference. We define light inference as a form of inference that does not require logical reflection with respect to the sentence to deduce the fact, which is simply true as long as the sentence is also true. In the second example, the relation is implicit, but the annotated fact is obviously true. Heavy inference, on the other hand, requires some reflection or combination of logical operations to imply the fact. A case of heavy inference is that which requires external knowledge, as in the third example, where knowledge of human culture and the principle of heredity is necessary to make the inference. Another example of heavy inference is generalization, as in the fourth example, where a stronger fact is implied, a generalization of what is expressed in the sentence using a single example. A final example of heavy inference is that of lower or upper limits. As in the fifth example, we don't want to generalize lower or upper bounds to entities that are not directly expressed in the sentence. We therefore include in the reference facts that can be inferred using light inference, but not those resulting from heavy inference.

Jason Charles Beck, a Jewish Canadian musician, was born in 1972.

✓ (Jason Charles Beck – is – Jewish)

Inference : Second example

Gonzales is the son of Ashkenazi Jews who were forced to flee from Hungary during World War II.

✗ (Gonzales – is – Jewish)

Inference : Third example

Gonzales is a McGill-trained virtuoso pianist.

✗ (McGill – trains – pianists)

Inference : Fourth example

H.8 Reformulation

If a relation or argument is expressed in a complex way in the text, a simpler re-formulation of the same fact is added in the same cluster, even if the relation in the two formulations is not the same and the level of detail may be different. This is a compromise between the goal of OIE of collecting all the factual information expressed in the text and the importance of formulating these facts in simple language, which is relevant but not necessarily OIE's primary goal. The example shows a

The prefecture is part of the world's most populous metropolitan area with upwards of 37.8 million people and the world's largest urban agglomeration economy.
✗ (the world – has **upwards** of – 37.8 million people)

Inference : Fifth example

case where the reformulated relation is different but conveys the same meaning.

Sam managed to convince John
✓ (Sam – **managed to convince** – John) (Sam – **convinced** – John)

Reformulation

H.9 Active and Passive Voice

Clusters are used to group together the active and passive formulations of an extraction in a single fact. If the active formulation is not present in the text, it should still be added in the same cluster if it is simpler than the original formulation present in the sentence. The example shows an originally passive tuple and it's active formulation added in the same cluster.

The apple was eaten by Kyle
✓ (The apple – **was eaten by** – Kyle) (Kyle – **ate** – the apple)

Active and Passive Voice

H.10 Attribution and Speculation

Some information in the text is speculative or attributed to an entity, so this characteristic must be included in the relationship, in the way it is formulated in the sentence. This makes it possible to preserve this information without having to introduce a particular structure. This information must be included in the relation, as it is in no way related to the arguments. The example shows a case where the attribution is added in the relation of the tuple.

The earth is flat, according to an Apple Valley man.
✓ (The earth – is **according to an Apple Valley man** – flat) ✗ (The earth – is – flat)

Attribution and Speculation

H.11 Correction

Occasionally, some tuples may consist of words from the original sentence but contain grammatical errors. In this case, the tuple formed from the original words and the corrected tuple should be included in the same cluster. This ensures that neither the systems making the correction nor those using the original text are penalized. The example shows that *newspaper* without an *s* is a grammatical error, so both the original and the correction should be included.

He has written several newspaper and magazine opinion pieces in The Guardian, Vice, Billboard, and others.
✓ (He – has written in – **newspaper**) (He – has written in – **newspapers**)

Correction

I Matching Guidelines

This appendix contains the matching guidelines for the open information extraction task and has been used in the development of the *BenchIE*^{FL} reference matching function. The various principles dictate which pairs of extractions made by systems and annotations should and should not match. The information is presented in the following format: Sentences are in the top cell and in the cell bellow, the different formulations of the same cluster (of the same fact) are in a paragraph and a line break separates them. Clusters in black represent annotations. Examples in green, preceded by a check mark are examples that match an annotation in the reference, while examples in red, preceded by a cross mark do not.

I.1 Exact match

Two absolutely identical extractions should match.

I.2 Relation specificity

Extractions are allowed very little flexibility in the specificity of the relation: the relation is the vehicle of information, so it's important that it's almost as specific as the reference. That said, a different formulation that is just as specific should be accepted. In the example, *was thrown* is not a relevant relation in the context of this extraction, as *was* or *was thrown out of* would have been (the word *out* in argument 2 changes the meaning of the relation).

The Finns party was thrown out of the government and the new "Blue Reform" group kept its cabinet seat.
(The Finns party – **was** – thrown out of the government) (The Finns party – **was thrown out of** – the government)
✗ (The Finns party – **was thrown** – out of the government)

Relation specificity

I.3 Errors

Some extractions made by systems may present syntax or grammatical errors, when a word is misplaced or unnecessary. If this error changes the meaning of the relation or one of the arguments, the extraction should not be matched. If not, it should match the corresponding annotation. In the example, the word *also* refers to the relation *is*, and does not change the meaning of the relation, whereas the word *and* changes the meaning of the extraction, making it nonsensical.

Known for his albums of classical piano compositions, he is also a producer and songwriter.
(He – is [also] – a songwriter)
✗ (He – is – a songwriter **and**)
✓ (He – is – a songwriter **also**)

Errors

I.4 Word Choice

Some words may be equivalent to those present in the annotations in certain contexts, even if we have chosen not to include them in the reference. If these words are used in the system extractions instead of those used in the reference, we still accept the system extraction. Some word choices may be wrong, but we still accept the extraction if the meaning remains. In the example, the determiner *the* is used instead of *a* in the extraction because it's the word found in the original sentence, but both are equally appropriate, so we accept the extraction.

He is the younger brother of the prolific film composer Christophe Beck.
(He – is – **a** younger brother)
✓ (He – is – **the** younger brother)

Word Choice

I.5 Level of Detail

We want to match extractions which have a level of detail higher than the annotation but that convey the same information. By level of detail we mean that they combine information from two annotated clusters. On the other hand, if an extraction combines information from three or more annotated clusters, we consider it to be too noisy and not precise enough to be useful. The positive example is matched because it conveys the same information as the second annotated cluster, and only adds a single level of detail from the third cluster. The negative example is not matched because it combines information from all three annotated tuples into a long and imprecise second argument.

Alex broadcasts a web series Music on a website.
(Alex – broadcasts – a web series)
(Alex – broadcasts – Music)
(Alex – broadcasts Music on – a website)
✓ (Alex – broadcasts – Music on a website)
✗ (Alex – broadcasts – a web series Music on a website)

Level of detail