

WiRe57 Annotation Guidelines

Table of contents

Abstract	1
General principles	2
Annotation guidelines	3
Exhaustiveness	3
Noun phrases	3
Prepositions in annotations	4
Verb phrases	4
Minimality	4
Possessives	5
Rephrasing	5
Passive and active voice	5
Tuples where there more than two arguments are necessary	6
Tuples calling for a single argument	6
Attribution and speculation	6
Pan-document anaphora resolution (optional, but highly recommended)	7
Verbal tenses	8
The limits of inference	8
Implementation suggestion	10

1 Abstract

This companion document to the WiRe57 Open Information Extraction (OIE) benchmark explains in detail the principles and guidelines used to produce the benchmark. It can also be employed by any researcher willing to annotate sentences with their reference (manual) tuples in order to create their own benchmark.

2 General principles

Briefly put, the goal of the annotation task for which we provide guidelines in this document is to produce reference tuples (extractions) from a text.

While there is always a part of arbitrary in such recommendations, we have sought to propose guidelines bound by the OIE principles that we identify and motivate below.

It is worth noting that these guidelines are a compromise between, on the one hand, the need for exhaustive and unambiguous annotations of tuples in sentences, and, on the other hand, an effort to simplify the annotation process for a human being (see for instance guideline 3.2).

The rest of the document explains the principles and annotation guidelines we used to build WiRe57, along with examples for each rule. For further examples, the reader can also simply consult the WiRe57 annotated benchmark itself.

Informativeness

Obviously, we want tuples that are meaningful elements of information. Like illustrated by [Fader et al. 2011](#), we therefore discard well-formed but meaningless tuples like (Faust, made, a deal) in favor of (Faust, made a deal with, the devil). This explains why we resolve anaphoras in our guidelines. Anaphoric mentions being so ubiquitous and being void of meaning outside of the context of their original sentence, we must do so.

Minimality

Extracted tuples should each be minimal, in the sense of conveying the smallest standalone element of information, although that piece must be completely expressed. This means that some facts must be extracted as n -ary relations. It also motivates our choice of striving to isolate the core elements of overlong noun phrases, in order to avoid being uselessly overspecific (i.e. non-minimal).

Exhaustiveness

We seek to manually extract all meaningful relationships occurring in the text, in other words we want to be as exhaustive as possible. This is to measure absolute recall for a system, a notoriously tricky evaluation metric for Open IE.

(Light) Inference

To some extent, the point of producing tuples is to normalize information, so we seek to rephrase the sentence text when it is necessary, which entails light inference. While light inference is helpful to make use of the information extracted, full-fledged inference should be the responsibility of a dedicated program, and is neither part of the Open IE task nor of our guidelines.

At the same time, we want to limit the amount of inference and reasoning necessary to produce these reference tuples. Obviously, “**limited inference**” is ambiguous, but we hope that the examples we provide here will help clarify this admittedly arbitrary choice on our part.

3 Annotation guidelines

In the following, we represent tuples as parenthesized elements, following the pattern $(arg_1, rel, arg_2, arg_3, \dots)$ where arg_i is the argument at position i and rel is the relationship expressed explicitly or implicitly in the source text between the arguments. The number of arguments can be one or more. For instance, `(Kyle, read, the book)` or `(Kyle, fell)`.

Typically, the tokens found in the tuples (e.g. `Kyle`) are found verbatim in the original text. However, this is not always the case. When one or more tokens in a tuple are not directly extracted from the underlying text, they are enclosed in square brackets, like in the example below.

<i>Jewish musician Bob Dylan</i>	<code>(Bob Dylan, [is], Jewish)</code> <code>(Boy Dylan, [is], [a] musician)</code>
----------------------------------	--

Our examples will always take this form: The source text on the left and the tuples that illustrate the guideline being discussed on the right. Sometimes, an example tuple list is not exhaustive, i.e. some other tuples could have been derived from the text, but were not shown in order to alleviate the presentation.

3.1 Exhaustiveness

The tuples should reflect all meaningful relationships found in the source text. Typically, this means that there are multiple tuples for a given sentence, although this is by no means a requirement. It also allows a single word to be involved in more than one tuple.

When in doubt, we recommend that the annotator err on the side of exhaustiveness.

<i>Kyle and Mary read the book, and then left for school.</i>	<code>(Kyle, read, the book)</code> <code>(Mary, read, the book)</code> <code>(Kyle, left for, school)</code> <code>(Mary, left for, school)</code>
---	--

The annotator must also consider relationships and facts that may not be expressed explicitly through a verb, i.e. *non-verbal relations*. Typically, the relation in these tuples will be inferred and written enclosed in square brackets, e.g. `[happened in]` or `[stands for]`.

<i>1923 Kantō earthquake</i>	<code>(Kantō earthquake, [happened in], 1923)</code>
<i>The United Nations (UN)</i>	<code>(UN, [stands for], The United Nations)</code>

3.2 Noun phrases

Often, noun phrases involved in a tuple can be rich in multiple elements of information. For instance, in the sentence `Solo Piano I is an album of classical piano compositions.`, the phrase `an album of classical piano compositions` should yield two tuples when annotating: one using its simplest form `an album` and another using its complete form `an`

album of classical piano compositions. We therefore extract two tuples, one very generic and another, more specific, for the same relation.

<i>Solo Piano I is an album of classical piano compositions.</i>	(Solo Piano I, is, an album)
	(Solo Piano I, is, an album of classical piano compositions)

The complete, longer form of a noun phrase does not include adjectives and other elements of meaning that can be easily separated from the noun phrase to create other tuples. For instance, in the following example, the adjective *great* need not appear in either the simple or extended form of the noun phrase, because it can easily be isolated in another triple (Solo Piano I, [is], great).

<i>Solo Piano I is a great album of classical piano compositions.</i>	(Solo Piano I, is, [an] album)
	(Solo Piano I, is, [an] album of classical piano compositions)
	(Solo Piano I, [is], great)

3.3 Prepositions in annotations

When a relationship verb calls for a preposition, the preposition is placed after the verb in the tuple’s relation slot (Kyle, left for, school), in line with the behavior of Reverb. When more than one preposition is involved (for a *n*-ary tuple), the subsequent ones are placed before their argument : (Kyle, left for, school, on Tuesday).

3.4 Verb phrases

When relationships are expressed with verb phrases containing nouns or other elements (e.g. Tokyo is the capital of Japan.), we prefer annotating the richer relationship (Tokyo, is the capital of, Japan) rather than the more basic (Tokyo, is, the capital of Japan). This allows tuple relations to be more meaningful, and more easily compared, clustered, and aggregated with other relations.

<i>Tokyo is home to Fuji TV.</i>	(Tokyo, is home to, Fuji TV) <i>essential tuple</i>
<i>Tokyo is home to Fuji TV.</i>	(Tokyo, is, home to Fuji TV) <i>facultative tuple</i>
<i>Tokyo’s population is over 13 million.</i>	(Tokyo’s population, is over, 13 million) <i>essential tuple</i>

3.5 Minimality

The tuples should split coordinate elements when appropriate, so as to have as many tuples as there are elements of information. Care should be taken when considering a split, as some coordinated elements should be kept joined (see second and third examples below).

<i>He has Cornish as well as Welsh ancestry.</i>	(He, has, Cornish ancestry)
--	-----------------------------

	(He, has, Welsh ancestry)
<i>This dog is black and brown.</i>	(The dog, is, black and brown)
<i>The printer alternates between portrait and landscape.</i>	(The printer, alternates between, portrait and landscape)
<i>He wears red shoes or black shoes.</i>	(He, wears, red shoes or black shoes)

Some elements may not be explicitly coordinated with a conjunction, yet they should be separate elements of meaningful information, belonging to distinct tuples. Noun phrases are good examples of this.

<i>Flubber is a critically acclaimed cult film.</i>	(Flubber, is, a cult film) (Flubber, is, critically acclaimed)
---	---

3.6 Possessives

Possessives are a special case of inferred relations where the relation is [has].

<i>Mary's dog is brown.</i>	(Mary's dog, is, brown) (Mary, [has], [a] dog)
<i>The prefecture of this city</i>	(this city, [has], [a] prefecture)
<i>The GOP is American.</i> <i>Its leaders include Ronald Reagan.</i>	(The GOP, is, American) (Its/(The GOP's) leaders, include, Ronald Reagan) (The GOP, [has], leaders)

3.7 Rephrasing

Complex verbal phrases are rephrased more simply, yielding two tuples: one using the original phrasing, and another using the simplified one.

<i>Sam managed to convince John.</i>	(Sam, managed to convince, John) (Sam, [convinced], John)
<i>The Kellers had to flee from Germany.</i>	(The Kellers, had to flee from, Germany) (The Kellers, [fled] from, Germany)

3.8 Passive and active voice

When a verb is used at the passive voice, it yields two triples, the original one and another one, rephrased at the active voice.

<i>The apple was eaten by Kyle.</i>	(The apple, was eaten by, Kyle) (Kyle, [ate], the apple)
-------------------------------------	---

3.9 Tuples where there more than two arguments are necessary

For a given relationship, the annotator must try to identify all arguments that are essential or very important to the meaning of the resulting tuple. Typically, this means adding place and

time arguments to a given verb that calls for them. The order of the arguments after the second argument does not reflect their relative importance when conveying the meaning of the tuple.

Note that the preposition `to` is placed in the relation slot while `in` is in `arg3`, in keeping with section 3.3.

<i>In 1869, the 17-year-old Emperor Meiji moved to Edo.</i>	(Meiji, moved to, Edo, in 1869)
<i>He has more apples than her.</i>	(He, has, more apples, than her)
<i>Emperor Meiji moved this seat to Tokyo from the old capital of Kyoto in 1868.</i>	(Emperor Meiji, moved, this seat, to Tokyo, from the old capital of Kyoto, in 1868)

When an argument adds little information or specificity to a tuple, it can be dropped. This happens typically with adverbs of manner.

<i>Bob liked to play piano from time to time.</i>	(Bob, liked to play, piano)
---	-----------------------------

When multiple arguments compete for a single slot in a tuple (through an `and` or an `or`, for instance), they produce as many tuples as there are competing arguments.

<i>Bob played the piano and the sax in the 80s.</i>	(Bob, played, the piano, in the 80s) (Bob, played, the sax, in the 80s)
<i>Bob both hated and loved this theory.</i>	(Bob, hated, this theory) (Bob, loved, this theory)

3.10 Tuples calling for a single argument

Some relationships have a single argument, typically the subject of the verb, which results in tuples of the form (`arg1`, `rel`). In the example below, `grow in popularity` is a non-compositional phrase that cannot be expressed as (`X`, `grow in`, `popularity`) and therefore does not call for any argument.

<i>Sangster grew in popularity.</i>	(Sangster, grew in popularity)
-------------------------------------	--------------------------------

3.11 Attribution and speculation

The truth value of some statements and relationships in a given text are influenced by either attribution (e.g. “according to X”) or speculation (e.g. “If Y holds, then Z.”). The annotation guidelines we propose here attempt to capture this, albeit superficially, by decorating such attributed and speculative tuples with an additional Boolean flag.

This means that the annotation process must be arranged so that the annotator can add the value of such a flag for any given tuple they produce. Typically, this takes the form of an additional field the annotator can tick when they deem the tuple to be attributed/speculative.

<i>An Apple Valley, California man plans on launching himself in a homemade rocket to prepare for obtaining evidence that the Earth is flat, according to The Washington Post.</i>	(the Earth, is, flat) [✓ attributive/speculative]
<i>The Earth is round, according to sane people.</i>	(The Earth, is, round) [✓ attributive/speculative]
<i>If everything goes well, the Sun shines on you.</i>	(The Sun, shines on, you) [✓ attributive/speculative]

When a sentence has the form *X said that Y* or equivalent, the annotator produces the triple (*X*, *said that*, *Y*), regardless of the syntactic or semantic complexity of *Y*.

<i>The idea is that all is pretty much on track for growth that will be stronger than in 2017.</i>	(The idea, is that, all is pretty much on track for growth that will be stronger than in 2017)
--	--

3.12 Pan-document anaphora resolution (optional, but highly recommended)

The authors of these guidelines are convinced that anaphora resolution is an integral part of open information extraction, whether it be carried out by machines or by humans. A large proportion of nouns and noun phrases involved in tuples are not explicitly mentioned, but rather used indirectly through anaphora. For WiRe57, for instance, we found that 57% of tuples fall into this category.

Anaphora resolution allows the tuples to convey immediate meaning, rather than having to be processed (imperfectly) by anaphora resolution pipelines. Consequently, we found it somewhat simplifies the assessment of their quality as the annotator produces them.

The antecedent of a given anaphora can be anywhere in the document, or can be deduced from prior knowledge, *as long as this antecedent is used consistently for all its anaphoric mentions*.

We use the forward slash to separate the mention from its antecedent, like so: *she/Mary*. When there are multiple words in either the mention or the antecedent, parentheses surround the multi-word expressions, like so: *she/(Mary Simpson)* or *(the girl)/(Mary Simpson)*.

<i>John likes Mary.</i>	(John, likes, Mary)
<i>But he shies away from the beautiful girl.</i>	(he/John, shies away from, (the beautiful girl)/Mary)

Possessive adjectives are also resolved by using an antecedent in a possessive form (e.g. *John's*).

<i>John plays football, but his friends do not.</i>	(John, plays, football) (His/(John's) friends, do not, [play] football)
---	--

Temporal anaphora is also resolved when possible..

<i>A government crisis shook the country last summer.</i>	(A government crisis, shook, the country, (last summer)/(in summer 2017))
---	---

3.13 Verbal tenses

The verbs in the tuples should employ the same verbal tense as that found in the original text. When a verb is inferred, however, its tense should be the one the writer would have used, had they had to make explicit the relation. In the example below, note the past for *took place in* and the present for *[has]*.

<i>The meeting took place in June, a day after Virtanen's co-ruling Finns party had elected anti-immigration hardliners as its new leaders.</i>	(The meeting, took place in, June) ([the] Finns party, [has], new leaders)
---	---

3.14 The limits of inference

Because the concept of "light inference" is subjective, we propose a few examples and counterexamples that delineate the limits between the two classes.

<i>Jason Charles Beck, a Jewish Canadian musician, was born in 1972.</i>	(Jason Charles Beck, [is], Jewish)
<i>Gonzales is the son of Ashkenazi Jews who were forced to flee from Hungary during World War II.</i>	(Gonzales, [is], Jewish) Complex inference based on culture and human heredity

<i>Gonzales is a McGill-trained virtuoso pianist.</i>	(McGill, trains, pianists) Don't infer generic truths (stronger plural statements) from isolated examples.
---	---

<i>Nomura is amongst the more bullish.</i>	(Nomura, [is], bullish) Light inference, easily accessible
<i>The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin.</i>	(Arthur Dempster, Nan Laird, and Donald Rubin, [wrote], (a classic paper)/(The Dempster-Laird-Rubin paper), [in] 1977)

	Unambiguous relation between the authors and their papers ([wrote]), but this tuple is borderline.
--	--

<i>But as in the fairy tale, let's go with just three: central banks, trade, and bubbles.</i> <i>The first is the danger is that there will be a policy mistake, squeezing debtors.</i>	(the first, [is], [central banks]) Complex inter-sentences reasoning
--	---

4 Implementation suggestion

One could argue that guidelines are as good as their achievability in a real-life setting. This is why we recommend facilitating the annotation process by using a spreadsheet program like Excel or, better yet, Google Sheets, to create the annotation task and to gather these annotations.

One simply has to provide the annotator with the original sentences and a few columns below each to enter the args, the relations, and additional information like the attributed/speculative aspect of these triples. Gathering the resulting data is straightforward.

The screenshot below shows one possibility using Google Sheets, for WiRe57.

EM 3	These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.				
attrib/spec?	arg1	relation	arg2	arg3	arg4 arg5
	(These parameter-estimates)/(r	are used to determine	the distribution of the latent vari	in the next E step	
	the latent variables	[have]	[a] distribution		
EM 4	The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin.				
attrib/spec?	arg1	relation	arg2	arg3	arg4 arg5
	The EM algorithm	was explained in	(a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin)/(The Dempst		
	The EM algorithm	was explained in	(a paper)/(The Dempster–Laird by Arthur Dempster, Nan Laird, and Donald Rubin		
	The EM algorithm	was explained [in]	1977		
	(a paper)/(The Dempster–Laird explained		The EM algorithm		
	(a classic 1977 paper by Arthur explained		The EM algorithm		
	The EM algorithm	was given its name in	(a classic 1977 paper by Arthur by Arthur Dempster, Nan Laird, and Donald Rubin		
	The EM algorithm	was given its name in	(a paper)/(The Dempster–Laird by Arthur Dempster, Nan Laird, and Donald Rubin		
	The EM algorithm	was given its name [in]	1977		
	(a classic 1977 paper by Arthur [gave] its name [to]		The EM algorithm		
	(a paper)/(The Dempster–Laird [gave] its name [to]		The EM algorithm		
	Arthur Dempster, Nan Laird, an [wrote]		(a classic paper)/(The Dempste [in] 1977		
	The EM algorithm	[has]	[a] name		
EM 5	They pointed out that the method had been "proposed many times in special circumstances" by earlier authors.				
attrib/spec?	arg1	relation	arg2	arg3	arg4 arg5
	They/(Arthur Dempster, Nan Le	pointed out that	(the method)/(The EM algorithm)	had been "proposed many times in special circumstar	
x	(the method)/(The EM algorithm had been proposed by	earlier authors	in special circumstances		
x	earlier authors	proposed	(the method)/(The EM algorithm	in special circumstances	