Information Extraction

Jonathan May

October 25, 2024

1   Origins

(A lot of this comes from [2]. A lot also comes from Heng Ji's (UIUC) IE class slides.)

By 1994 the US government was already familiar with information retrieval: search a corpus of documents and retrieve those documents that match your search terms – now you don't have to read so many documents! But documents can be long; how about we search for just the facts we want instead? Originally, there were templates of all the info the gov't wanted to find (e.g. locations and actions of ships scraped from navy telegraph cables). This was tested in a series of evaluations (this is how evaluations to drive NLP research got going!). Originally, you had to participate in the entire pipeline of various kinds of information retrieval but then (1995) they split into independent and more generic tasks to encourage more participation by smaller teams. One of the tasks that year was 'Named Entity Recognition.' This eventually led to the 'Automatic Content Extraction' program (ACE) which focused on even more fine-grained, independent tasks. The corpora produced in 2005 by ACE are still used, nearly 20 years later.

2   Why

Why do we want to have a big knowledge graph at all? Ultimately, don't we just want to interact via natural language?

Yes, a really good version of, e.g., Siri is an end goal. But even ChatGPT (in 2023) is not all that great at respecting the truth on its own. A fact repository, i.e. a knowledge graph, is fundamental to that kind of bot (we think).
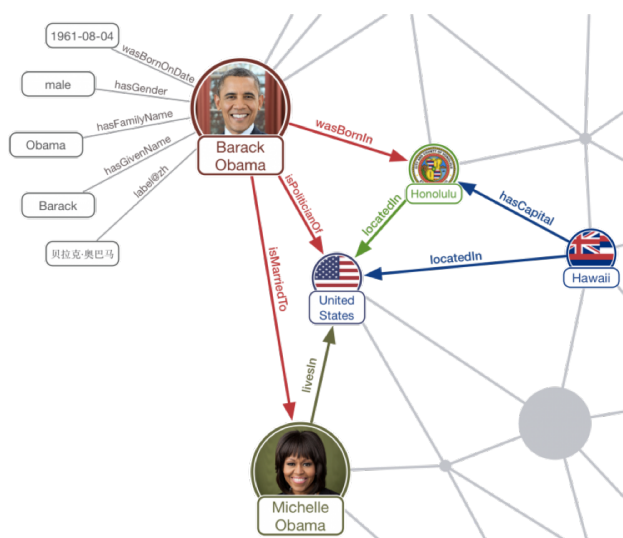
Consider also investigative journalism. In 2020, BuzzFeed got access to a lot of 'FinCEN' financial irregularity disclosures[1] which were a lot of financial transaction data but also narratives by bankers detailing the irregularities. The team actually tried to use IE but couldn't; so they read everything by hand and compiled networks showing who was involved in what[2]. It's not feasible to read in a database of reports and then write a news article; that's not likely in the current visible AI horizon (as of 2024, though possibly an 80% solution is now available). But it is feasible to be able to build an information network.

---

[1] https://www.buzzfeednews.com/fincen-files
[2] https://www.datanami.com/2020/09/25/icij-turns-to-big-data-tech-to-unravel-fincen-files/

3 Tasks

Here I'll mostly outline the individual tasks that we see today that are called IE and try to get into their details a bit. The figures show the types that were used in ACE and are still widely used today, though there are additions/exceptions. The ultimate output of IE can be thought of as a *knowledge graph*, that is, a representation of every conceivable piece of information and how it relates to every other conceivable piece of information, a snippet of which is shown below. With such a graph, information queries would be more straightforward to look up. However, in practice, it is far more common to work on one aspect of constructing this graph at a time.

3.1 Named Entities

Type	Examples
person	Fred Smith; the undertaker
organization	Ford; San Francisco 49ers; a car manufacturer
GPE	France; Los Angeles
location	Nile; Mt. Everest; southern Africa
facility	Disneyland; the Berlin Wall; Aden's streets
vehicle	the U.S.S. Cole; the train; the helicopter
weapon	Anthrax; bullets; tear gas

The general idea here is to find all references to things that have proper names. In general, we'd also like all kinds of 'language external' elements like addresses, times of day, etc. Subsequent evaluations/definitions/systems have added more types.

But we don't always just find the names. Sometimes, we find the nominal references too (i.e., references to named things without using the names) as well as the pronominal references (i.e., references using a pronoun). Often these are evaluated separately; names are easier to detect and type than nominals, which are easier than pronominals.

In other circumstances, we may want to find special-purpose entities. One particular type of interest is chemicals, proteins, enzymes, etc. These are often identified in various kinds of medical literature.

3.1.1 Detection

The detection task is: given a text, find the spans of the entities. What constitutes a 'span' can be trickier to define than you might think. Consider:

```
The Los Angeles Times, a fine newspaper, arrives in my town on Saturdays,
but it is usually late, because Culver City has a traffic problem and
God hates me.
```

Some questions that have to be answered (typically consult your annotation guidelines; I provide my best guess):

- Is 'Los Angeles Times' marked? It's the paper, not the organization, I think... (yes) What about 'newspaper'? Still an org? (yes?) What about 'it'?

- Is the 'The' included? (yes)

- Is 'Los Angeles' the GPE marked? (No)

- Is the 'City' in 'Culver City' marked? (Yes)

- Do we count tokens or characters? (I think tokens)

- If tokens, is the comma marked? Or the period in 'me.'? (no; there is a 'standard' tokenization)

- Is God a person? (No)

What makes matters worse is that despite annotation guidelines, people don't necessarily read them, and might evaluate/tokenize differently and then report results that are not replicable. It can be a big mess! Assuming you can agree on those standards, Micro-F1 on exact span match is typically what is reported.

The best way to learn this data is as a tagging problem, but with tags like `PER, ORG, GPE, LOC, FAC` instead of `NN, JJ, ADV`, etc. And since the elements we are learning are multi-token, we use what is known as *BIO* notation; the beginning of an entity is tagged with `B`, and other tags of that entity are tagged with `I`. Words not in an entity are marked with `O`. Thus:

```
B-ORG I-ORG I-ORG   I-ORG O  B-GPE I-GPE   O  O      O
The    Los   Angeles Times in El    Segundo is great .
```
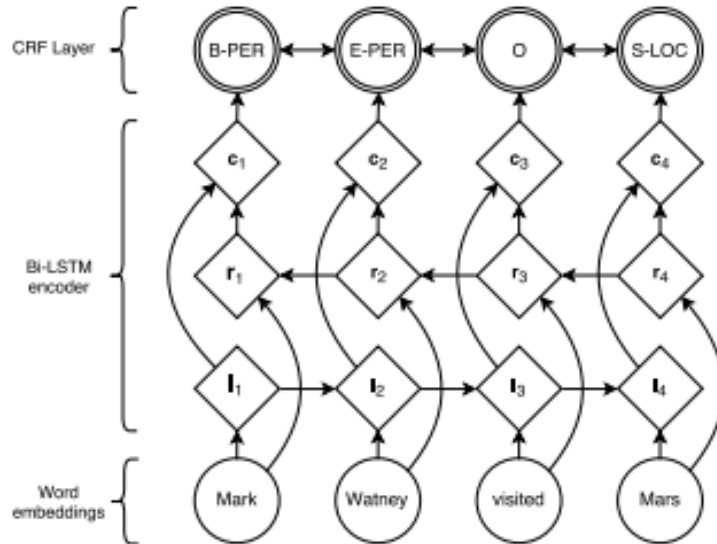
Another variant is *BIOSE* which distinguishes between `Beginning`, `Inside`, `End`, and `Solo` entities. Either can be learned with an HMM or LR/MaxEng tagger, just like a POS tagger. Even better is a CRF which considers the entire *sequence* to be an item that is predicted; for tag sequence $T = t_1, \ldots t_n$ and word sequence $W = w_1, \ldots, w_n$:

$$P_{HMM}(T, W) = \Pi_i P(w_i|t_i)P(t_i|t_{i-1})$$

$$P_{MEMM}(T|W) = \Pi_i \frac{\exp(\theta \cdot f(t_i, t_{i-1}, w_i))}{\sum_t \exp(\theta \cdot f(t, t_{i-1}, w_i))}$$

$$P_{CRF}(T|W) = \frac{\Pi_i \exp(\theta \cdot f(t_i, t_{i-1}, w_i))}{\sum_{T'} \Pi_i \exp(\theta \cdot f(t'_i, t'_{i-1}, w_i))}$$

In the vanilla HMM, we are just using word, tag and tag, tag co-occurrence, but in MEMM and CRF we can define various features. Naturally, those features could themselves be the outputs of a neural network, saving us the trouble of having to come up with features ourselves. A pretty good model [5] (90.94 on CoNLL 2003; SOTA as of 2021 (updated in 2024) is 94.6 [14]) is the bidirectional LSTM; here's a figure from the paper:

They also use *character* embeddings; same idea. Note that these results are not hugely better than CRFs with hand-defined features, but no features needed to be defined. This works well across a variety of languages.

3.2 Coreference

Coreference is the task of clustering entity mentions that all refer to the same entity. This is particularly important when considering nominal and pronominal mentions but also when considering entities that might be named different ways (Los Angeles Times, The Los Angeles Times, The Times, LA Times) or ambiguous people names (President Clinton, Secretary

Clinton, Senator Clinton, Clinton). Here's an example from a recent survey paper [8] that illustrates how difficult this is:

> The Queen Mother asked Queen Elizabeth II to transform her sister,
> Princess Margaret, into a viable princess by summoning a renowned
> speech therapist, Nancy Logue, to treat her speech impediment.

- How many people are described in the above sentence?

- Label all person mentions (name, nominal, pronoun) and indicate coreference.

Features we are using as humans to get coreference right:

- Gender match (Queen, her)

- Number match (The team, The Yankees, they)

- Entity type match (Joe Smith loves New York. The city this father of three grew up in...)

- Metonymy – tricky! (The Yankees returned to their city winners. New York had lost the first two games...)

> But the May 2 clash between separatists and Ukrainian government supporters
> in Odessa that took nearly 50 lives,... That battle was portrayed by
> Kremlin-controlled Russian media as evidence that the Kiev government is
> bent on recovering the occupied areas even if it has to shoot innocent
> bystanders to do so.

Is 'government' referring to the same government?

Pronouns in particular are not easily resolved with surface features; this has led to the 'Winograd challenge' exemplified as follows:

- The city council refused to give the demonstrators a permit because they feared violence.

- The city council refused to give the demonstrators a permit because they advocated violence.

- When Sue went to Nadia's home for dinner, she served sukiyaki au gratin.

- When Sue went to Nadia's home for dinner, she ate sukiyaki au gratin.

I think even the best models are still not using all of the context available to humans (and not even all the context made available). It becomes a computationally explosive task, and in a sense, the search (attention?) through the information is not done in a principled way yet.

3.3 Evaluation

While scores are calculated as F1, evaluating this properly is quite tricky since one has to consider how to deal with 'polluted' clusters, multiple clusters of the same entity, and absent clusters. Traditionally an average of 3 different F1 scores is taken.[3]

There are a variety of techniques for doing entity coreference. The top performer described in [1] and [6] is built as a classifier on top of a bidirectional LSTM (fine-tuned from ELMo); for each mention, a distribution over all possible antecedents is predicted. In general, n^2 comparisons must be made!

Per [1], top Coreference scores on the Ontonotes data set (from 2007) are in the mid .70s to mid .80s for name mentions, depending on the type. And .58 to .77 for nominals and .26 to .51 for pronominals. A new state of the art was reached in 2024 [7] of 83.6 overall F1 (2019 overall around 79 overall). This is still very much an unsolved problem. It is also important to distinguish between coreference given perfect mentions and end-to-end coreference (i.e. after mention identification). See results below.

	PER			ORG			GPE		
	Chains not found	NEC F1	Coref F1	Chains not found	NEC F1	Coref F1	Chains not found	NEC F1	Coref F1
(Raghunathan et al., 2010)	16%	0.55	0.50	34%	0.42	0.41	14%	0.67	0.56
(Clark and Manning, 2015)	36%	0.46	0.56	40%	0.39	0.46	21%	0.61	0.61
(Clark and Manning, 2016a,b)	21%	0.61	0.67	29%	0.50	0.52	17%	0.68	0.65
(Lee et al., 2017)	28%	0.58	0.69	26%	0.58	0.56	12%	0.76	0.68
(Lee et al., 2018)	7.5%	0.80	0.77	15%	0.69	0.61	8%	0.81	0.69

Table 3: Performance of systems. Chains not found and NEC F1 refer to the new named entity focused metrics. Coref F1 refers to the evaluation combining MUC, B^3 and CEAFE, on test data.

	PER			ORG			GPE		
	Name	Pronoun	Nominal	Name	Pronoun	Nominal	Name	Pronoun	Nominal
(Raghunathan et al., 2010)	0.55	0.45	0.23	0.47	0.35	0.11	0.73	0.44	0.19
(Clark and Manning, 2015)	0.50	0.34	0.10	0.46	0.34	0.15	0.65	0.57	0.22
(Clark and Manning, 2016a,b)	0.66	0.49	0.15	0.54	0.47	0.33	0.72	0.59	0.41
(Lee et al., 2017)	0.64	0.41	0.15	0.65	0.48	0.39	0.80	0.70	0.47
(Lee et al., 2018)	0.85	0.58	0.26	0.76	0.64	0.47	0.85	0.77	0.51

Table 4: NEC F1 by type of mention. The errors on names are high, though it is possible to resolve these with NER and string matching or similarity. Pronoun errors are high as expected.

Coref Results

3.3.1 Grounding/Linking

Having formed a cluster of entity mentions, one may also link them to a pre-existing knowledge base and thus *ground* them to some already existing facts. A typical general purpose knowledge base is Wikipedia; this is sometimes then known as *wikification*. Other than Wikipedia, there are special-purpose KBs. One well-known one is 'GeoNames' which is a listing of 11 million placenames. Another is Chemanalyser, which has many chemical compounds, substances, and compound classes. Freebase and YAGO are other all-purpose KBs.

Grounding is not at all trivial; consider *Chicago*, which can refer to a city, a band, a typeface, or one of many professional sports teams. In addition to grounding to the KB, there will generally be some entities that aren't in the KB; these should be properly identified as NIL.

---

[3]See [1] if interested.

Note that it is possible to simply link without clustering first and if done properly all non-NIL links will also implicitly be proper clusters; this can often be easier than simply determining coreference. However, it is still necessary for NIL mentions.

As with coreference linking can be done with perfect mentions or end-to-end; SOTA is 86.6 micro end-to-end [4] and 94.9 with perfect mentions [12]; determining the methodologies is an exercise left to the reader.

3.3.2 Make It Harder

Some more challenging extensions on the basic entity identification task:

- Few-Shot: Given 20 examples of a new type, how well can you recognize it?

- Cross-Lingual (often termed 'zero-shot'): Given training data in one language (typically English), how well can you do in another language where no training data is given?

- Cross-Domain: Turns out the way entities are mentioned and co-referred varies across domains e.g., news text, conversation, web text.

- Fine-Grained: Instead of 6–10 types, how about hundreds of types, e.g. `basketball-player`. Then the same span can have many different labels (think of them as properties).

- Ultrafine-Grained: Extract thousands from large ontology (e.g. YAGO). Predict the location where a mention should be in an ontology graph

3.4 Entity-Entity Relations (Relation Extraction)

Type	Examples
physical	location of a person: Fred was in France
part-whole	the lobby of the hotel; Paris, France
personal-social	his lawyer; his wife; his neighbor
org-affiliation	the CEO of Microsoft; a student at Harvard
agent-artifact	my home; my car
gen-affiliation	a Methodist minister; American troops

Relations are ways entities are related. Typically, relations are between exactly two entities and the order matters. The table above lists relations from ACE (there are subtypes not listed); a SemEval task in 2010 [3] defines other relational types. There are special-purpose relation types too, e.g., in medical literature, we often want to know about relations like 'bonds' or 'phosphorylates.'

3.4.1 Basic Methods

Early methods that can work reasonably well are so-called "Hearst Patterns":

1. Come up with some basic patterns, such as `[PERSON], born in [LOCATION]` for the "Born in" relation (think of some for "agent-artifact", "part-whole.")

2. Extract a lot of entity pairs from a large corpus (e.g. "Einstein, born in Germany")

3. Find these entity pairs elsewhere in the text (e.g. "Einstein, who began life in Germany")

4. Learn new patterns from these spans (`[PERSON], who began life in [LOCATION]`)

5. Repeat!

Usually we want training data to do some form of supervised learning. The above method is a way of obtaining training data 'automatically.' Another general method is "Distant Supervision:"

1. Obtain a partial list of relations (e.g. from Wikipedia)

2. Find sentences that contain the entities in the relation pair and label them

3. (Self-training extension) Build a model with the data, use it to label more data, and build another model (this can get noisy and degrade quickly, though)

Prior NLP analysis, such as Entity mentions, coref, dependency parses, POS tags, and semantic relatedness, are all good features that can be used to model this analysis task.

3.4.2 Outside Information

A major difficulty with relation identification (and IE in general) is that humans intuit information is based on a lot of background or even common-sense knowledge, which is hard to convey to machines. Having access to *background information* makes the task a lot easier for machines. Here is a (made-up) example:

`David Cone was seen at the premiere of the new Star Trek movie last night. The former Royal and Yankee color commentator said science fiction was one of his biggest passions, along with his family and baseball.`

There is a `works-for` relationship between `David Cone` and `Kansas City Royals`. There are many inference steps needed to make this connection. However, it's relatively easy to find the Wikipedia page for Cone and then to find links for the Royals (and the Yankees). This information makes it easier to elicit the relationship.

Some more examples of how it's complicated to determine relations can be found below.

Reasoning Types	%	Examples
Pattern recognition	38.9	[1] *Me Musical Nephews* is a 1942 one-reel animated cartoon directed by Seymour Kneitel and animated by Tom Johnson and George Germanetti. [2] Jack Mercer and Jack Ward wrote the script. ... **Relation:** publication_date **Supporting Evidence: 1**
Logical reasoning	26.6	[1] "Nisei" is the ninth episode of the third season of the American science fiction television series The X-Files. ... [3] It was directed by David Nutter, and written by Chris Carter, Frank Spotnitz and Howard Gordon. ... [8] The show centers on FBI special agents *Fox Mulder* (David Duchovny) and Dana Scully (Gillian Anderson) who work on cases linked to the paranormal, called X-Files. ... **Relation:** creator **Supporting Evidence: 1, 3, 8**
Coreference reasoning	17.6	[1] *Dwight Tillery* is an American politician of the Democratic Party who is active in local politics of Cincinnati, Ohio. ... [3] He also holds a law degree from the University of Michigan Law School. [4] *Tillery* served as mayor of Cincinnati from 1991 to 1993. **Relation:** educated_at **Supporting Evidence: 1, 3**
Common-sense reasoning	16.6	[1] *William Busac* (1020-1076), son of William I, Count of Eu, and his wife Lesceline. ... [4] *William* appealed to King Henry I of France, who gave him in marriage Adelaide, the heiress of the county of Soissons. [5] Adelaide was daughter of Renaud I, Count of Soissons, and Grand Master of the Hotel de France. ... [7] *William* and Adelaide had four children: ... **Relation:** spouse **Supporting Evidence: 4, 7**

- People Magazine has confirmed that actress Julia Roberts has given birth to her third child a boy named **Henry Daniel Moder**. Henry was born Monday in Los Angeles and weighed 8 lbs. Roberts, 39, and husband **Danny Moder**, 38, are already parents to twins Hazel and Phinnaeus who were born in November…

- He [**Pascal Yoadimnadji**] has been evacuated to France on Wednesday after falling ill and slipping into a coma in Chad, Ambassador Moukhtar Wawa Dahab told The Associated Press. His wife, who accompanied Yoadimnadji to **Paris**, will repatriate his body to Chad, the amba. → is he dead? in Paris?

- Until last week, **Palin** was relatively unknown outside **Alaska**… → does she live in Alaska?

- The list says that the state is owed $2,665,305 in personal income taxes by singer **Dionne Warwick** of South Orange, **N.J.**, with the tax lien dating back to 1997. → does she live in NJ?

3.5 Extensions

- Cross-Document relation extraction – What are the set of relations expressed by considering a whole corpus, rather than a single document? Show supporting evidence.

- Cross-Lingual/Cross-Modality – Now the documents are in Chinese, Spanish, and English. There are movies as well as text files.

- Fine-grained relations (as with fine-grained entity types)

- Relation and Entity at the same time – Maybe the decisions shouldn't be pipelined, but a joint decision should be made (can be done as semantic parse, can be done as a multi-task model).

3.6 Open IE

You need not have an ontology of relations, or even entity types for that matter, but could conceivably identify all relations of any kind between all entities in the world. An example from Stanford's webpage on open IE: the sentence "Barack Obama was born in Hawaii" would create a triple (`Barack Obama; was born in; Hawaii`),

In a survey on the matter [9] pointed out that a prime difficulty here is evaluation; data sets vary considerably and have very different properties. The goal of open IE is to scale to very large volumes of text, but this makes preparing reference data difficult. Most evaluation sets remain in Wikipedia and News domains, even though another point was to be able to extract in heterogeneous domains. There is still a lot of research to be done.

3.7 Events

Type	Examples
life	is born; marries; dies
movement	transport; travel
transaction	sell; purchase; acquire
business	found; merge
conflict	attack; demonstrate
contact	meet; phone; write
personell	hired; fired; elected
justice	arrest; trial; convict

Events are specific things that happen, involving participants, causing a change in state. There are various ways to define what constitutes an event, but in ACE, there is typically a *trigger*, i.e. the word in a sentence that connotes the event, and zero or more *arguments*, that is, labeled spans of entities involved in the event. In an example from Eisenstein:

> Elected mayor of Atlanta in 1973, Maynard Jackson was the first African
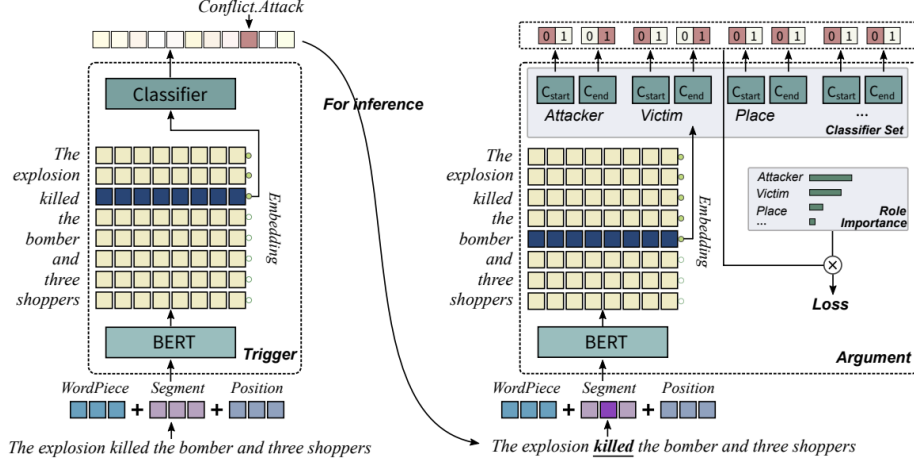> American to serve as mayor of a major southern city.

The event is an *election*, with roles office: mayor, district: Atlanta, date:1973, and person-elected: Maynard Jackson.

Events in most data sets are discovered in the context of a single sentence. Multiple events can appear in one sentence, and the same span can serve as more than one argument in multiple events. For example:

```
John shot Mark for killing his brother Dan.
```

There is a *shot* event with agent: John and patient: Mark, as well as a *killing* event with agent: Mark and patient: Dan.

Various classifiers have been used to first detect and label triggers, then for each trigger detect and label spans. Trigger classification is in the 80s or so, and argument classification in the 50s-60s. Here is a figure and table of results from [15]:

Phase Model	Trigger Identification(%)			Trigger Calssfication(%)			Argument Identification(%)			Argument Calssfication(%)		
	P	R	F	P	R	F	P	R	F	P	R	F
Cross Event		N/A		68.7	68.9	68.8	50.9	49.7	50.3	45.1	44.1	44.6
Cross Entity		N/A		72.9	64.3	68.3	53.4	52.9	53.1	51.6	45.5	48.3
Max Entropy	76.9	65.0	70.4	73.7	62.3	67.5	69.8	47.9	56.8	64.7	44.4	52.7
DMCNN	80.4	67.7	73.5	75.6	63.6	69.1	68.8	51.9	59.1	62.2	46.9	53.5
JRNN	68.5	75.7	71.9	66.0	73.0	69.3	61.4	64.2	62.8	54.2	56.7	55.4
DMCNN-DS	79.7	69.6	74.3	75.7	66.0	70.5	71.4	56.9	63.3	62.8	50.1	55.7
ANN-FN		N/A		79.5	60.7	68.8		N/A			N/A	
ANN-AugATT		N/A		78.0	66.3	71.7		N/A			N/A	
PLMEE(-)	84.8	83.7	**84.2**	81.0	80.4	**80.7**	71.5	59.2	64.7	61.7	53.9	57.5
PLMEE							71.4	60.1	**65.3**	62.3	54.2	**58.0**

Table 2: Performance of all methods. Bold denotes the best result.

Events face a number of challenges. IAA is quite low on them in general. The largest data sets have fewer than 40k events annotated. Multilingual events fare even worse; ACE 2005 has Arabic and Chinese annotation as well but the amount is lower and the quality even worse.

3.7.1 Ontology

There are several. REO (rich event ontology) and ACE (automatic content extraction) are popular. Could also consider the frames of FrameNet and verbs of VerbNet. The types are in the hundreds. Each event type has a set of valid arguments though most have an agent (often referred to as 'arg0') and patient (often referred to as 'arg1').

3.7.2 Evaluation

As with Entities, F-measure is used. However multiple steps are needed to 'find' an event and a portion of these could be considered 'gold', could be ignored, or could be included in the calculation:

- Find the trigger word or span

- Label the trigger

- Find the argument word or span (done multiple times)

- Label the argument (done for each found argument)

Often you will see argument and trigger statistics where the arguments are extracted given *gold* triggers. It's important to read carefully!

3.8 Event-Event Relations

One can discuss relations between events as well. These are typically temporal or causal relations though there are also subpart relations and some others. Temporal seem to have been studied the most; the *Timebank* corpus [11] was constructed and annotated to try to capture temporally related events. An example is below:

```
There was <SIGNAL sid="8"> no </SIGNAL> <EVENT EID="57" CLASS="STATE"
TENSE="PAST" ASPECT="NONE"> hint of trouble </EVENT> <SIGNAL id="11">
in </SIGNAL> the last <EVENT class="OCCURRENCE" aspect="NONE"
eid="10" tense="NONE"> conversation </EVENT> between controllers and
TWA pilot Steven Snyder. But <TIMEX3 TID="58" val="PT1M30S"
type="DURATION" temporalFunction="false"> a minute and a half
</TIMEX3> <SIGNAL SID="59"> later </SIGNAL>, a pilot from a nearby
flight <EVENT aspect="NONE" eid="18" tense="PRESENT"
CLASS="REPORTING"> calls </EVENT> in.

<SLINK eventInstanceID="eiid57" signalID="8" relType="NEGATIVE"/>
<TLINK eventInstanceID="eiid57" relatedToEvent="eiid10"
        signalID="s11" relType="IS_INCLUDED"/>
<TLINK eventInstanceID="eiid18" relatedToEvent="eiid10"
        signalID="s59" relType="AFTER" magnitude="t58"/>
```

How might events be temporally related?

- A started and finished before B

- A and B partially overlap

- A starts before and ends after B

- A and B completely coincide

However, most of the time the start and/or end of an event is unspecified, unknown, or unknowable from text.

Events can also be structured relative to other events, e.g. an `election` is an event that involves, among other subevents, several `debate`s.

Newer work [10] shows a variety of ways of looking at evaluation (there are again numerous data sets so timebank itself is not necessarily being always used); in general determining the order of events, whether they overlap or have containment relationships, etc. is not particularly agreed upon by annotators, and thus it is difficult to build consensus in corpora let alone build satisfactory systems. Causality can be even worse.

3.9   Non-Events

A tricky bit about events is that while we use language to describe things that did or do happen, we also use it to describe events that are not so concrete:

- Events that did not happen ('The Dodgers failed to capture a World Series Title last night') – this example includes an event that didn't happen as well as a negative event (the failure) that did!

- Events that might happen ('If inflation rises too fast the economy could collapse, say experts.')

- Non-specific events that may have happened ('Whenever there is a fire, crew 147 is on the scene to put it out.')

- Hedged events ('These results suggest that the D gene might be involved in granulocyte differentiation.')

In most event recognition work the task is to *avoid* recognizing these events and to only focus on actual events that occurred. To the degree that these not-quite-event cases are pursued, there is the FactBank [13] which annotated 77,000 tokens and 9500 events on 208 documents for how factual the events are (factual, non-factual) and how certain the claim is (certain, probable, possible). They claim IAA of 0.81! I haven't seen a lot of recent work trying to solve this as a task.

3.10   Scripts/Schema Induction

A theory, according to Roger Schank is the idea that all events comprise sequences of more fundamental events. These sequences are known as 'scripts.' The classic example is 'eating at a restaurant':

1. Enter = walk in, look at tables, figure out where to sit, sit.

2. Order = acquire menu, choose food, get waiter attention, provide order, waiter provides order to cook.

3. Eat = cook gives food to waiter, waiter gives food to you, you put food in mouth, chew, and swallow.

4. Exit = waiter gives check, you pay, waiter gives you the receipt, you stand up, you walk out.

A recent DARPA program was concerned with discovering these scripts (also known as schemas). It is quite difficult to find text evidence for these components despite having enormous corpora. Some of the more helpful data sets are instructional corpora such as WikiHow, but even then there are many holes in knowledge.

There is prior empirical work in working with schemas. Chambers and Jurafsky extract *chains* of events from documents where the events have a common argument (a 'main character'). From these chains, then a Cloze test can be proposed, where one event is hidden and the goal is to determine what it is. This could be useful for narrative generation. There has been a little recent follow-up work, but the task still needs more definition and the quality of results indicates we're still at the beginning of this complicated analysis.

References

[1] Oshin Agarwal et al. "Evaluation of named entity coreference". In: *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*. Minneapolis, USA: Association for Computational Linguistics, June 2019, pp. 1–7. DOI: 10.18653/v1/W19-2801. URL: https://www.aclweb.org/anthology/W19-2801.

[2] Ralph Grishman. "Twenty-five years of information extraction". In: *Natural Language Engineering* 25.6 (2019), pp. 677–692. DOI: 10.1017/S1351324919000512.

[3] Iris Hendrickx et al. "SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals". In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 33–38. URL: https://www.aclweb.org/anthology/S10-1006.

[4] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. "End-to-End Neural Entity Linking". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 519–529. DOI: 10.18653/v1/K18-1050. URL: https://www.aclweb.org/anthology/K18-1050.

[5] Guillaume Lample et al. "Neural Architectures for Named Entity Recognition". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: https://www.aclweb.org/anthology/N16-1030.

[6] Kenton Lee, Luheng He, and Luke Zettlemoyer. "Higher-Order Coreference Resolution with Coarse-to-Fine Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 687–692. DOI: 10.18653/v1/N18-2108. URL: https://www.aclweb.org/anthology/N18-2108.

[7]   Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. "Maverick: Efficient and Accurate Coreference Resolution Defying Recent Trends". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13380–13394. DOI: `10.18653/v1/2024.acl-long.722`. URL: `https://aclanthology.org/2024.acl-long.722`.

[8]   Vincent Ng. "Machine Learning for Entity Coreference Resolution: A Retrospective Look at Two Decades of Research". In: *AAAI*. 2017.

[9]   Christina Niklaus et al. "A Survey on Open Information Extraction". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3866–3878. URL: `https://www.aclweb.org/anthology/C18-1326`.

[10]   Qiang Ning et al. "Improving Temporal Relation Extraction with a Globally Acquired Statistical Resource". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 841–851. DOI: `10.18653/v1/N18-1077`. URL: `https://www.aclweb.org/anthology/N18-1077`.

[11]   James Pustejovsky et al. "The TimeBank corpus". In: *Proceedings of Corpus Linguistics* (Jan. 2003).

[12]   Jonathan Raiman and Olivier Raiman. "DeepType: Multilingual Entity Linking by Neural Type System Evolution". In: *CoRR* abs/1802.01021 (2018). arXiv: `1802.01021`. URL: `http://arxiv.org/abs/1802.01021`.

[13]   Roser Saurí and James Pustejovsky. "FactBank: A corpus annotated with event factuality". In: *Language Resources and Evaluation* 43 (Sept. 2009), pp. 227–268. DOI: `10.1007/s10579-009-9089-9`.

[14]   Xinyu Wang et al. "Automated Concatenation of Embeddings for Structured Prediction". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 2643–2660. DOI: `10.18653/v1/2021.acl-long.206`. URL: `https://aclanthology.org/2021.acl-long.206`.

[15]   Sen Yang et al. "Exploring Pre-trained Language Models for Event Extraction and Generation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5284–5294. DOI: `10.18653/v1/P19-1522`. URL: `https://www.aclweb.org/anthology/P19-1522`.