

Information Extraction

Andrea Esuli

Sequence labeling

Many NLP and information extraction tasks are focused on determining some properties of interest inside a piece of text:

- PoS of every word
 - Syntactic role of every word
 - Determining if a word, or a sequence of words, identifies a certain type of information, e.g., the name of a person/location/brand
 - Infer other properties, e.g, the unit of measure of a number, "I am 1.80" vs "I am 42"
 - Link pieces of text that are related, e.g., "*Andrea* is a researcher, *he* is from Pisa"
 - Link a piece of text to element of a knowledge base/ontology
-

Sequence labeling

In these tasks, a document is no more an atomic entity, but it is processed as a sequence of token.

There is not a one-to-one relation between document and output, such as in classification.

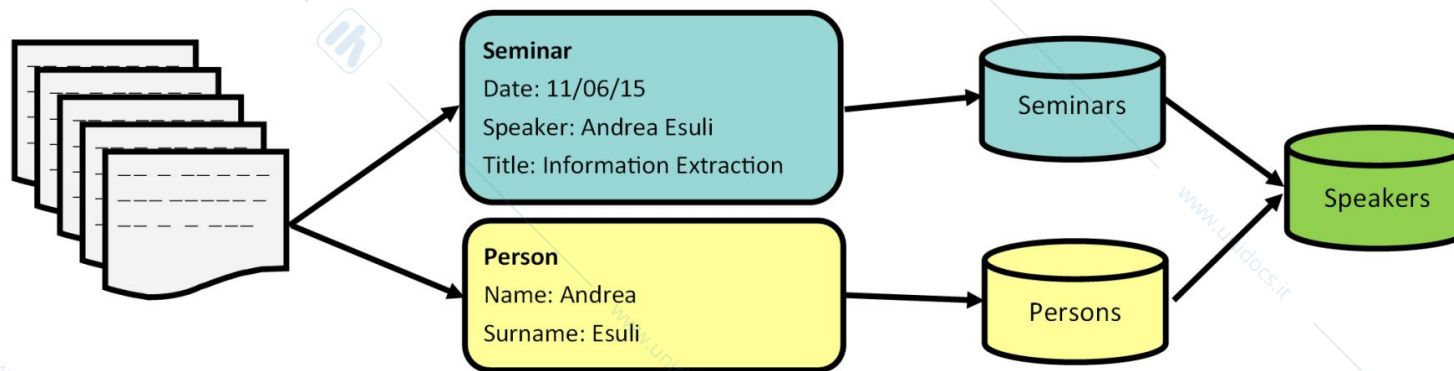
We can extract from text a variable amount of information, depending on its length, but also on its relevant to the specific tasks we apply to it.

Every token is the subject of the analysis and their order in text obviously play a relevant role in the outcome of the analysis.

Information Extraction

Information Extraction (IE) is about extracting structured information from unstructured or partially structured text.

IE is a step toward higher level (semantic) representation of knowledge with respect to classical IR (e.g., web search).



Information Extraction

Two key IE tasks:

- Named entities recognition

Andrea Esuli is a researcher as *ISTI-CNR*

$p_1 = \text{Person}(\textit{Andrea Esuli})$

$o_1 = \text{Organization}(\textit{ISTI-CNR})$

- Relation extraction

Andrea Esuli is a researcher at *ISTI-CNR*

$r_1 = \text{Role}(\textit{researcher}, p_1, o_1)$

Named Entity Recognition

Named Entity Recognition (NER) is the problem of identifying pieces of text that refer to elements belonging to predefined categories such as:

- Persons
Andrea Esuli, Mario Rossi, Rossi, President of USA, President
 - Organizations
Inter, Milan, Roma, Lazio
 - Locations
Milan, Pisa, via Garibaldi, Lazio, Tuscany, Arno, Tirreno
 - Temporal expressions
July 3, Friday, today, last century, the '60, for an hour
 - Quantities
one kilogram, one kilo, 2 tera, a quarter, a dozen
-

Named Entity Recognition

The problem can be split into subproblems:

- entity spotting:

I saw *Andrea Esuli* riding his bike.

- entity classification:

Andrea Esuli → Person

- entity identification (a.k.a. entity linking, wikification):

Andrea Esuli → <http://www.esuli.it> (URI)

The first two steps are usually performed together.

NER using Rules

Lexicons (dictionaries, gazetteers, ontologies) play a relevant role in *entity spotting*.

Rules are usually hand-made, and have the form of patterns and properties that have to match the context of the NE to have a recognition.

Example of extraction rule, adapted from [ANNIE](#)

```
Rule: isFemale({  
    Lookup.class == female_person_first_name,  
    Lookup.ontology == "gate:/creole/ontology/demo.daml"  
}):person
```

NER using ML

Machine learning-based IE usually translates the extraction problem into *a word classification problem*.

Barack Obama flew to Rome last week

[Barack Obama]_{per} flew to [Rome]_{loc} [last week]_{time}

A *binary word classifier* is learned for each type of recognized entity.

- A classifier classifies every word as representing or not an entity.
- Depending if annotation can overlap or not, i.e., a word can have only one label type or more, the output of the classifiers is combined in a *single-label* classification or a *multi-label* one.

NER using ML

Each word is represented by features that capture its morphologic, syntactic, and semantic properties...

$r(\text{Barack}) = [\text{'Barack'}, \text{'barack'}, \text{firstCap}, \text{mixCase}, \text{NNP}, \text{male} \dots]$

...and those of its context, usually defined as a set of preceding and following words.

The vector that represents a word is thus the concatenation of the features that define the observed word and those of the context words (taking into account their relative position):

$$v(w_i) = r(w_i) + r(w_{i-1}) + r(w_{i-2}) + r(w_{i+1}) + r(w_{i+2})$$

NER using ML

Once we assign a proper representation (e.g., probabilistic or vectorial) to every element that is the object of the annotation, traditional machine learning methods seen for text classification can be applied to IE.

Neural networks (recurrent and attention models) have also found successful application to tagging, IE, Entity linking.

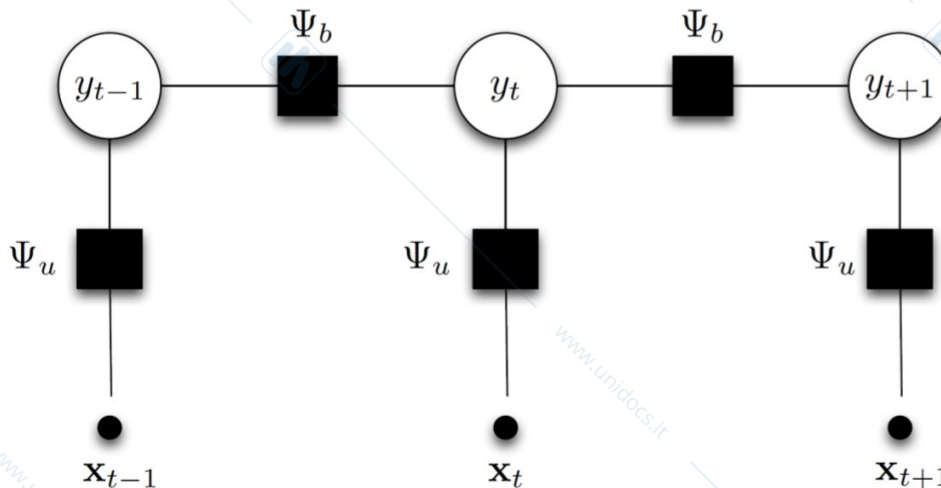
The linear structure of text can further exploited (in addition to features) by using *graphical models*.

A graphical model is a *probabilistic model* that represents the dependencies between the observed objects with a graph.

NER using graphical models

Conditional Random Fields (CRFs) are a kind of **probabilistic graphical models** that explicitly model dependencies among variables of the problem, including those that happen among labels.

A *linear chain* CRF determines the labeling of a piece of text on all the words at the same time, by maximizing the labeling probability of the whole sequence.



Evaluation

The accuracy in annotation of the relevant parts of text is usually measured by finding *matching* annotations between the *true annotations* in the dataset and the *predicted ones*.

The matching criterium can be **strict** (exact match) or **lenient** (starting at the same word, or just overlapping on some parts).

- Exact match does not capture the gravity of errors.
- Lenient match can be tricked.



Evaluation

Evaluating at the word level gives a more graded evaluation, like lenient match, but it cannot be fooled.

A	TN	FP	TP	FN	TP	TN	FN	TN	TN	FP	(B as the gold standard)																																																					
	Lo	re	m	i	p	s	u	m	d	o	l	o	r	s	i	t	a	m	e	t	,	c	o	n	s	e	c	t	e	t	u	e	r	a	d	i	p	i	s	c	i	n	g	e	l	i	t	.	Q	u	i	s	q	u	e	a	c	c	u	m	s	a	n	.
B	TP	FN	TP	FP	TP	TN	FP	TN	TN	FN	(A as the gold standard)																																																					

Classifying also the separators between words gives an evaluation that coincides with exact match on perfect annotation, yet it is graded.

A	TP	TP														
	Q	u	i	s	q	u	e	a	c	c	u	m	s	a	n	.
B	TP	TP														

A	TP	FN	TP																				
	Q	u	i	s	q	u	e	[b	l	a	n	k]	a	c	c	u	m	s	a	n	.
B	TP	FP	TP																				

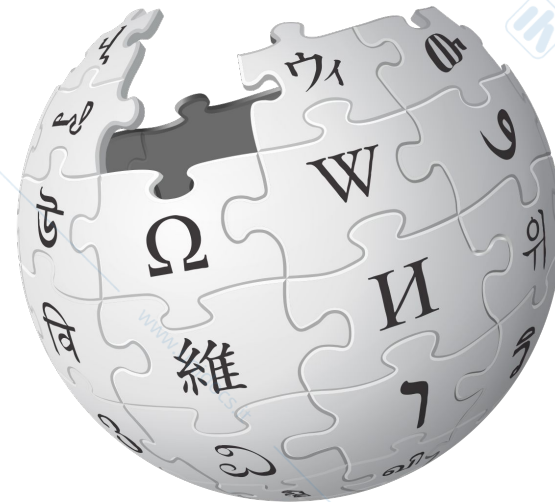
Wikification

Wikification is the task of linking the relevant parts of a piece of text to the relative Wikipedia entities.

- Identification of relevant parts of text is usually made by matching on a list of *surface forms* for entities.
- When more than one entity is assignable to a piece of text (and also to remove spurious matches) the Wikipedia link graph is exploited.

<http://dexter.isti.cnr.it/>

<http://tagme.di.unipi.it/>



Opinion Extraction

Opinion Extraction is focused on establishing *relations* between relevant *entities* of the domain and *subjective expressions* associated to them.

An Opinion Extraction system must be able to perform:

- entity recognition (domain-dependent, possibly including attributes)
battery, screen, signal, GPS, memory...
- subjectivity recognition (polarity can be done in a second time)
short battery life, very large screen, not so strong signal...

Opinion Extraction

Entities can be identified using predefined lists or ontologies.

Once entities are marked, their context is analyzed to find the related subjective expressions.

The *screen* is very nice, but it results in a short battery life.

Parsing trees may be of help to connect entities with evaluations, e.g.:

- Verbal phrases may link entity and subjectivity.
- Noun phrases may contain both entity and subjectivity.

```
(S (S (NP (DT The) (NN screen))
      (VP (VBZ is)
           (ADJP (RB very) (JJ nice))))
  (, ,)
  (CC but)
  (S (NP (PRP it))
      (VP (VBZ results)
           (PP (IN in)
                (NP (DT a) (JJ short) (NN battery) (NN life))))))
```

GATE

GATE is a text processing tool that includes support for human annotation of entities in text.

general architecture
for text engineering

L'esame È stato eseguito con sequenza T2 STIR e con sequenze T1 3D dinamiche prima e dopo somministrazione di mdc paramagnetico, acquisite secondo piani di scansione assiali.	Esiti chirurgici
Diffusi esiti cicatriziali in sede retroareolare sinistra.	BIRADS
Non si apprezzano potenziamenti sospetti bilateralmente.	Enhancement descrizione
In particolare non si È osservato un corrispettivo RM dell'immagine descritta mammograficamente a sinistra.	Enhancement presenza/assenza
In sede ascellare bilateralmente si apprezzano alcune linfadenopatie di verosimile significato reattivo.	Indicazioni Esame
In relazione al quadro RM, si ritiene sufficiente eseguire controllo con esame ecografico tra 6 mesi.	Informazioni Tecniche
	Linfonodi locoregionali
	Protesi descrizione
	Terapie/follow-up

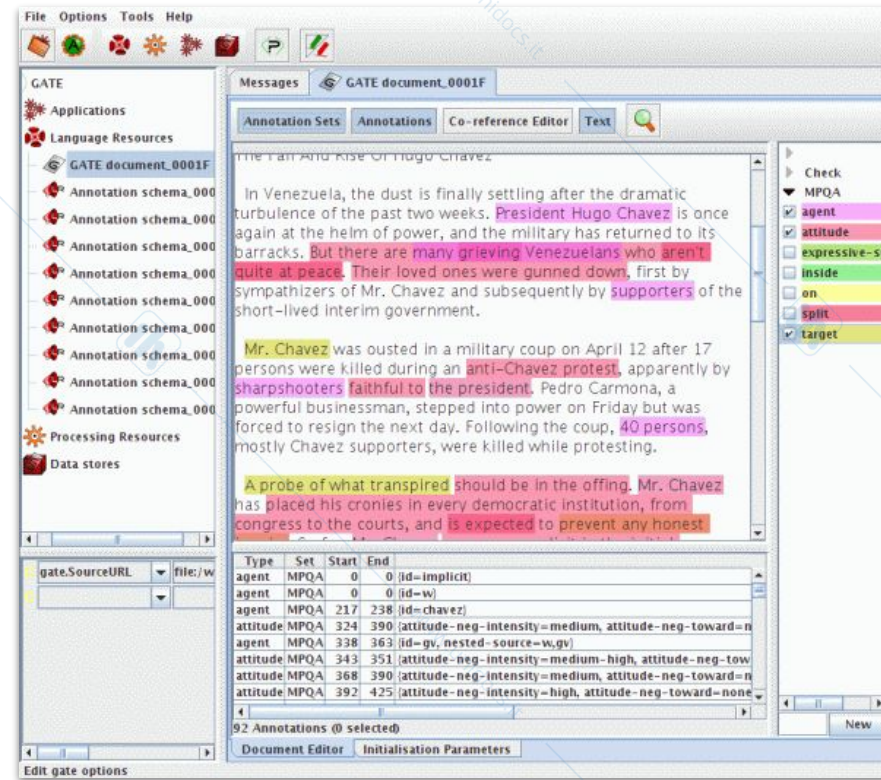
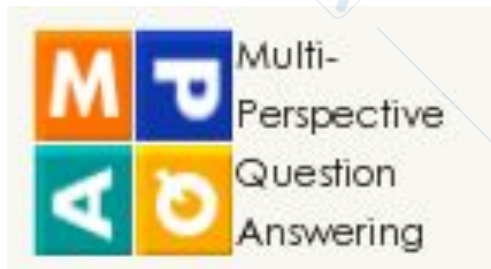
CODICE ACR:06.1

The screenshot displays the GATE software interface. On the left, a sidebar lists 'Applications' and 'Language Resources', including 'GATE document_0001F' and several 'Annotation schema_000' entries. The main window shows a document titled 'GATE document_0001F' with text from 'The Fall And Rise Of Hugo Chavez'. The text is annotated with various entities and relations, such as 'agent', 'attitude', and 'target'. A table at the bottom of the window lists the annotations:

Type	Set	Start	End	Annotation
agent	MPQA	0	0	[id=implicit]
agent	MPQA	0	0	[id=w]
agent	MPQA	217	238	[id=chavez]
attitude	MPQA	324	390	[attitude-neg-intensity=medium, attitude-neg-toward=n]
agent	MPQA	338	363	[id-gv, nested-source=w,gv]
attitude	MPQA	343	351	[attitude-neg-intensity=medium-high, attitude-neg-tow]
attitude	MPQA	368	390	[attitude-neg-intensity=medium, attitude-neg-toward=n]
attitude	MPQA	392	425	[attitude-neg-intensity=high, attitude-neg-toward=none]

MPQA

The [MPQA Opinion Corpus](#) contains news articles from a wide variety of news sources manually annotated for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, etc.).



The screenshot shows the GATE (General Architecture for Text Engineering) software interface. The main window displays a text document titled 'GATE document_0001F' with several paragraphs of text. The text is annotated with various colored boxes and labels, indicating the presence of entities, attitudes, and other linguistic features. A table at the bottom of the window lists the annotations, showing their type, set, start and end positions, and associated metadata.

Type	Set	Start	End	Annotations
agent	MPQA	0	0	[id=implicit]
agent	MPQA	0	0	[id=w]
agent	MPQA	217	238	[id=chavez]
attitude	MPQA	324	390	[attitude-neg-intensity=medium, attitude-neg-toward=n]
agent	MPQA	338	363	[id=gv, nested-source=w,gv]
attitude	MPQA	343	351	[attitude-neg-intensity=medium-high, attitude-neg-tow]
attitude	MPQA	368	390	[attitude-neg-intensity=medium, attitude-neg-toward=n]
attitude	MPQA	392	425	[attitude-neg-intensity=high, attitude-neg-toward=none]

INCEpTION

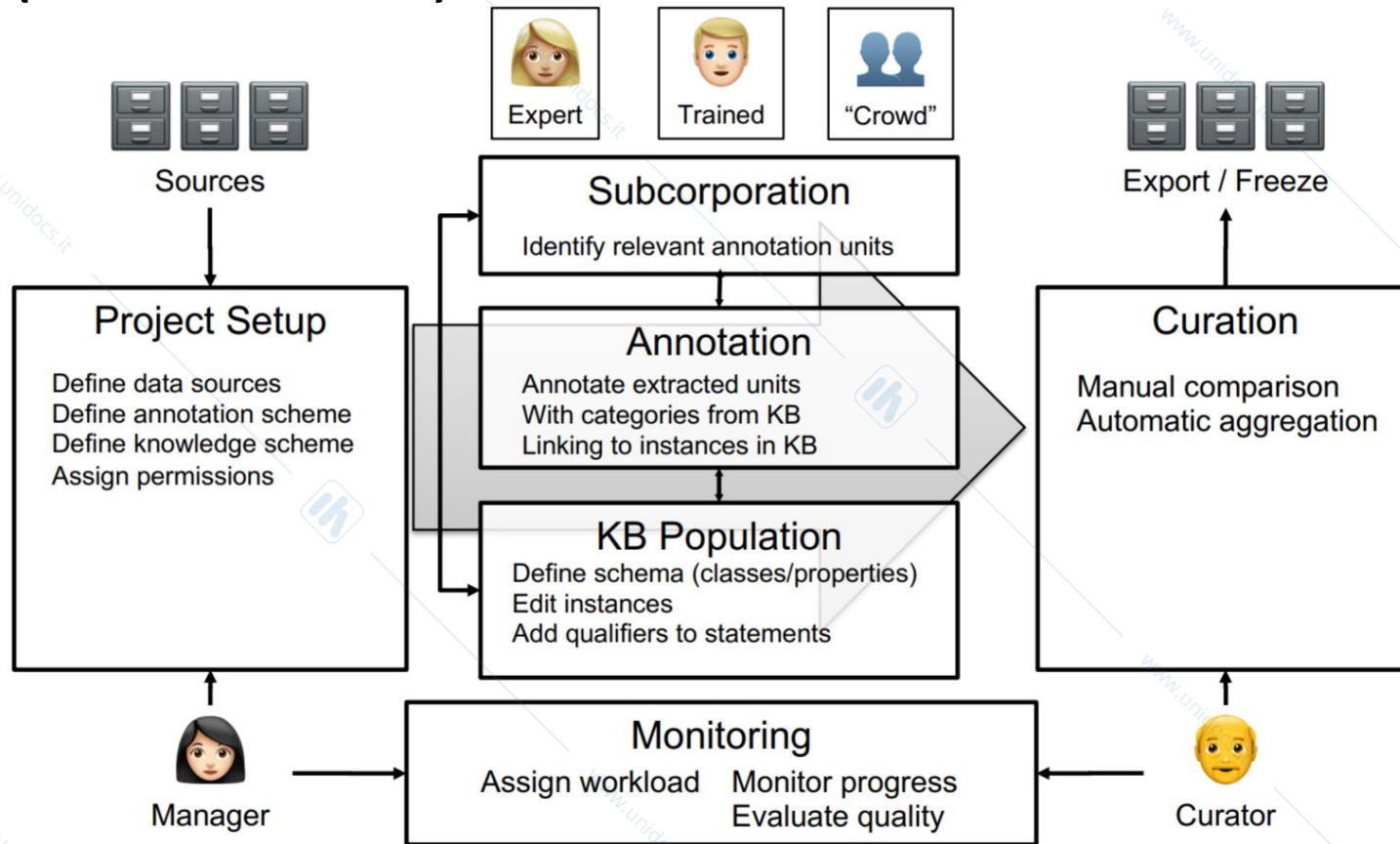
INCEpTION is an annotation platform that supports the definition of custom annotation schema, multi-annotator collaboration (with agreement measurement), and **interactive training of machine learning models for automatic annotation**.

The screenshot displays the INCEpTION interface with three main panels:

- Active Learning:** Shows session settings (Layer: Named entity, Terminate button), a recommendation for the text "Illinois" with Label "LOC", Score "1", and Delta "1". It includes "Accept", "Reject", and "Skip" buttons and a Learning History table.
- Annotation:** Shows a text snippet with annotations. The first sentence is "Barack Hussein Obama II born August 4, 1961) is an American politician who served as the 44th President of the United States from 2009 to 2017". The second sentence is "The first African American to assume the presidency, he was previously the junior United States Senator from Illinois from 2005 to 2008." The third sentence is "He served in the Illinois State Senate from 1997 until 2004." Annotations include "Barack Obama I PER", "born", "August 4, 1961)", "is an", "American politician", "served as the 44th", "President of the United States of America", "2009", "2017", "TIME", "Illinois River", "Illinois", "Illinois Senate", "LOC", and "LOC".
- Right Panel:** Shows a dropdown menu for "Layer" (Surface form), an "Annotation" section with "Delete" and "Clear" buttons, and a list of identifiers including "Illinois", "Illinois Senate", "Illinois River", "Governor of Illinois", "Alton", "Illinois Country", and "Illinois Territory".

Learning History	Label	Score	Delta	Action
Berkeley	http://www.wikidata.org/entity/Q168756	skipped		⊗
Tesla	PER	accepted		⊗
Tesla	PER	accepted		⊗
Tesla	PER	accepted		⊗
Tesla	PER	accepted		⊗
Tesla	PER	accepted		⊗
Science	OTH	rejected		⊗
Tesla	PER	accepted		⊗

INCEpTION: workflow



INCEpTION: assisted annotation

Active Learning

Session

Layer Named entity

Recommendation

Text Illinois

Label LOC

Score 1

Delta 1

Accept Reject Skip

Learning History

Text	Label	Score	Delta	Action
Berkeley	PER	0.95	0.95	skipped
Tesla	PER	0.95	0.95	accepted
Tesla	PER	0.95	0.95	accepted
Tesla	PER	0.95	0.95	accepted
Tesla	PER	0.95	0.95	accepted
Tesla	PER	0.95	0.95	accepted
Science	OTH	0.95	0.95	rejected
Tesla	PER	0.95	0.95	accepted

Annotation

Layer Surface form

Annotation Layer Named entity

Text Illinois

Identifier illi x

val: Illinois, Illinois Senate, Illinois River, Governor of Illinois, Alton, Illinois Country, Illinois Territory

Annotation Layer Named entity

Text position held

Identifier There is no statement in the KB which matches this SPO.

2) Subject Barack Hussein Obama II, Barack Obama

3) Object President of the United States, President of the United States of A...

4) Qualifiers end time 2017, 2017, start time 2009, 2009

Choose a relation Add

Human in the loop

Recommenders

- Continually learn from the users actions
- Asynchronous training does not slow down user interface
- Automatic evaluation to avoid inaccurate predictions (configurable quality threshold)
- Built-in recommenders: Dictionary-based, OpenNLP-based sequence classifier (part-of-speech, named entities, ...)

Active Learning

- Aims at reducing the time to learn by asking specific feedback from the user
- Using uncertainty-sampling strategy
- Compatible with any recommender that provides a confidence score
- User can freely switch between active learning and normal annotation

spaCy

Models from [spaCy](#) can be updated with new training data.
New models that annotate user-defined entities can be trained from scratch.

SpaCy uses (simple) custom data structures to represent training data, and supports data conversion from other formats (e.g., BILUO)

```
train_data = [  
    ("Uber blew through $1 million a week", [(0, 4, 'ORG')]),  
    ("Android Pay expands to Canada", [(0, 11, 'PRODUCT'), (23, 30, 'GPE')]),  
    ("Spotify steps up Asia expansion", [(0, 8, "ORG"), (17, 21, "LOC")]),  
    ("Google Maps launches location sharing", [(0, 11, "PRODUCT")]),  
    ("Google rebrands its business apps", [(0, 6, "ORG")]),  
    ("look what i found on google! 😊", [(21, 27, "PRODUCT")])  
  
doc = Doc(Vocab(), words=["Facebook", "released", "React", "in", "2014"])  
gold = GoldParse(doc, entities=["U-ORG", "O", "U-TECHNOLOGY", "O", "U-DATE"])
```

BEGIN	The first token of a multi-token entity.
IN	An inner token of a multi-token entity.
LAST	The final token of a multi-token entity.
UNIT	A single-token entity.
OUT	A non-entity token.

spaCy

```
optimizer = nlp.begin_training(get_data)
for itn in range(100):
    random.shuffle(train_data)
    for raw_text, entity_offsets in train_data:
        doc = nlp.make_doc(raw_text)
        gold = GoldParse(doc, entities=entity_offsets)
        nlp.update([doc], [gold], drop=0.5, sgd=optimizer)
nlp.to_disk("/model")
```

