

Extracción de relaciones semánticas

Ontología

¿Cómo construir una ontología?

Reglas escritas a mano, de tipo *pattern-matching*

Aprendizaje automático supervisado

Etiquetamiento manual

Problema:

Cómo evaluar el desempeño de estos algoritmos

Auto-supervisado

Bootstrap

Algoritmo de Dripe

Distant Supervision

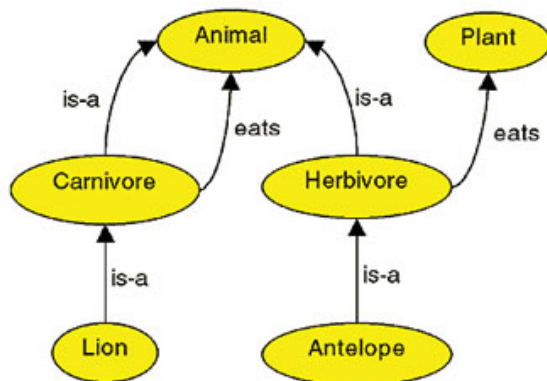
No supervisado para la web

Open Information Extraction

ECMes

Extractor de Conocimiento Mejorado en Español

Ontología



- **Is-a (hipónimo):** Jirafa es un rumiante es un mamífero es un vertebrado es un animal
- **Instance-of:** Buenos Aires es una instancia de ciudad

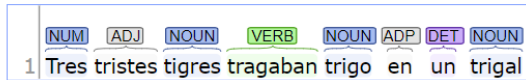
Forma de representar o estructurar el conocimiento.

Grafo dirigido

<https://corenlp.run/>

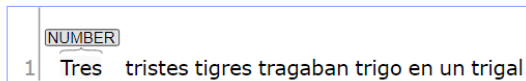
Part-of-Speech:

1 Tres tristes tigres tragaban trigo en un trigal



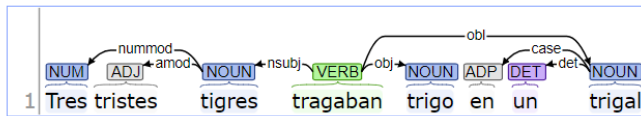
Named Entity Recognition:

1 Tres tristes tigres tragaban trigo en un trigal



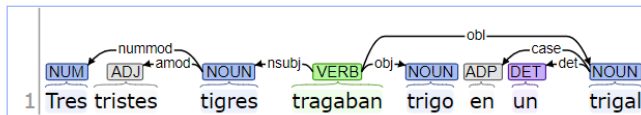
Basic Dependencies:

1 Tres tristes tigres tragaban trigo en un trigal



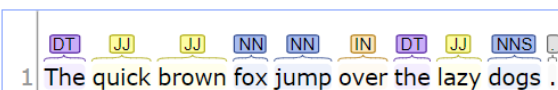
Enhanced++ Dependencies:

1 Tres tristes tigres tragaban trigo en un trigal



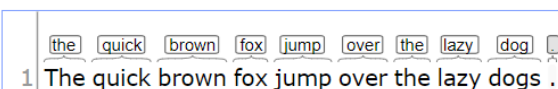
Part-of-Speech:

1 The quick brown fox jump over the lazy dogs .



Lemmas:

1 The quick brown fox jump over the lazy dogs .



Hace una tokenizacion → split por espacios, coma, punto y coma, signos de exclamacion, etc.

Usa diccionarios para saber la categoria gramatical

¿Cómo construir una ontología?

Hay varias maneras:

- Reglas escritas a mano, de tipo *pattern-matching*
- Aprendizaje automático supervisado

- Auto-supervisado
- No supervisado para la web (open information structure)

Reglas escritas a mano, de tipo *pattern-matching*

Reglas ontológica: IS-A

X e Y terminos que me interesan vincular

- Y como X ((, X*) (,and|or) X)
- Tanto Y como X
- X o otra Y
- X y otra Y
- Y incluyendo X
- Y, especialmente X

Patron	Ejemplos
X y otro Y	... templos, haciendas y otros edificios públicos importantes
X (o u) Y	... contusiones, heridas, fracturas u otras lesiones ...
Y como X	El laúd arco, como los bambara ndang ...
Y incluyendo X	... varios países, incluyendo Inglaterra, Canada ...
Y, sobre todo X	Países europeos , sobre todo Francia, Inglaterra y España ..

Pros de este método:

- Tienden a tener una alta precisión
- Puede ser adaptado a dominios específicos

Contras:

- Suelen tener muy bajo *recall* (exhaustividad)
- Implica una gran cantidad de trabajo pensar en todos los patrones posibles...Más aún para todas las relaciones
- Se puede mejorar la precisión con otros métodos

Aprendizaje automático supervisado

Pasos:

1. Decidir qué relaciones nos interesa extraer
2. Decidir qué nombres de entidades son pertinentes a dichas relaciones
3. Encontrar un conjunto de datos propicio (corpus)
 - a. Etiquetar las entidades detectadas en el corpus
 - b. Etiquetar a mano las relaciones entre esas dos entidades
 - c. Partir este conjunto de datos en entrenamiento y prueba
4. Entrenar un clasificador sobre el conjunto de entrenamiento

Etiquetamiento manual

1. Buscar dos nombres de entidades, generalmente en la misma oración.
2. Decidir si están o no relacionadas
3. Si lo están, clasificar la relación

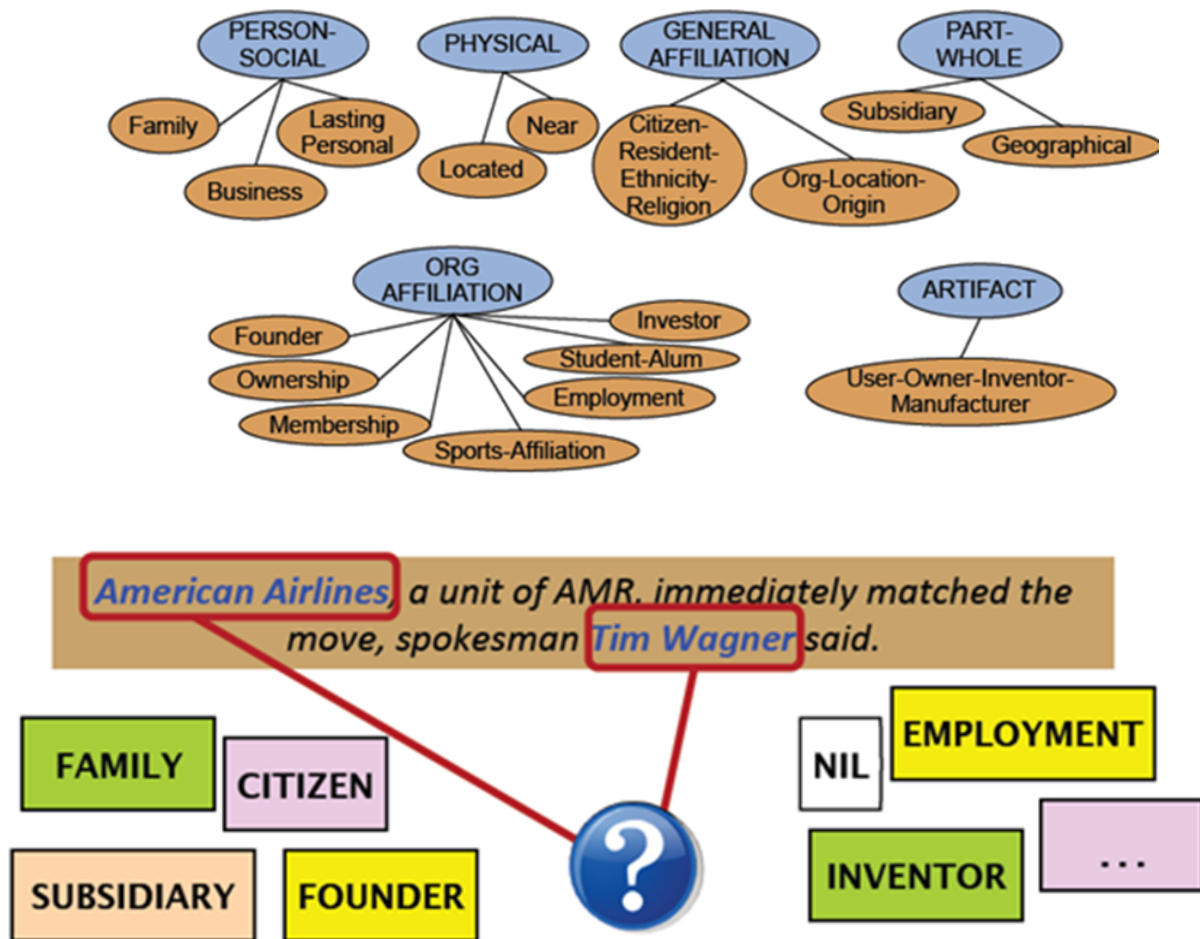


American Airlines, a unit of AMR. immediately matched the move, spokesman Tim Wagner said.

Detectamos dos entidades:

- American Airlines: una organizacion
- Tim Wagner: una persona

Me interesa encontrar la relacion entre estas dos entidades, pero no es la unica que me interesa, me interesan varias relaciones posibles



Bag of Word

Bayes Naive

Quitar stop words

Características: Palabras y qué más ?

Palabras

Entidades

Análisis Sintáctico

Stop words: palabras que tienen una frecuencia muy alta, que aparecen practicamente todo el tiempo en todos los textos y no aportan nada de valor, muy poco. Como por ejemplo: a, de, of, at, etc.

Que podemos utilizar para darle a Bayes como metodo de aprendizaje supervisado para detectar relaciones semanticas?

Mención1: American Airlines y **Mención2:** Tim Wagner

1. bigramas solo de las menciones: { American; Airlines; Tim; Wagner; “American Airlines” ; “Tim Wagner” }
2. **Agregar:**
Mención2 -1 : “Spokeman”
Mención2 +1: “Said”
3. Bolsa de palabras entre las entidades: {*unit, AMR, immediately, matched, move, spokeman, said*}

Bigramas: aquellas palabras que aparecen de forma inmediata, una detras de la otra en una oracion. Y las puedo juntar para armar una nueva palabra y darsela al algoritmo

Características relacionadas con las entidades detectadas:

1. Tipos de la entidades
 - a. Mención1: Organización
 - b. Mención2: Persona
2. Concatenación entre ambas: **Organización-Persona**

Características relacionadas con análisis sintáctico:

1. Utilización de la **categoría gramatical** de cada palabra o las secuencias de palabras
por ejemplo: NP, NP, PP, VP, NP, NP

- Utilización del path de cada constituyente del árbol sintáctico
por ejemplo: NP ↑ NP ↑ S ↓ S ↓ NP
- Utilización del árbol de dependencias. Indica como una parte de la oración depende de otra
por ejemplo: Airlines matched Wagner said

Part-of-Speech:

	<u>NNP</u>	<u>NNPS</u>	,	<u>DT</u>	<u>NN</u>	<u>IN</u>	<u>NNP</u>	,	<u>RB</u>	<u>VBD</u>	<u>DT</u>	<u>NN</u>	,	<u>NN</u>	<u>NNP</u>	<u>NNP</u>	<u>VBD</u>	.
1	American	Airlines	,	a	unit	of	AMR	,	immediately	matched	the	move	,	spokesman	Tim	Wagner	said	.

Lemmas:

	<u>American</u>	<u>Airlines</u>	,	<u>a</u>	<u>unit</u>	<u>of</u>	<u>AMR</u>	,	<u>immediately</u>	<u>match</u>	<u>the</u>	<u>move</u>	,	<u>spokesman</u>	<u>Tim</u>	<u>Wagner</u>	<u>say</u>	.
1	American	Airlines	,	a	unit	of	AMR	,	immediately	match	the	move	,	spokesman	Tim	Wagner	say	.

Named Entity Recognition:

	<u>ORGANIZATION</u>						<u>ORGANIZATION</u>										<u>PERSON</u>	
1	American	Airlines	,	a	unit	of	AMR	,	immediately	matched	the	move	,	spokesman	Tim	Wagner	said	.

Clasificadores que se pueden utilizar:

- MaxEnt
- Bayes Naïve
- SVM

Problema:

Requiere un volumen grande de datos etiquetados de forma manual, es decir, que nosotros tenemos que marcar en los textos la entidad y la relacion que hay entre ellas.

Cómo evaluar el desempeño de estos algoritmos

Precisión:

total de extracciones correctas / total de extracciones

Recall (exhaustividad o exactitud):

total de extracciones correctas / total de relaciones existentes

F1 (combina ambas medidas):

$2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$

Auto-supervisado

Estos metodos pueden generar su propio conjunto de entrenamiento etiquetado

Bootstrap

1. Para un par de “entidades” semilla cuya relación es conocida buscar coincidencias.
2. Extraer el contexto de la oración, en particular las “entidades” y reemplazarlas por *comodines*
3. Realizar búsquedas con esos patrones para encontrar nuevas entidades y repetir.

Ejemplo: PERSONA - LUGAR

- **Jorge Luis Borges** está enterrado en **Ginebra**, Suiza
 - **Patrón:** X está enterrado en Y
- La tumba de **Borges** está en **Ginebra**.
 - **Patrón:** la tumba de X está en Y
- **Ginebra** es el lugar de descanso final de **Borges**.
 - **Patrón:** X es el lugar de descanso final de Y

Entonces ahora tenemos patrones que los podemos buscar en el corpus y encontrar otras personas y lugares. Y al encontrar nuevas personas y lugares puedo buscar y encontrar nuevos patrones, y así sucesivamente hasta lograr tener un gran volumen de patrones que me van a servir para generar un conjunto de datos autoetiquetado.

Algoritmo de Dripe

Buscar patrones para encontrar relaciones entre autores y libros.

Partió de 5 semillas:

- Isaac Asimov => The Robots of Dawn
- David Brin => Startide Rising
- James Gleick => Chaos: making a new science
- Charles Dickens => Great Expectations
- William Shakespeare => The Comedy of Errors

Encontro patrones:

- The Comedy of Errors, **by** William Shakespeare, was
- The Comedy of Errors, **by** William Shakespeare, is
- The Comedy of Errors, **one of** William Shakespeare's earliest attempts
- The Comedy of Errors, **one of** William Shakespeare's most

Si utilizamos solo la parte común los patrones quedarían:

X, by Y,

X, one of Y's

Distant Supervision

Combina **Boostrapping** con **aprendizaje supervisado**.

1. En vez de usar solo 5 semillas utiliza una gran base de datos para obtener una enorme cantidad de entidades-semillas.
 2. Con los patrones que obtiene, extrae las características como se vio en el punto anterior
 3. Entrena un clasificador.
- En común con los supervisados
 - Usa un clasificador con varias características
 - No requiere iterar N veces para extraer los patrones
 - En común con los no-supervisados
 - Usa grandes cantidades de datos sin etiquetar
 - No es sensible a cómo se generó el corpus

Detalle del algoritmo

1. Para cada relación, por ejemplo: “nació-en”
2. Para cada tupla en una gran base de datos: <Borges, Buenos Aires>, < Albert Einstein, Ulm>, < Gabriel García Márquez, Aracataca>....
3. Encuentra sentencias en un gran corpus, en donde aparecen estas entidades semillas:
Borges nació en Buenos Aires; Einstein, quien nació en 1879 en Ulm; Gabriel García Márquez: lugar de nacimiento Aracataca, Colombia...
4. Extraer las características frecuentes. PERSONA nació en LUGAR, ...
5. Entrenar un clasificador, utilizando cientos de patrones. $P(\text{"nació-en"}|p_1, p_2 \dots p_{4000})$



Las relaciones las tenemos que conocer de antemano
Nosotros definimos cuales van a ser
Son fijas

No supervisado para la web

Open Information Extraction

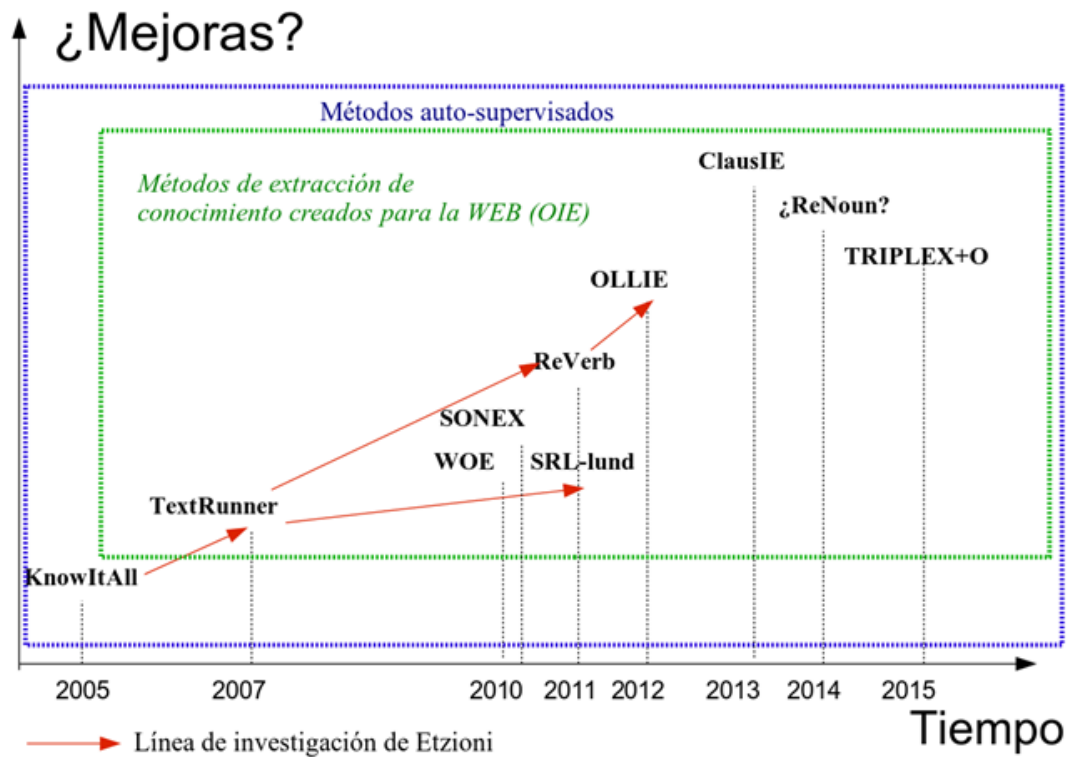
2005 **KnowItAll**

- no supervisado
- independiente del dominio
- escalable → soporta grandes volúmenes de datos

2007 **TEXTRUNNER**

Va a detectar todas las relaciones que el crea que existen en una oración

1. Realiza una sola pasada sobre el conjunto de datos como ya se mencionó
2. Extrae relaciones semánticas no definidas a priori. Puede extraer “cualquier cosa” que considere que es una relación entre dos entidades.



Año 2020

Tabla 8. Métodos evaluados en el artículo de [Kolluru et al., 2020a, p. 6].

Método	Bases de datos de prueba						
	Conjunto A		Conjunto B		Conjunto C		Conjunto D
	F1	AUC	F1	AUC	F1	AUC	F1
MinIE	0,419	-	0,384	-	0,523	-	0,285
ClausIE	0,450	0,220	0,402	0,177	0,610	0,380	0,332
OpenIE4	0,516	0,295	0,405	0,201	0,543	0,371	0,344
OpenIE5	0,480	0,250	0,427	0,206	0,599	0,399	0,354
SenseOIE	0,282	-	0,239	-	0,311	-	0,107
SpanOIE	0,485	-	0,379	-	0,540	-	0,319
RnnOIE	0,490	0,260	0,395	0,183	0,560	0,320	0,264
IMoJIE	0,535	0,333	0,414	0,222	0,568	0,396	0,360
OpenIE6	0,527	0,337	0,464	0,268	0,656	0,484	0,400

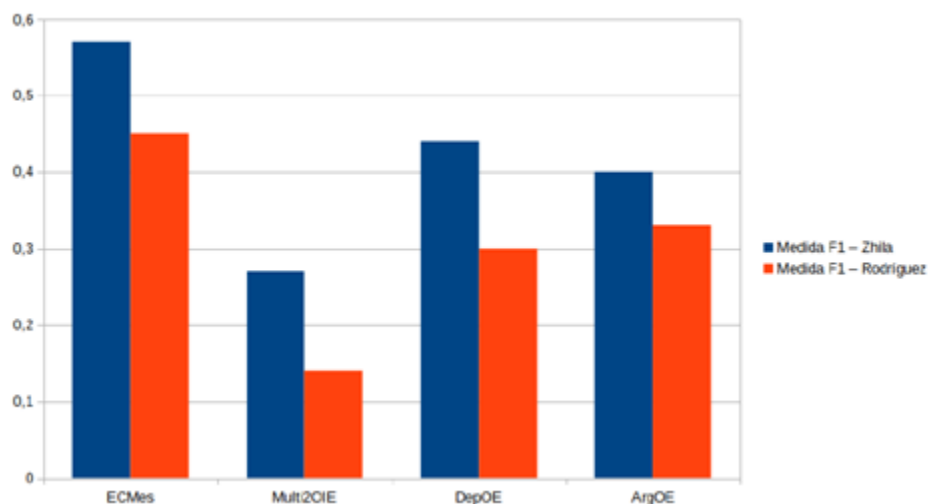
Estos son los metodos de Open Information Extraction

Idioma Español

Método	Artículo	Año	Lenguaje
DepOE	[Gamallo y Garcia, 2012]	2012	multilenguaje
ExtrHech	[Zhila y Gelbukh, 2014]	2014	español
ArgOE	[Gamallo y Garcia, 2015]	2015	multilenguaje
Multi ² OIE	[Ro et al., 2020]	2020	multilenguaje

ECMes Extractor de Conocimiento Mejorado en Español

2021



Funciona con el parser de Stanford

Albert Einstein, quien nació en Ulm, ganó el Premio Nobel

Análisis superficial

PROPN PROPN PUNCT PRON VERB ADP PROPN PUNCT VERB DET PROPN PROPN
Albert Einstein , quien nació en Ulm , ganó el Premio Nobel

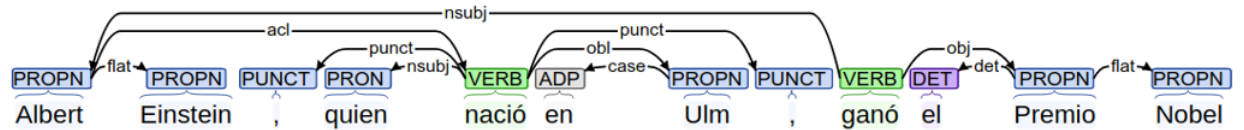
Reconocimiento de nombres de entidades

PERSON CITY MISC
Albert Einstein , quien nació en Ulm , ganó el Premio Nobel

Utiliza el parser de Stanford como biblioteca

Análisis superficial: análisis donde pongo solo categorías gramaticales

Árbol de dependencias sintácticas

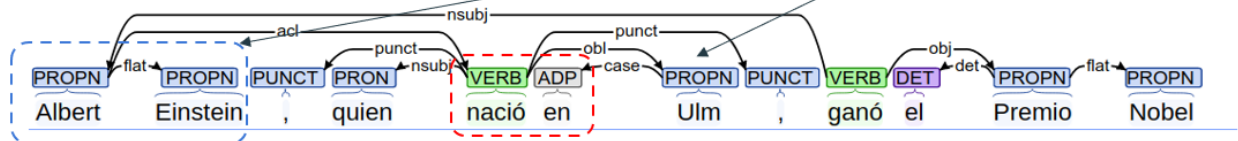


Utiliza un conjunto de datos etiquetados de la siguiente manera:

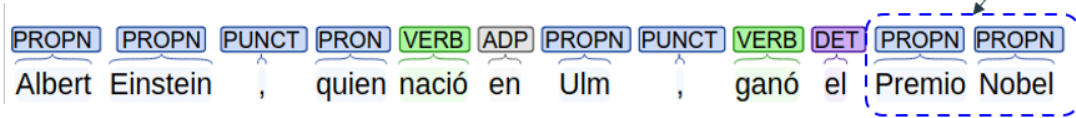
Albert Einstein, quien nació en Ulm, ganó el Premio Nobel

Crea patrones, para buscar en el árbol

- (Albert Einstein; nació en; Ulm)
- (Albert Einstein, ganó el; Premio Nobel)




Busca frase nominal para la 3 parte



Google Knowledge Graph

Introducing the Knowledge Graph: things, not strings

Search is a lot about discovery-the basic human need to learn and broaden your horizons. But searching still requires a lot of hard work by you, the user. So today I'm really excited to launch the

 <https://blog.google/products/search/introducing-knowledge-graph-h-things-not/>

