TP-OIE-ES: Método autónomo de extracción de relaciones semánticas para la Web en Español

Juan M. RODRÍGUEZ
Programa de Doctorado en Ciencias Informáticas. Facultad de Informática. Universidad Nacional de La Plata
La Plata, Buenos Aires (1900), Argentina

Hernán D. MERLINO
Grupo de Investigación en Sistemas de Información. Departamento de Desarrollo Productivo y Tecnológico. Universidad Nacional de Lanús.
Lanús, Buenos Aires (1900), Argentina

RESUMEN

En el siguiente trabajo se presenta un método novedoso de extracción de relaciones semánticas para la Web (*Open Information Extraction*) en español llamado TP-OIE-ES[1]. El mismo es capaz de aprender a partir de ejemplos, por lo cual tiene una autonomía mayor que aquellos métodos basados en reglas fijas, implementa además diversos mecanismos para evitar errores comunes en la extracción de relaciones semánticas. Por último TP-OIE-ES es comparado contra otros métodos en el estado del arte y obtiene, en una base de datos de prueba conocida, una mayor exhaustividad (*recall*) y una mayor media F1.

Palabras Claves: Extracción de conocimiento, extracción de relaciones semánticas, *open information extraction*, procesamiento de lenguaje natural.

1. INTRODUCCIÓN

La extracción de relaciones semánticas para la Web, en inglés *Open Information Extraction*, es un paradigma de extracción de información presentado por primera vez por Banko y otros en 2007 [1]. Los métodos de extracción de información que respetan este paradigma devuelven una tupla por cada relación semántica descubierta en un texto dado. La tupla tiene la forma (*argumento 1*, *relación*, *argumento 2*), en donde *argumento 1* y *argumento 2* son usualmente entidades —es decir objetos perfectamente identificables como personas, lugares, fechas, empresas, etc. — y la *relación* es la relación semántica entre ambos en una oración dada. Típicamente información fáctica del tipo: "Quien hizo qué a quién y cómo" [2].

Un método de *Open Information Extraction* debe cumplir las siguientes condiciones [1]:

- hacer una sola pasada por el corpus garantizando la escalabilidad, independientemente del tamaño del corpus.
- ser independiente del dominio
- tener una sola entrada (*input*): un corpus y su salida (*output*) debe ser un conjunto de relaciones extraídas.
- ser no supervisado.

[1] Todo el código fuente y el resultado de las pruebas puede encontrarse online: https://github.com/juanma1982/tp-oie-es

Para ejemplificar cómo estos métodos funcionan, considere la siguiente oración:

El cubo de Rubik es un rompecabezas mecánico tridimensional inventado por el escultor y profesor de arquitectura húngaro Ernő Rubik en 1974. **(ejemplo 1)**

Si extraemos las relaciones semánticas y las expresamos como tuplas, de la forma: (*argumento 1*, *relación*, *argumento 2*) obtendríamos las siguientes:

- (El cubo de Rubik, es un, rompecabezas mecánico tridimensional)
- (El cubo de Rubik, inventado por, Ernő Rubik)
- (El cubo de Rubik, inventado en, 1974)
- (Ernő Rubik, es, escultor y profesor de arquitectura húngaro)

Desde la presentación de TEXTRUNNER [1] muchos otros métodos fueron desarrollados, una lista actualizada puede encontrarse en [3].

En este documento presentamos un método novedoso de extracción de relaciones semánticas para la Web en idioma español llamado TP-OIE-ES (*Tree Pattern – Open Information Extraction* – Español). El algoritmo principal intenta identificar relaciones semánticas en una oración dada utilizando una lista de patrones. Estos patrones son aplicados sobre el árbol de dependencias sintácticas de la oración (enriquecido con información adicional: la categoría gramatical y nombres de entidades) hasta que encuentra alguna coincidencia. La coincidencia debe darse para la *relación* semántica propiamente dicha y para el *argumento 1*. El *argumento 2* es obtenido mediante una serie de reglas para buscar la frase nominal más próxima, tal y como lo hace ReVerb [4] en idioma inglés.

2. TRABAJOS RELACIONADOS

Investigaciones documentales [5], [3] muestran que fueron desarrollados diversos métodos de OIE. Solo algunos de ellos disponibles públicamente. De todos los que fueron presentados solo algunos fueron comparados de forma experimental con métodos ya existentes para determinar su desempeño relativo. La mayoría de ellos fueron creados para idioma inglés exclusivamente. Los más destacados en cantidad y calidad de

extracciones son ClausIE [6], ReVerb [4], OLLIE [7], MinIE [8], ArgOE [9], Stanford OpenIE [10], DepOE [11] y ExtrHech [12].

De la lista anterior solo funcionan con textos en idioma español ExtrHech, ArgOE y DepOE. ArgOE y DepOE fueron diseñados para soportar múltiples lenguajes [3]. ExtrHech soporta español e inglés [12].

3. PROBLEMAS ABIERTOS

Hay diversos problemas con los sistemas de Open IE en español los más significativos son como incrementar la precisión y la exhaustividad (*recall*). Otros problemas usuales son la falta de informatividad en las relaciones extraídas y el manejo de información subjetiva. Estos problemas serán discutidos en las siguientes secciones.

Precisión y exhaustividad
La precisión de tres de los métodos mencionados anteriormente para el idioma inglés (ReVerb, OLLIE, ClausIE), fue calculada en [2] y los resultados obtenidos se listan en la Tabla I.

TABLA I: Medidas calculadas en [2]

Medidas	ClausIE	OLLIE	Reverb
Precisión	0.467	0.456	**0.633**
Exaustividad (recall)	**0.519**	0.416	0.319
Medida-F1	**0.492**	0.435	0.424

Otro método en idioma inglés, el cual fue construido sobre ClausIE y según los autores supera en precisión y exhaustividad a los anteriores es MinIE [8].

Para los métodos en español: DepOE (2012), ExtrHech (2014) y ArgOE (2015), se muestra la precisión en la Tabla II. Corresponde a las pruebas realizadas por los autores de cada método, desafortunadamente no se cuenta con el valor de exhaustividad.

TABLA II: Medidas calculadas para métodos en español

Medidas	ExtrHech	ArgOE	DepOE
Precisión	0.55 [12]	0.55 [9]	**0.68** [11]

Falta de informatividad
Algunas extracciones, pueden ser correctas. Es decir que corresponden a un relación semántica presente en una oración, pero ser, sin embargo, poco útiles porque aportan poca o ninguna información relevante. Por ejemplo, en la siguiente oración:

El aumento también estuvo ligeramente por debajo de la tasa de crecimiento del 3,3% que el ministro de Finanzas, Michael Wilson, predijo para 1986 en el presupuesto de febrero.
(ejemplo 2)

Una extracción valida pero poco informativa sería:

(febrero, tiene, presupuesto)

La extracción anterior, y la oración son una traducción de un ejemplo real en idioma inglés. La extracción fue hecha por ClausIE. El problema es menos grave que el problema de no tener extracciones validas. En particular porque estas extracciones podrían ser descartadas en un paso posterior o bien incorporadas a una base general de conocimiento sin que los hechos importantes se vean afectados. Pero si este tipo de extracciones se computan como validas, pueden distorsionar los valores reales de precisión y exhaustividad del método. Idealmente, si es posible detectar extracciones poco informativa, el método debería descartarlas para generar "piezas de conocimiento" de mejor calidad, según la definición dada en [13], [14].

Manejo de información subjetiva
Considérese el siguiente ejemplo con su correspondiente relación semantica:

Los primeros astrónomos creían que la Tierra era el centro del universo. **(ejemplo 3)**

(la Tierra; era; el centro del universo)

Esta extracción es correcta desde el punto de vista sintáctico. Pero la información allí presente no es objetiva, corresponde a una opinión. Es información "no-factica". Este tipo de extracciones no son tenidas en cuenta de forma particular por métodos como Reverb o ClausIE. En cambio sí es manejada por métodos más nuevos como OLLIE y MinIE.

Por ejemplo MinIE anota cada extracción con información acerca de su "factualidad". MinIE representa la factualidad de una extracción con dos piezas de información: polaridad (+ o -) y modalidad (CT o PS; para indicar certeza o posibilidad) [8]. Sin embargo, los métodos que trabajan en idioma español no tienen en cuenta este problema.

4. TP-OIE-ES

En esta sección se explica cómo funciona TP-OIE-ES y que recaudos toma el método para lidiar con los problemas mencionados en la sección 3. Primeramente se explica cómo TP-OIE-ES extrae patrones a partir de ejemplos y qué forma tienen estos patrones. A continuación se explica cómo es el proceso de extracción.

Proceso de aprendizaje
Este proceso no es un paso obligatorio. El entrenamiento fue realizado por los autores de antemano utilizando una base de datos de ejemplos y no es necesario un reentrenamiento. Pero, si se agregaran más ejemplos a dicha base de datos, TP-OIE-ES podría ser reentrenado para mejorar, en principio, su exhaustividad.

TP-OIE-ES es capaz de generar patrones a partir de estos ejemplos. La base de datos de ejemplos consiste en un archivo JSON con el siguiente formato:

{examples:[{sentence:"",
relations: [{entity1:", relation:"", entity2:""},...]
},...]}

Este objeto JSON tiene un solo atributo llamado *examples*, el cual es un *array* de objetos JSON. Cada uno de estos objetos, contiene dos atributos: *sentence* y *relations*. El primero es una oración de ejemplo y el segundo un *array* de objetos JSON, siendo este último objeto una representación de una relación

semántica existente en la oración contenida en el campo *sentence*.
Una oración de ejemplo (*sentence*) podría ser:

Albert Einstein fue galardonado con el Premio Nobel en Suecia en 1921. **(ejemplo 4)**

Y una relación semántica existente (*relations*) podría ser:

(Albert Einstein, fue galardonado con, el Premio Nobel)

Para este ejemplo, (cómo para cualquier otro) TP-OIE-ES ejecutará un *parser* de dependencias sintácticas para obtener un árbol de dependencias. El *parser* utilizado por TP-OIE-ES es *depparse* de la biblioteca *Stanford CoreNLP* [15]. En el mismo proceso TP-OIE-ES identificará los nombres de entidades (NER) de la oración usando la anotación "ner" del mismo *parser Stanford CoreNLP*. Una representación gráfica de este árbol se muestra en la Figura 1.

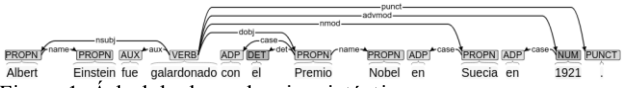

Figura 1: Árbol de dependencias sintáctico.

Luego para cada relación semántica en el ejemplo, TP-OIE-ES intentará obtener el patrón que le servirá luego para extraer relaciones. De acuerdo con Fader [4] conviene siempre encarar la extracción de relaciones semánticas empezando por la *relación* propiamente dicha, entonces para cada unigrama en la *relación,* TP-OIE-ES intentará extraer el *path* en el árbol de dependencias sintáctico. Siendo el *path* la secuencia de aristas en el árbol, desde la raíz (en este caso el verbo) hasta el unigrama. Por ejemplo, el unigrama: "Nobel" tiene el siguiente *path: root→nmod→name. C*ontinuando con el ejemplo, la relación "fue galardonado" tiene dos unigramas "fue" y "galardonado", el *path* de cada palabra en el árbol es el siguiente:

1. **fue**: *root* → *auxpass*
2. **galardonado**: *root*

A continuación se añade información adicional a cada unigrama, si es entidad (NER) se agrega la anotación "ner", por ejemplo: "ner=PERSON". Si no es una entidad, se utiliza la categoría gramatical, y se añade la anotación "pos", ejemplo "pos=VERB". Pero si la categoría gramatical es "ADP" (adposición) se pone la etiqueta "word" y se indica directamente el unigrama. En el ejemplo anterior quedaría así:

1. **fue**: *root* → *auxpass* [pos=VBD]
2. **galardonado**: *root* [pos=VBD]

Por último se genera el patrón de extracción con la forma de un árbol. En donde la raíz será el *path* al primer unigrama (con la especificación adicional: *ner*, *pos* o *word*). En el ejemplo anterior "root→auxpass[pos=VBD]". Luego el segundo unigrama será un hijo de esta raíz. El último de los *paths* será marcado como un nodo hoja. Luego de entrenar a TP-OIE-ES con varios ejemplos, habrá varios árboles y estos árboles podrán crecer y tener varios hijos. En estos casos un nodo originalmente marcado como hoja podrá luego ser padre. En ese caso se lo marca para indicar que allí termina un patrón valido de extracción. Estos patrones también se guardan en un archivo JSON y se cargan siempre que TP-OIE-ES inicia en modo de extracción. En el archivo JSON, cada nodo tiene la siguiente forma:

```
"root auxpass[pos=VBD]": {
"isLeaf": false,
"nextPatterns": { "root[pos =VBN]": {
"isLeaf": true,
"nextPatterns": {}
} } }
```

Una vez obtenido el patrón correspondiente a la *relación*, se procede a extraer el patrón correspondiente al *argumento 1*. Este paso es similar al anterior. Los argumentos primeros se extraen utilizando la misma lógica. Siguiendo con el ejemplo 4, el sujeto: "Albert Einstein" generará un patrón como el que sigue en formato JSON:

```
"root nsubjpass compound[ner=PERSON]": {
"isLeaf": false,
"nextPatterns": { "root nsubjpass[ner=PERSON]": {
"isLeaf": true,
"nextPatterns": {}
} } }
```

La única diferencia entre estos patrones y los patrones para extraer *relaciones* es que estos patrones se almacenan referenciados a un patrón de *relación*. De esta forma al momento de producirse una coincidencia entre una oración y un patrón de *relación*, no se realiza una búsqueda exhaustiva en la lista de patrones de *argumentos* sino que solo se prueban los patrones asociados, lo cual es más eficiente.

Conjunto de entrenamiento
TP-OIE fue entrenado utilizando tres conjuntos de datos diferentes que son subconjuntos de los construidos por Luciano Del Corro y Rainer Gemulla en [6]. Todas estas oraciones y sus correspondientes relaciones semánticas están en idioma inglés y están constituidos de la siguiente forma:

- **Conjunto de datos de ReVerb**: consta de 322 oraciones obtenidas a través del servicio de enlace aleatorio de Yahoo. Contienen mucho ruido.
- **Conjuntos de datos de Wikipedia**: consiste en 102 oraciones obtenidas al azar de las páginas de Wikipedia. Estas oraciones son más cortas, más simples y menos ruidosas que al conjunto anterior.
- **Conjunto de datos del New York Times**: consta de 131 oraciones aleatorias de la colección [16]; estas oraciones son generalmente muy limpias pero tienden a ser largas y complejas.

Para todas estas oraciones se tomaron las extracciones válidas hechas por ClausIE en [6], además se agregaron 12 ejemplos a mano. El número total de oraciones en el conjunto de entrenamiento fue de 567 y el número total de diferentes relaciones semánticas es 1425.

Dado que el árbol de dependencias generado por el *parser* de dependencias sintáctico es universal en el sentido de que las aristas que conectan las palabras son siempre las mismas independientemente del idioma [17], es posible usar los mismos patrones generados para idioma inglés en idioma español.

Por otro lado, las categorías gramaticales utilizadas por el *parser Stanford CoreNLP* en español son conocidas como *Universal POS tags* [18] y son distintas a las utilizadas en idioma inglés conocidas como *Penn Treebank POS tags* [19], para ello se convirtieron utilizando las equivalencias mostradas en la Tabla III.

Se utilizó una base de datos de entrenamiento en idioma inglés por dos razones principales, la primera es por la disponibilidad de los datos: no se cuenta con la misma cantidad en idioma español y la segunda es que se quiere crear un método de Open IE que funcione en idioma inglés, bajo las mismas premisas que TP-OIE-ES.

TABLA III: Conversión de categorías gramaticales

Penn Treebank POS tags a Universal POS tags			
#→SYM	EX→PRON	NNPS→PROPN	UH→INTJ
$→SYM	FW→X	NNS→NOUN	VB→VERB
"→PUNCT	HYPH→PUNCT	PDT→DET	VBD→VERB
, →PUNCT	IN→ADP	POS→PART	VBG→VERB
-LRB-	JJ→ADJ	PRP→PRON	VBN→VERB
→PUNCT	JJR→ADJ	PRP$→DET	VBP→VERB
-RRB-	JJS→ADJ	RB→ADV	VBZ→VERB
→PUNCT	LS→X	RBR→ADV	WDT→DET
. →PUNCT	MD→VERB	RBS→ADV	WP→PRON
: →PUNCT	NIL→X	RP→ADP	WP$→DET
AFX→ADJ	NN→NOUN	SYM→SYM	WRB→ADV
CC→CCONJ	NNP→PROPN	TO→PART	. →PUNCT
CD→NUM			
DT→DET			

Proceso de extracción
El proceso de extracción es el proceso por el cual TP-OIE-ES extrae relaciones semánticas de textos dados como entrada. Consta de los siguientes pasos para un texto dado como entrada:

1. Divide cada párrafo en oraciones.
2. Para cada oración ejecuta el *parser* de dependencias sintácticas y un *parser* de análisis sintáctico superficial (*NP-Chunking*). El *parser* de dependencias sintácticas es el mismo utilizado en el proceso de aprendizaje. El *parser* de *NP-chunking* es el de biblioteca *OpenNLP*, el mismo utilizado por ReVerb en [4].
3. TP-OIE genera internamente un XML en memoria que representa el árbol de dependencias sintácticas enriquecido con las etiquetas NER, POS y WORD descriptas. Estas indican el tipo de entidad, la categoría gramatical y la palabra. La razón de construir un XML es la de poder utilizar JSoup[2] una poderosa herramienta de Java para poder encontrar patrones en arboles, habitualmente en el árbol DOM de HTML.
4. Luego para cada patrón de *relación* existente en la base de datos de TP-OIE-ES, JSoup intenta buscar una coincidencia. Si el primer elemento coincide sigue con los hijos. Así hasta encontrar un hijo que esté marcado como hoja. Una vez que tiene la *relación*, busca la lista asociada de patrones para *argumentos 1*. Y realiza una búsqueda similar.
5. El método de extracción de *argumentos 2* es diferente. En este punto TP-OIE-ES utiliza el mismo enfoque que ReVerb. Intenta encontrar la frase nominal más cercana a la derecha de la relación en la oración. Si el *argumento 1* está a la derecha de la relación busca la frase nominal más cercana a izquierda.

[2] https://jsoup.org/

5. MANEJO DE ERRORES COMUNES

En esta sección describimos como TP-OIE-ES maneja los errores descriptos en la sección 3.

Mejora en la precisión y exhaustividad
La precisión y la exhaustividad de TP-OIE-ES fue medida originalmente en idioma inglés, se utilizó para ello un conjunto de datos basado en el conjunto de cables Reuters-21578 [2]. Para descartar las extracciones erróneas y mejorar así la precisión se implementó un sistema de puntaje para cada extracción. El sistema es similar al que utiliza ReVerb [4] con algunas modificaciones. La principal es que le asigna un puntaje muy malo a extracciones en donde el argumento es una sola palabra y esta es un determinante, un pronombre posesivo o un conector entre frases. Otras validaciones adicionales que se agregaron fueron las siguientes: que los *argumentos 1* y *2* sean distintos, que la *relación* no esté dentro del *argumento 1*, que el *argumento 1* no termine con la misma palabra con la cual comienza la *relación*.

Para mejorar la exhaustividad (*recall)* se fueron agregando más y más ejemplos al conjunto de entrenamiento con lo cual se fueron añadiendo más patrones de extracción. Al principio se tenía una precisión muy alta, cercana al 70% pero con una exhaustividad muy pobre de apenas 15%. Finalmente, luego de agregar varios ejemplos en varios entrenamientos, la precisión se estabilizó en 65% y la exhaustividad creció hasta alcanzar un 25% (siempre probando con el conjunto de datos en inglés).

Falta de informatividad
Para mejorar la calidad de las extracciones y hacerlas más informativas, se verifica si los *argumentos*, dentro de la oración, están unidos a una frase nominal mediante algún conector como: "a", "de", "en", "los". En ese caso se agrega al argumento el conector y la frase nominal.

Por otro lado TP-OIE-ES intentará mantener juntos todos los unigramas pertenecientes a una misma *entidad* que haya sido detectada como tal por el *parser CoreNLP de Stanford.*

Manejo de información subjetiva
Para detectar información subjetiva, TP-OIE-ES tiene dos características: la primera consiste en reemplazar todo texto entrecomillado por una palabra comodín que será analizada como un sustantivo simple. De esta manera evita mezclar en la misma extracción algo que claramente está separado de la semántica original de la misma. La segunda característica es un tratamiento especial para extracción de argumentos cuando la relación contiene palabras como "dijo", "creyó", etc. En este caso el *argumento* se tomará como todas las palabras desde la relación hasta el primer signo de puntuación. En ambos casos, este *argumento* se volverá a analizar como una nueva oración separada de la original. Esta nueva sentencia se marcará como dependiente de la original y por lo tanto como subjetiva. En la oración del **ejemplo 3**, TP-OIE-ES realizará las siguientes extracciones:

- *1 (Los primeros astrónomos, creían, que la tierra era el centro del universo) => (44)*
- *5 (la tierra, era, el centro del universo) => (23) DEPENDS OF 1*

La primera está identificada con el ID 1, la segunda con el ID 5 (hubo en el medio otras extracciones descartadas) se indica entre paréntesis el puntaje de cada relación y luego en la segunda se indica que depende de 1 y es por ello *no-fáctica*.

Para que TP-OIE-ES muestre toda esta información hay que ejecutarlo con el parámetro *-full*, de lo contrario solo muestra las extracciones.

6. RESULTADOS Y CONCLUSIONES

Para medir la precisión, la exhaustividad (*recall*) y la medida F1 se utilizó una base de datos de 69 sentencias extraídas de Wikipedia y propuestas por Gamallo en [11]. Se comparó TP-OIE-ES con otros dos métodos de Open IE en español. El más recientemente publicado: ArgOE **[**9**]** y el más preciso: DepOE [11].

Se detallan a continuación las formulas utilizadas para calcular la precisión, la exhaustividad y la medida F:

$$\text{Precisión} = \frac{\text{extracciones consideradas correctas}}{\text{total de extracciones}} \quad (1)$$

$$\text{Exhaustividad} = \frac{\text{extracciones consideradas correctas}}{\text{totalidad de hechos facticos}} \quad (2)$$

$$F_\beta = \frac{(1+\beta^2) \cdot \text{Precisión} \cdot \text{Exhaustividad}}{(\beta^2 \cdot \text{Precisión}) + \text{Exhaustividad}} \quad (3)$$

En Ec(2) la variable “totalidad de hechos facticos” se tomó el valor dado por Gamallo en [11] que es 137.

En la Ec(3) el parámetro β se estableció igual a 1, para que la precisión y la exhaustividad tuviesen el mismo peso en la fórmula. Por ello nos referimos a la medida F directamente como medida F1 o simplemente F1.

A partir de los resultados obtenidos y que se resumen en la Tabla IV, es posible concluir que TP-OIE-ES es un método dentro del estado-del-arte de los métodos de extracción de relaciones semánticas para la Web en español. También es notorio que se debe trabajar aún más en mejorar su precisión y su exhaustividad.

TABLA IV: Resultados obtenidos

Medidas	TP-OIE-ES	DepOE	ArgOE
Precisión	0.62	**0.89**	0.67
Exhaustividad (*recall*)	**0.36**	0.29	0.29
Medida-F1	**0.46**	0.44	0.40

7. FUTURAS LINEAS DE INVESTIGACIÓN

Entre los próximos trabajos se pretende añadir más ejemplos en la base de datos de entrenamiento de TP-OIE-ES, (preferentemente en idioma español) para intentar subir su exhaustividad, al mismo tiempo se buscará refinar el sistema de puntaje para evitar que el aumento en la exhaustividad repercuta negativamente en la precisión.

Por otro lado se buscará crear un conjunto de evaluación más grande y diverso para una nueva comparación.

8. REFERENCIAS

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction for the web," **IJCAI**, vol. 7, pp. 2670-2676, January 2007.

[2] Juan M. Rodríguez, Hernán D. Merlino, Patricia Pesado, and Ramón García-Martínez, "Evaluation of open information extraction methods using Reuters-21578 database," in **2nd International Conference on Machine Learning and Soft Computing (ICMLSC '18)**, 2018, pp. 87--92.

[3] Rafae Glauber and Barreiro Claro Daniela, "A systematic mapping study on open information extraction," in **Expert Systems with Applications**, 2018, pp. 372--387.

[4] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in **Association for Computational Linguistics**, 2011, pp. 1535-1545.

[5] J. M. Rodríguez, H. Merlino, and R. García-Martínez, "Revisión Sistemática Comparativa de Evolución de Métodos de Extracción de Conocimiento para la Web," in **XXI Congreso Argentino de Ciencias de la Computación (CACIC 2015)**, Buenos Aires, Argentina, 2015.

[6] L. Del Corro and R. Gemulla, "ClausIE: clause-based open information extraction," in **22nd international conference on World Wide Web**, 2103, pp. 355-366.

[7] M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open language learning for information extraction," in *2012 Joint Conference on Empirical Methods in* **Natural Language Processing and Computational Natural Language Learning**, 2012, pp. 523-534.

[8] Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro, "Minie: minimizing facts in open information extraction," in *Proceedings of the 2017 Conference on Empirical Methods in* **Natural Language Processing**, 2017, pp. 2630--2640.

[9] Pablo Gamallo and Marcos Garcia, "Multilingual open information extraction," in **Portuguese Conference on Artificial Intelligence**, 2015, pp. 711--722.

[10] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning, "Leveraging linguistic structure for open domain information extraction," in **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing**, vol. 1, 2015, pp. 344--354.

[11] Pablo and Garcia, Marcos and Fern, "Dependency-based open information extraction," in **Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP**, 2012, pp. 10--18.

[12] Alisa Zhila and Alexander Gelbukh, "Comparison of open information extraction for English and Spanish," in **Computational Linguistics and Intelligent Technologies**, vol. 12, number 19, pp. 714--722.

[13] C. Rancan, A. Kogan, P. Pesado, and R. García-Martínez, "Knowledge discovery for knowledge based systems. Some experimental results," in **Research in Computing Science Journal**, vol. 27, 2007, pp. 3--13.

[14] A. Gómez, N. Juristo, C. Montes, and J. Pazos, **Ingeniería del conocimiento**.: Editorial Centro de Estudios Ramón Areces, 1997.

[15] Danqi Chen and Christopher Manning, "A fast and accurate dependency parser using neural networks," in **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**, 2014, pp. 740--750.

[16] Sandhaus Evan, "The new york times annotated," , 2008.

[17] S. Buchholz and E. Marsi, "CoNLL-X shared task on multilingual dependency parsing," in **Proceedings of the tenth conference on computational natural language learning**, 2006, pp. 149-164.

[18] S., Das, D., McDonald, R. Petrov, "A universal part-of-speech tagset," , 2011.

[19] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in **Conference on Empirical Methods in Natural Language Processing**, 1996.