



Identificación y extracción de relaciones entre entidades empleando árboles de dependencia

Identification and extraction of relationships between entities using dependency trees

Orlando Ramos-Flores¹ , David Pinto¹ 

¹Benemérita Universidad Autónoma de Puebla, Puebla, México.
orlando.ramos@alumno.buap.mx, dpinto@cs.buap.mx

(Recibido: 28 julio 2021; aceptado: 14 septiembre 2021; Publicado en Internet: 1 diciembre 2021)

Resumen. En este trabajo se presenta un enfoque no supervisado para identificar y extraer relaciones entre dos entidades nombradas. El enfoque se conforma por casos, estableciendo un conjunto de patrones para identificar relaciones previamente establecidas. Además, se estudia un conjunto de casos para identificar y extraer relaciones de forma automática. Se emplean las dependencias universales *appos* y *amod*, así como los elementos clave de la oración: el *verbo* entre dos entidades nombradas, y el *sujeto* y objeto. Este proceso se realiza de forma automática sobre documentos no estructurados en el dominio de noticias políticas en idioma español. Para verificar las relaciones se realizó una evaluación manual sobre un conjunto seleccionado.

Palabras clave: Extracción de relaciones, Árboles de dependencia, Noticias políticas.

Abstract. In this paper, we present an unsupervised approach to identify and extract relationships between two named entities. The approach is made up of cases, establishing a set of patterns to identify previously established relationships. In addition, a set of cases is studied to identify and extract relationships automatically. The universal dependencies *appos* and *amod* were used, as well as the sentence's key elements, such as the *verb* between two named entities, and the *subject* and *object*. This process is carried out automatically on unstructured documents in the domain of political news in Spanish. We made a manual evaluation on a selected set to verify the relationships extracted.

Keywords: Relation extraction, Dependency trees, Political news.

Tipo de artículo: Artículo de investigación.

1 Introducción

La Extracción de Relaciones (ER) es una subtarea de la Extracción de la Información (EI) y consiste en identificar, extraer y estructurar los datos contenidos en documentos de texto. La EI generalmente se divide en tres subproblemas: Resolución de Correferencia, Reconocimiento de Entidades Nombradas, y Extracción de Relaciones (Bunescu & Mooney, 2005). En este trabajo se presenta un enfoque no supervisado para identificar y extraer relaciones entre dos entidades nombradas. Las entidades nombradas han sido previamente definidas, como se describe en la [Tabla 1](#). La ER es una tarea que no se ha resuelto del todo, aunque se han abordado diferentes métodos no supervisados principalmente empleando agrupamiento (Hasegawa et al., 2004; Yan et al., 2004), usando modelos probabilísticos sobre Extracción de Información Abierta (EIA) (Etzioni et al., 2008), y recientemente sobre la misma línea (EIA) empleando frases verbales y basadas en cláusulas (Vo & Bagheri, 2017).

El aporte de este trabajo se centra en la identificación y extracción de relaciones basado en un conjunto de casos. El enfoque propuesto permite la extracción de relaciones definidas con antelación, además de poder extraer relaciones de forma automática, es decir, no se presupone un conjunto de relaciones. La relación se obtiene intrínsecamente de la oración, y es formada de una o más palabras, y no necesariamente se tienen que encontrar entre ambas entidades. En este proceso se hace uso de las dependencias universales en todos los casos, así como la etiqueta POS de las palabras. El trabajo puede ser consultado en <https://github.com/orlandxrf/relation-extraction>.

La organización del documento se describe a continuación. En la Sección 2 se describen trabajos previos abordando esta tarea con árboles de dependencia. El conjunto de datos es descrito en la Sección 3. En la

Sección 4 se presenta el método propuesto sobre cada caso para la identificación y extracción de relaciones. La Sección 5 presenta los resultados obtenidos. Finalmente, en la Sección 6 se presentan las conclusiones.

2 Revisión de la literatura

En el trabajo de Bunescu & Mooney (2005), los autores proponen un método para extraer entidades entre dos entidades nombradas, usando el análisis de dependencias al generar árboles de dependencia. Proponen obtener la ruta más corta entre ambas entidades, y ocurre cuando ambas entidades están conectadas al predicado. Emplean el corpus *Automated Context Extraction* (ACE) con relaciones y entidades anotadas. En sus experimentos emplearon un clasificador *Support Vector Machine* (SVM) obteniendo un *F-score* de 52.5 %.

El uso de árboles de análisis de dependencias es usado por Fundel et al. (2007), en el cual usan un conjunto pequeño de reglas, un diccionario de sinónimos que contienen nombres de proteínas y genes, y una lista de términos de restricción que se utilizan para describir relaciones de interés. Las relaciones se crean extrayendo rutas que conectan pares de proteínas de árboles de análisis de dependencia. Se identifican y extraen tres tipos de relaciones. Previo a la evaluación realizaron una evaluación manual. Obtienen un *F-score* de 78 % en la evaluación.

Frases de análisis de dependencia son definidas en el trabajo de Wu et al. (2009). Estas frases están compuestas por características de productos y expresiones de opinión. Extraen la relación que existe entre las mismas a través de un subárbol (frases de análisis de dependencia) y debe tener una distancia en el árbol menor a cinco hacia la relación. El corpus fue anotado de forma manual y usaron un clasificador SVM, con un *F-score* de 46.3 %.

Un enfoque no supervisado para extraer relaciones usando el análisis de dependencias es presentado por Afzal et al. (2011). En primera instancia reconocen entidades nombradas con *GENIA tagger*, después reemplazan el texto de la entidad por su clase dentro de cada oración y aplican el análisis de dependencia sobre esta nueva oración. Del árbol generado obtienen los patrones candidatos, para luego jerarquizarlos en base a su relevancia en el corpus usando diferentes métricas para ello. En sus evaluaciones de las métricas usando *Chi-Square* y *Normalised Mutual Information* presentan los mejores resultados en base a su precisión en los patrones de árboles de dependencia.

Otro enfoque no supervisado para la extracción de relaciones es el descrito por Quan et al. (2014). Usan agrupaciones de patrones basada en un kernel polinomial (vectores de características) para identificar la interacción de palabras. Esas palabras se combinan con el análisis sintáctico de la estructura de la frase y el análisis de la dependencia para la extracción de relaciones. Utilizan un clasificador *k-nearest neighbors* (KNN), en su evaluación usan el conjunto de datos AImed que es un *benchmark* para interacciones entre dos proteínas, además se comparan con otros trabajos del estado del arte. En sus resultados alcanzan un *F-score* de 82.05 % sobre el primer conjunto de datos y un *F-score* de 47.20 % en el segundo conjunto de datos.

En el trabajo de Miwa & Bansal (2016) proponen un modelo con arquitectura *long short-term memory recurrent neural networks* (LSTM-RNN) que extrae relaciones entre entidades en secuencias de palabras y estructuras de árboles de dependencia (Li et al., 2015). Su modelo es capaz de detectar entidades nombradas y relaciones. Hacen uso de la estructura con la ruta más corta, un subárbol, y el árbol completo, además de un conjunto de características adicionales. En sus experimentos usan el corpus ACE 2004 y ACE 2005, obteniendo un *F-score* de 48.4 % y de 55.6 % respectivamente.

3 Conjunto de datos

El conjunto de datos utilizado consta de 32.147 documentos de noticias políticas. El conjunto de datos contiene quince clases de entidades nombradas como se describe en la [Tabla 1](#). La columna FU representa al número de entidades únicas y la columna FT enumera el total de entidades nombradas por clase.

Para cada uno de los documentos se identificaron sus oraciones y se filtraron aquellas con al menos dos entidades nombradas. Cuando existen más de dos entidades en una oración, se crea una ventana para visualizar únicamente dos entidades por oración. Por ejemplo, una oración s con tres entidades e_1, e_2, e_3 se

convierte en dos oraciones: la primera oración s_1 con las entidades e_1 y e_2 , y la segunda oración s_2 con las entidades e_2 y e_3 .

Tabla 1. Entidades nombradas empleadas en la extracción de relaciones.

No.	Clase	Entidad	FU	FT
1	PER	Persona	15,132	110,455
2	ORG	Organización	8,485	75,066
3	TIT	Título	7,903	78,801
4	GPE	Geopolítica	2,152	26,707
5	MNY	Moneda	1,810	3,650
6	DAT	Fecha	1,785	13,027
7	FAC	Instalación	1,499	6,591
8	DOC	Documento	692	2,360
9	EVT	Evento	424	2,555
10	PRO	Producto	407	1,490
11	PEX	Partido Político	344	20,260
12	TIM	Tiempo	324	5,147
13	AGE	Edad	129	1,489
14	DEM	Gentilicio	99	3,793
15	LOC	Lugar	27	117

4 Identificación y extracción de relaciones

Un enfoque no supervisado se usó para abordar esta tarea usando el análisis de dependencias para las oraciones. En primera instancia, se realizó un estudio sobre el análisis de dependencias en oraciones generando sus respectivos árboles, para ello se empleó el modelo en español de la biblioteca de Python Spacy¹ para el análisis, y las bibliotecas Networkx² y Graphviz³ para su visualización. Después de realizar el análisis se definieron los casos para obtener relaciones entre dos entidades nombradas.

Los casos que se proponen consisten en identificar patrones para obtener un conjunto de relaciones predefinidas, casos basados en las relaciones de dependencia universal que ligan a dos entidades, y aquellos casos que toman elementos clave de la oración como son el *sujeto* y *objeto*.

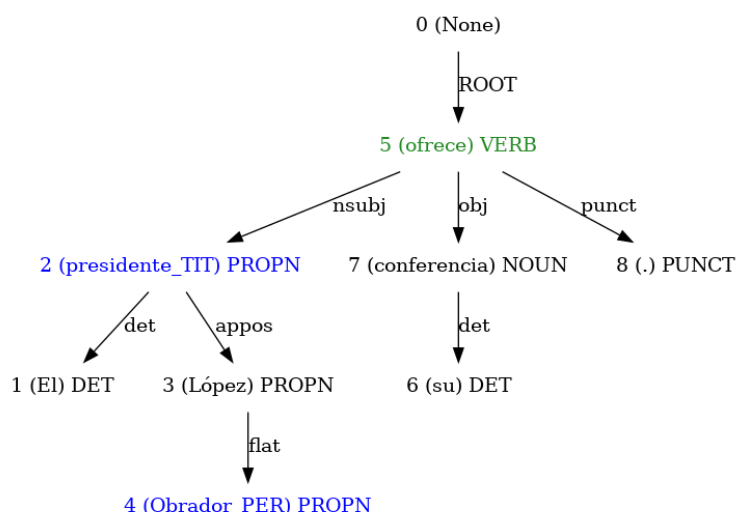


Figura 1. Árbol de dependencias en su forma gráfica.

¹ Disponible en: <https://spacy.io/models/es>

² Disponible en: <https://networkx.org/documentation/stable/>

³ Disponible en: <https://graphviz.readthedocs.io/en/stable/>

La oración “El presidente_TIT López Obrador_PER ofrece su conferencia.” genera un árbol de dependencias como se muestra en la Figura 1, donde los tokens (palabras o símbolos) representan a los nodos y las aristas (líneas con dirección) son las relaciones de dependencia universal. Cada nodo contiene *id* (token) POS. El *id* es también el orden en que aparece el token en la oración, y POS es la categoría gramatical (*Part-of-Speech*) del token.

El proceso consiste en transformar la oración obtenida del modelo de análisis de dependencias a un grafo, y se define como $G = (V, E, \theta)$ donde V son los vértices o nodos, E son las aristas y $\theta : E \rightarrow \{\{x, d, y\} : x, y \in V \wedge d \in D = \{dependencias\ universales\}\}$, es decir, se asigna una arista d a un par de nodos. Los vértices x e y contienen la estructura (*id*, POS, token). El conjunto de “dependencias universales” se define como $D = \{ROOT, acl, advcl, advmod, amod, appos, aux, case, cc, ccomp, compound, conj, cop, csubj, dep, det, expl, pass, fixed, flat, iobj, mark, nmod, nsubj, nummod, obj, obl, parataxis, punct, xcomp\}$. Y al conjunto que contiene a las etiquetas POS como $S = \{ADJ, ADP, ADV, AUX, CONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB\}$ además se establece al conjunto $O = \{Tokens\ de\ la\ oración\}$. Estos conjuntos son empleados en el grafo. El conjunto O es el único que cambia dependiendo la oración que se esté procesando. En cada uno de los métodos propuestos como paso inicial se busca un camino simple entre la primera entidad (e_1) y la segunda entidad (e_2), donde $e_1, e_2 \in V$, exceptuando los métodos donde se identifica al *sujeto* y *objeto* de la oración.

Relaciones basadas en patrones. En este método se definieron cuatro tipos de relaciones de forma manual. La primera relación que se identifica es del tipo “*puestos de trabajo*”. Para definir los patrones se usan las clases de las entidades nombradas: *ORG*, *PER*, *PEX* y *TIT*. Los patrones se definen en pares usando la clase de ambas entidades (e_1, e_2), el conjunto de patrones definido es $P = \{(ORG, PER), (ORG, TIT), (PER, ORG), (PER, PEX), (PER, TIT), (PEX, PER), (PEX, TIT), (TIT, ORG), (TIT, PER), (TIT, PEX)\}$. Para representar la relación de dependencia de dos nodos se establece como $nodo_1 \xrightarrow{r} nodo_2 | r \in D$. Las condiciones que se deben cumplir para identificar este tipo de relaciones se establecen como: $R_1 = \{e_1 \xrightarrow{d} e_2 \wedge (e_2 \xrightarrow{punct} x \vee e_2 \xrightarrow{none} \emptyset) | d \in D, (e_1, e_2) \in P\}$. De forma gráfica la Figura 2 ilustra como e_2 debe ser descendiente directo de e_1 (no deben existir nodos entre ellas). Además, se debe cumplir que el nodo $x = (id, PUNCT, t) | t \in T, T = \{",", ";", "(", ")"\}$. Donde e_2 puede tener una dependencia *punct* hacia un nodo con una etiqueta POS de tipo *PUNCT*, o bien e_2 puede no tener descendientes.

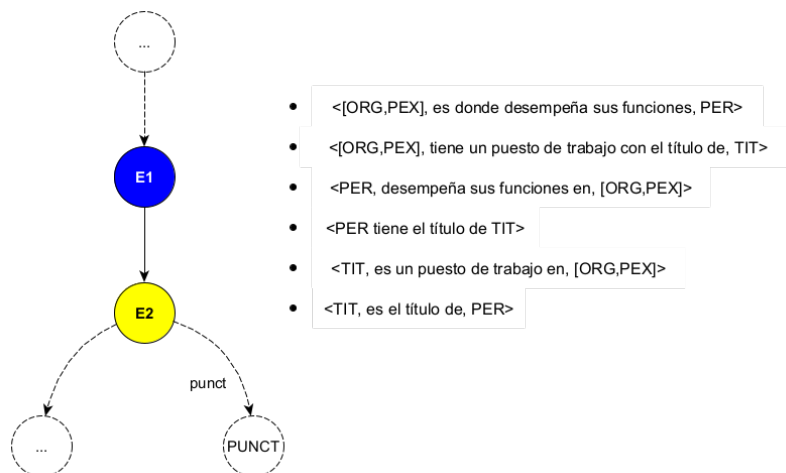


Figura 2. Estructura de las relaciones sobre puestos de trabajo.

El conjunto de relaciones que se pueden identificar sobre *puestos de trabajo* son las que se observan en la Figura 2, siempre y cuando se cumpla la clase de la entidad nombrada en cuestión. Por ejemplo, en $\langle [ORG, PEX], es\ donde\ desempeña\ sus\ funciones, PER \rangle$ se puede cumplir que e_1 sea una organización (*ORG*) o un partido político (*PEX*), y e_2 debe de ser una entidad de tipo persona (*PER*) además se deben cumplir los requisitos de R_1 para poder extraer la relación en forma de tripleta.

La segunda relación para identificar es de tipo “*tiene el acrónimo de*” se redefine el conjunto P . El conjunto de patrones $P = \{(PER, PER), (ORG, ORG), (TIT, TIT), (GPE, GPE), (PEX, PEX),$

$(FAC, FAC), (EVT, EVT), (DOC, DOC), (PRO, PRO), (LOC, LOC)$. Para obtener este tipo de relaciones se debe cumplir que $R_2 = \{e_1 \xrightarrow{d} e_2 \wedge (e_2 \xrightarrow{punct} s_a \wedge e_2 \xrightarrow{punct} s_b) | (e_1, e_2) \in P\}$. Donde el vértice $s_a = (id, PUNCT, "(")$ y el vértice $s_b = (id, PUNCT, ")")$. En la [Figura 3](#) se observa que e_2 sea descendiente directo de e_1 , y que e_2 debe tener dos descendientes específicos (paréntesis de apertura y cierre).

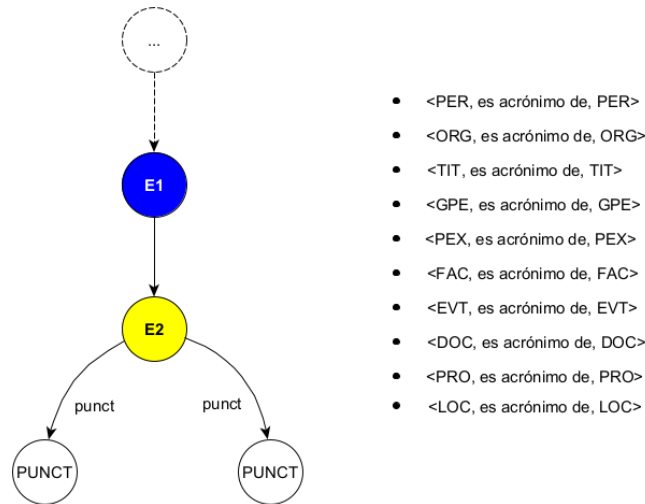


Figura 3. Estructura para identificar la relación acrónimo.

La relación de acrónimo puede ser identificada y extraída de las combinaciones de entidades mostradas en la [Figura 3](#).

La tercera relación es del tipo “*que pertenece a*”, el conjunto de patrones se redefine como $P = (FAC, GPE), (ORG, GPE)$, el conjunto $D_a = \{flat, nmod, appos\}$, $D_b = \{case, det\}$, y $S_a = \{ADP, DET\}$. Dados los conjuntos se tiene que $R_3 = \{e_1 \xrightarrow{r_a} e_2 \wedge ((e_2 \xrightarrow{punct} s_a \wedge e_2 \xrightarrow{punct} s_b) \vee e_2 \xrightarrow{r_b} s) | r_a \in D_a, r_b \in D_b, (e_1, e_2) \in P\}$. Donde el vértice $s_a = (id, PUNCT, "(")$ y el vértice $s_b = (id, PUNCT, ")")$, y se tiene que $s = (id, ADP, token) \vee (id, DET, token)$. En la [Figura 4](#) se puede observar cómo e_1 debe ser ancestro directo de e_2 , a su vez e_2 puede tener descendientes con etiqueta *PUNCT* (signo de puntuación) paréntesis de apertura y cierre o tener descendientes con etiquetas *ADP* (Adposición) y *DET* (Determinante).

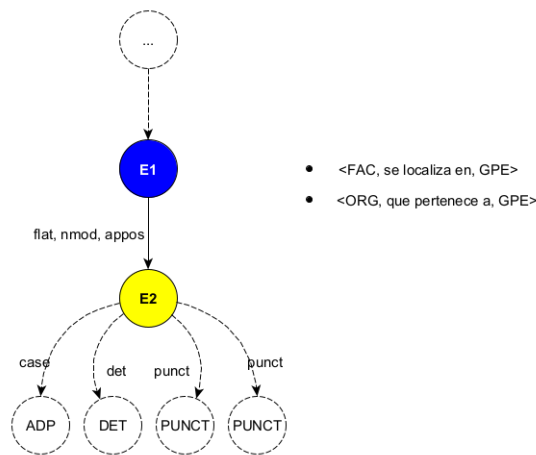


Figura 4. Estructura para identificar relaciones

Las relaciones que se pueden extraer de la estructura de la [Figura 4](#), son relaciones que indican en que ciudad se ubica una instalación, así como organizaciones que pertenecen a un estado o ciudad.

La cuarta relación definida de forma manual en base a sus patrones es del tipo “representado por”. El conjunto de patrones se redefine como $P = \{(GPE, PER)\}$ y $D_a = \{flat, appos\}$. Para identificar esta regla se debe cumplir que $R_4 = \{e_1 \xrightarrow{ra} e_2 \wedge e_2 \xrightarrow{punct} s | r_a \in D_a, (e_1, e_2) \in P\}$, donde el vértice $s = (id, PUNCT, ", ")$. La [Figura 5](#) representa las condiciones para identificar esta relación. La entidad e_1 debe ser ancestro de e_2 y debe tener alguna de las dependencias *flat* o *appos*. Así como e_2 debe tener una dependencia de tipo *punct* hacia un nodo con un token “,”. Cuando se cumplen las condiciones en R_1, R_2, R_3 y R_4 se define la tripleta $\langle e_1, relación, e_2 \rangle$ resultante y se almacena.

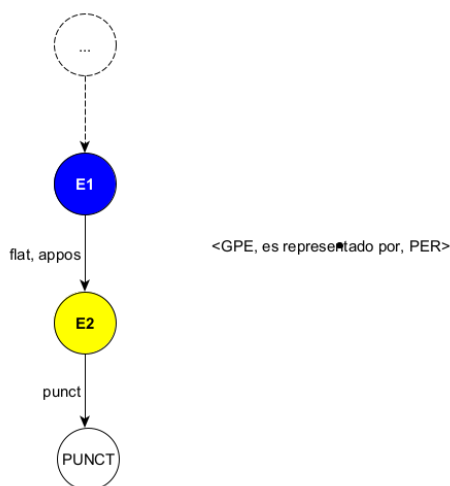


Figura 5. Estructura para identificar relaciones *es representado por*.

La única relación definida en la [Figura 5](#), se refiere a los estados, ciudades que son representados por una persona.

Relaciones extraídas automáticamente. Para identificar y extraer relaciones de forma automática se emplea las dependencias universales entre tokens (nodos), así como localizar el *verbo* entre ambas entidades, e identificar elementos clave como *sujeto*, y *objeto* de la oración enlazados al *predicado* (*verbo*). Todos los nodos tienen la estructura (*id, POS, token*) aunque solo se refiera a ellos por su etiqueta POS.

En la [Figura 6](#) se ilustran los métodos en su forma básica para identificar y extraer relaciones entre entidades nombradas. Para identificar relaciones entre entidades usando la dependencia “*appos*” (modificador de sustantivo) como se observa en la [Figura 6](#), se debe cumplir en su forma más básica que $R_5 = \{e_1 \xrightarrow{d} NOUN \wedge NOUN \xrightarrow{appos} e_2 | d \in D\}$, esto indica que entre los vértices e_1 y e_2 exista un nodo *NOUN*, y además que contenga una dependencia *appos* de e_1 hacia el nodo e_2 . Por ejemplo, si se tiene que $R_{ex} = \{(e_1 \xrightarrow{d} ADV) \wedge (ADV \xrightarrow{nmod} NOUN) \wedge (NOUN \xrightarrow{appos} X) \wedge (X \xrightarrow{d} e_2) | \exists d \in D\}$ donde *ADV* y *NOUN* son etiquetas *POS* específicas de nodos, y *X* es la etiqueta *POS* de cualquier vértice, así se establecen las condiciones que se deben cumplir para identificar la relación R_{ex} , y los nodos almacenados solo serán aquellos que sean específicamente definidos, es decir el nodo con etiqueta *POS X* será descartado. Para extraer la relación se obtiene el conjunto de los vértices obtenidos, del ejemplo anterior se tiene $R_{ex} = \{(id_1, ADV, token_1), (id_2, NOUN, token_2)\}$ y de cada vértice se obtiene una lista $r = [token_1, token_2]$ de los tokens en los vértices seleccionados, además se ordenan en forma ascendente en base a su *id*. Cada uno de los elementos de la lista r conforman la tripleta $\langle e_1, r, e_2 \rangle$ y se almacena en la base de datos. De forma opcional, si se seleccionan los nodos descendientes de *NOUN* (nodo en color rojo de la [Figura 6](#)) se deben cumplir las dependencias descritas en forma gráfica de la [Figura 7](#), creando diferentes combinaciones para identificar relaciones de forma automática.

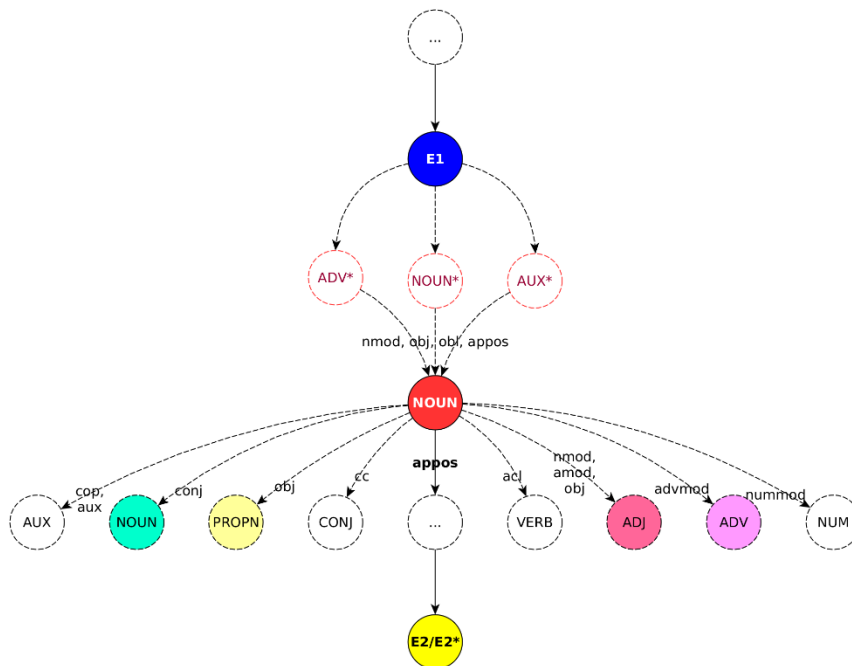


Figura 6. Estructura básica para identificar relaciones de dependencia *appos*.

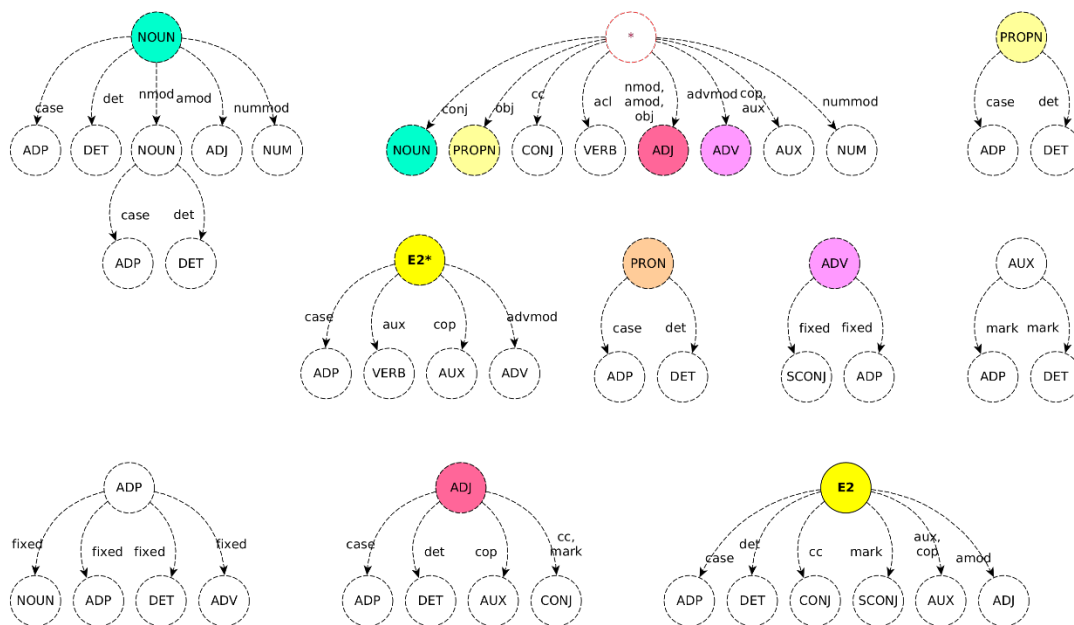


Figura 7. Estructura complemento para identificar relaciones usando *appos*.

El caso donde se identifican relaciones usando la dependencia universal “*amod*” (modificador de adjetivo) se define como $R_6 = \{(e_1 \xrightarrow{d} ADJ) \wedge (ADJ \xrightarrow{amod} e_2) | d \in D\}$ en su forma más básica, como se observa en la Figura 8.

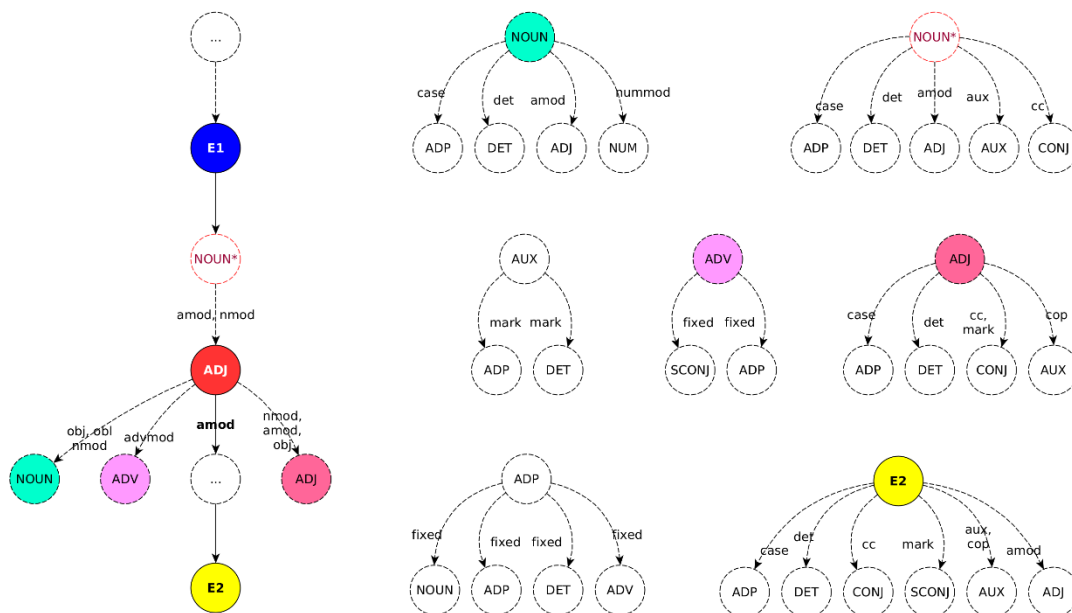


Figura 8. Estructura para identificar relaciones usando *amod*.

El proceso inicia en el nivel más alto, es decir parte de e_1 hacia e_2 . Sin embargo, pueden ocurrir diferentes combinaciones para identificar este tipo de relaciones las cuales deben seguir la estructura mostrada en la Figura 8.

Otro caso analizado es identificar al menos un “*verbo entre las dos entidades*” empleando la etiqueta POS (nodo) de los tokens, y se define como $R_7 = \{(e_1 \xrightarrow{d} VERB) \wedge (VERB \xrightarrow{d} e_2) | d \in D\}$. La Figura 9 describe la estructura en forma gráfica para identificar este tipo de relaciones. Cuando la oración presenta más de un vértice *VERB* entre ambas entidades, se selecciona el primer vértice siguiendo la ruta $e_2 \rightarrow e_1$. En caso de existir un vértice *VERB* como ancestro del vértice *VERB* seleccionado previamente, se tiene que cumplir que ambos vértices estén ligados por alguna de las dependencias mostradas en la Figura 9. También puede ocurrir que el ancestro *VERB* del nodo principal presente más opciones que pueden aportar información para formar la relación. Para ello se presentan las diferentes opciones que pueden tomar los nodos seleccionados, como se muestra en la Figura 10.

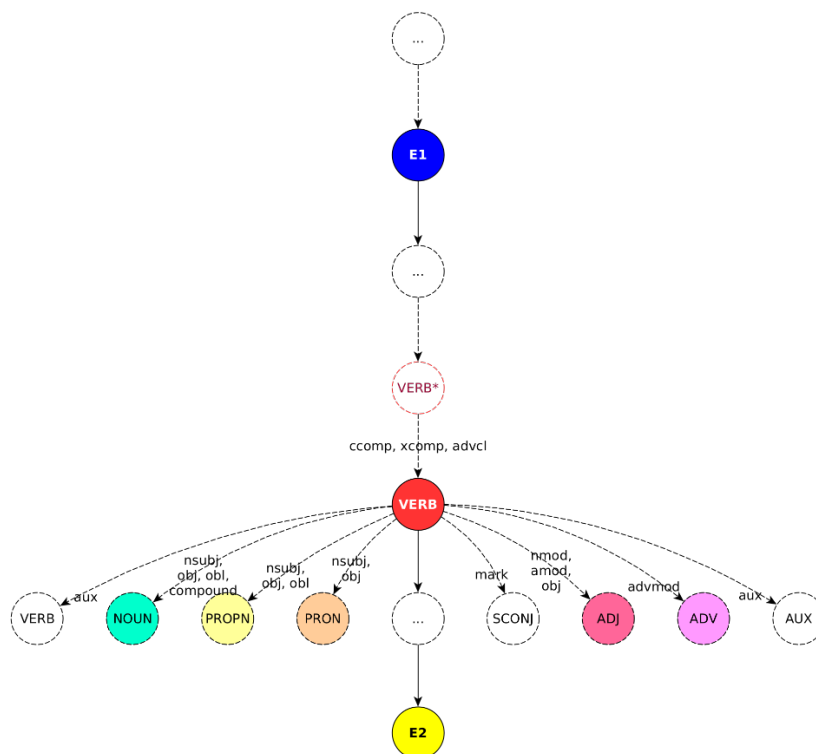


Figura 9. Estructura principal para identificar relaciones con un verbo entre entidades.

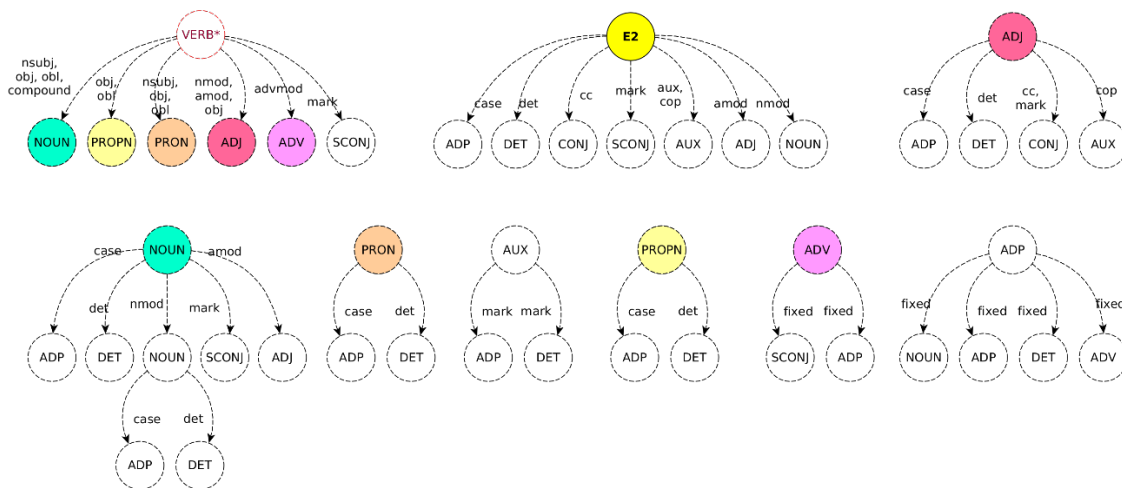


Figura 10. Estructura complemento para identificar relaciones con un verbo entre entidades.

Dos casos más fueron analizados. En el primer caso se identifica al *sujeto* (e_1) y al *objeto* (e_2) y el vértice principal será el *predicado* que es un vértice *VERB*. A partir de este nodo se identifican las dependencias universales para el *sujeto* y *objeto* como se describe en la Figura 11. La definición está dada por $R_8 = \{(VERB \xrightarrow{nsubj} e_1) \wedge (VERB \xrightarrow{obj,objl} e_2)\}$. Como se observa pueden existir cualesquiera vértices entre el vértice *VERB* y cualquiera de las dos entidades.

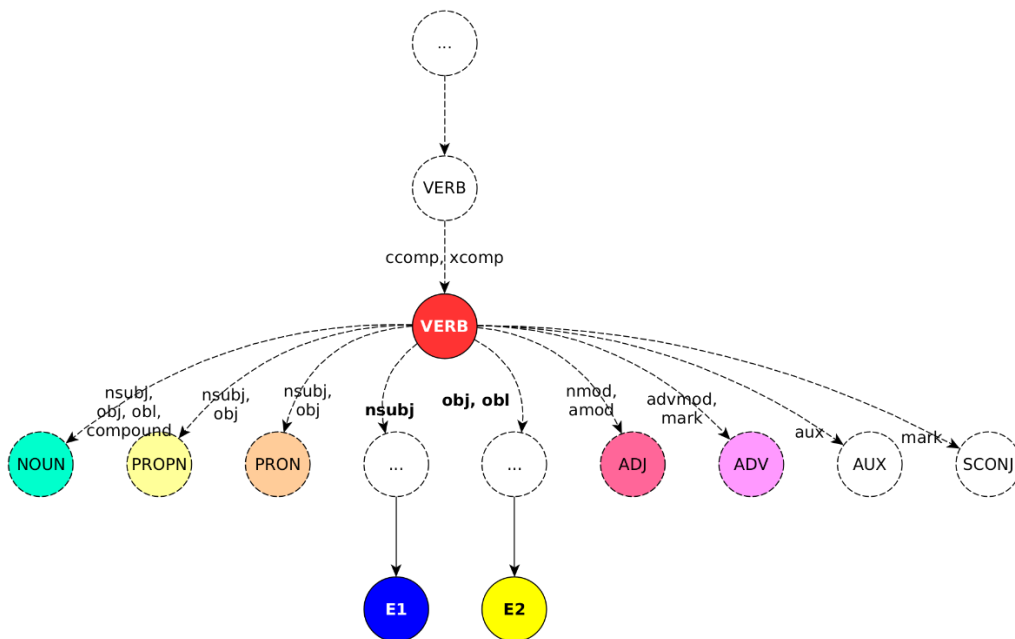


Figura 11. Estructura principal para identificar relaciones con *sujeto* (e_1) y *objeto* (e_2).

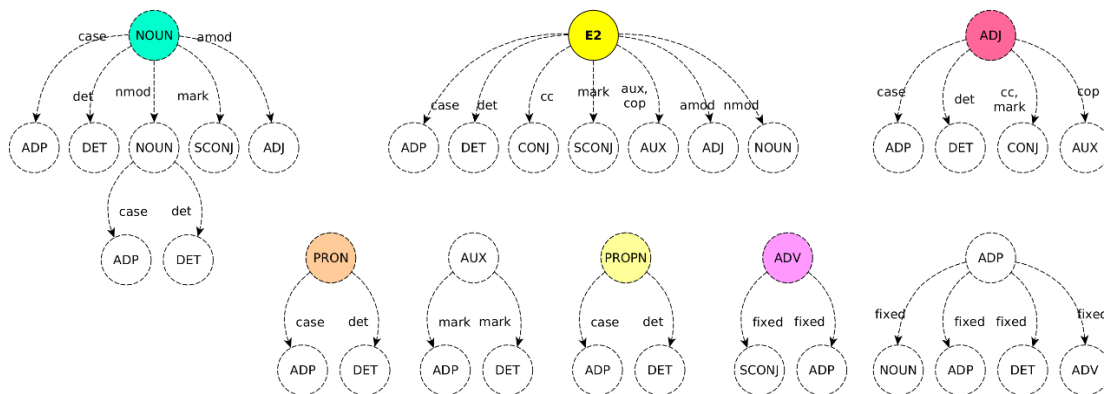


Figura 12. Estructura complemento para identificar relaciones con *sujeto* (e_1) y *objeto* (e_2).

Como en los casos anteriores, es posible la selección de diferentes vértices a partir del nodo objetivo (*VERB*), los cuales deben cumplir con las dependencias y tipo de etiqueta *POS*, como se detalla en la Figura 12.

El siguiente caso se encuentra basado en la misma premisa que el caso anterior. La diferencia radica en la posición en la que se encuentran las entidades. Es decir, el nodo objetivo es el *predicado* vértice *VERB* y de este debe existir una relación de dependencia *sujeto* hacia e_2 y una dependencia *objeto* hacia e_1 , como se ilustra en la Figura 13. La definición es $R_9 = \{(VERB \xrightarrow{nsubj} e_2) \wedge (VERB \xrightarrow{obj,obl} e_1)\}$.

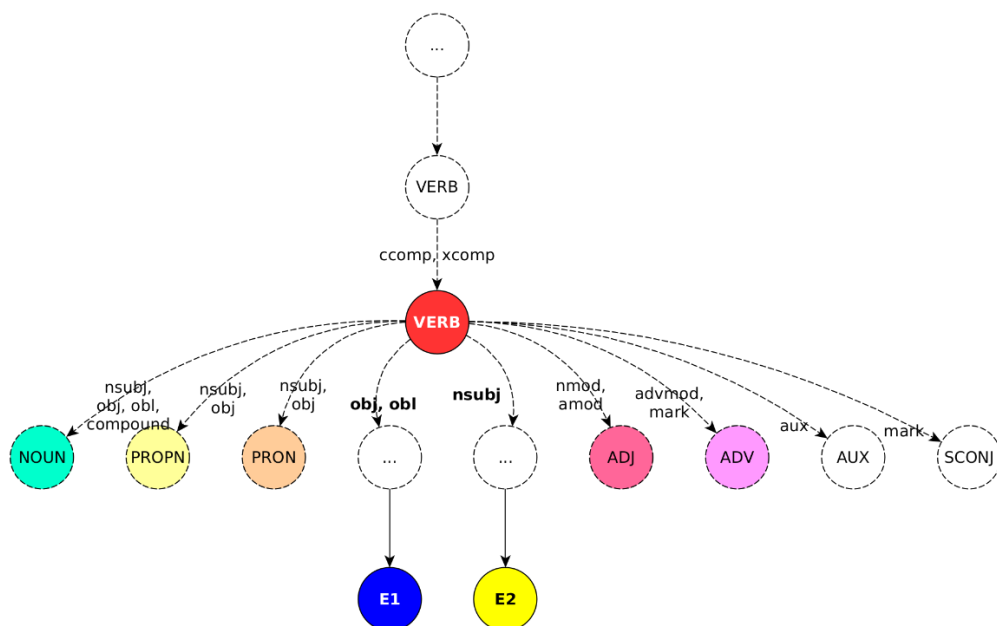


Figura 13. Estructura principal para identificar relaciones con *sujeto* (e_2) y *objeto* (e_1).

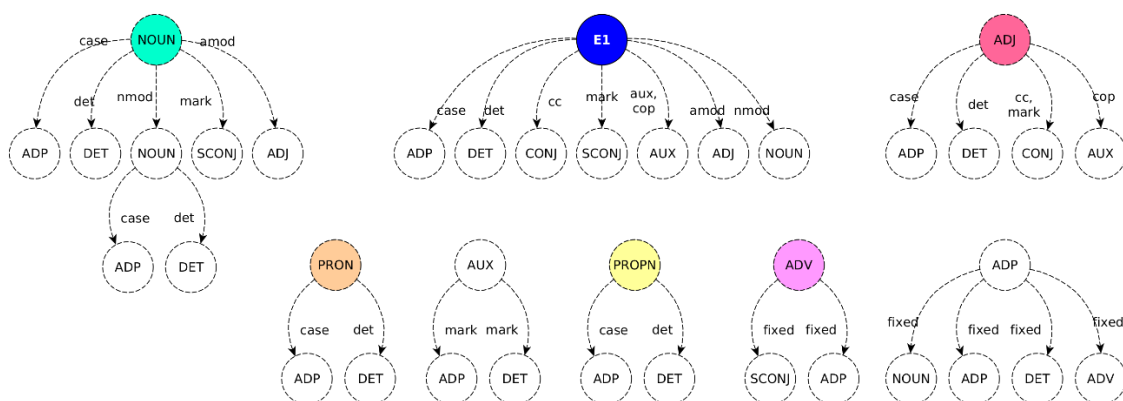


Figura 14. Estructura complemento para identificar relaciones con *sujeto* (e_2) y *objeto* (e_1).

En este último caso se busca identificar la relación que existe entre dos entidades, asumiendo que e_2 será la primera entidad en la tripleta resultante y e_1 será la segunda, por ello se ha intercambiado el orden de las relaciones de dependencia del nodo principal hacia las entidades. Bajo estas condiciones se estableció otro cambio. No se buscará identificar nodos descendientes en e_2 , ahora esta tarea se asigna a e_1 como se observa en la Figura 14.

En estos casos analizados se tienen que cumplir cada una de las estructuras planteadas para la identificación y extracción automática de relaciones.

5 Resultados

La Tabla 2 muestra el top 10 de las relaciones identificadas y extraídas. La columna FR representa: Frecuencia de Relaciones extraídas. FO: Frecuencia de Oraciones donde se identificó a la relación. FD: Número de Documentos cuyas oraciones se emplearon para identificar y extraer relaciones. La columna C: Caso usado para extraer la relación; 1: *Puestos de Trabajo*, 2: relaciones donde se identificó al *sujeto* (e_1) y *objeto* (e_2). De las 175.754 relaciones obtenidas se obtuvo un conjunto de datos con 86.917 relaciones únicas ordenadas en forma descendente iniciando con la relación más frecuente.

Tabla 2. Top 10 de las relaciones identificadas y extraídas.

#	FR	FO	FD	C	Entidad 1	Relación	Entidad 2
1	6,437	4,977	4,201	1	Presidente	<i>Es el título de</i>	Andrés Manuel López Obrador
2	3,311	332	1,667	1	Gobernador	<i>Es el título de</i>	Adán Augusto López Hernández
3	2,871	20	2,864	1	Diputada	<i>Es el título de</i>	Martha Tagle
4	2,567	1	2,567	2	Sinaloa	<i>El municipio podría recibir una nueva visita de</i>	AMLO
5	2,557	1	2,557	2	UNAM	<i>Expertos resolverán preguntas a partir de las</i>	15:00 horas
6	1,450	8	1,448	1	Secretaría de Cultura	<i>Es el título de</i>	Yolanda Osuna Huerta
7	1,443	1	1,443	2	Gobierno del Estado	<i>Adelantó que realiza las gestiones la</i>	Biblioteca Pino Suárez
8	1,442	1	1,442	2	Sistema DIF Tabasco	<i>Sostuvo que con este acto contribuyen al crecimiento de la</i>	Biblioteca Pino Suárez
9	1,442	1	1,442	2	Gobernador	<i>Entregó una en el poblado de</i>	Comalcalco
10	1,442	1	1,442	1	Director general de la Red Estatal de Bibliotecas	<i>Es el título de</i>	Ariel Gutiérrez Valencia

Además, se realizó una evaluación manual seleccionando una cantidad específica de relaciones extraídas. Para ello se realizaron dos experimentos. En el primer experimento se ordenaron las relaciones en forma descendente en base a su frecuencia del corpus de tripletas obtenido, y se seleccionaron las primeras 200 relaciones por cada caso. En el segundo experimento, se tomó un corpus de 300 documentos del corpus de entidades nombradas sin procesar de forma aleatoria, para después aplicar el método propuesto e identificar y extraer relaciones en este subconjunto de 300 documentos anotados.

La evaluación se realizó usando un sistema web, donde se recuperan las relaciones (tripletas) almacenadas en el base de datos. Se despliega una tripla a la vez y se visualizan a lo más diez oraciones en las que ocurre la tripla. Lo que permite al evaluador calificar como correcta o incorrecta cada una de las partes que componen la tripla (e_1 , *relación*, e_2). Las oraciones desplegadas sirven como referencia al evaluador para corroborar que la relación fue identificada y definida de forma correcta. La evaluación consistió en evaluar de forma *positiva* o *negativa* a cada elemento de la tripla: e_1 , *relación* y e_2 , y una segunda evaluación únicamente para la *relación* de la tripla sin importar las entidades.

5.1 Experimento 1

Para este experimento de las 86.917 relaciones extraídas se tomaron 200 tripletas por cada caso, con excepción del caso (4), donde se tomaron las 33 disponibles. Una descripción detallada de la evaluación se describe en la [Tabla 3](#) para cada uno de los casos usados. La columna *Total* enumera la cantidad neta de relaciones extraídas por caso. Como se ha mencionado previamente para la columna *Eval* (Evaluación) en este experimento se tomaron 200 relaciones (tripletas), siendo las primeras 200 tripletas, esto después de ordenarlas en base a su frecuencia en forma descendente. La columna *CR* (Conteo de Relaciones) hace referencia al número de relaciones evaluadas de forma correcta, sin tomar en cuenta la evaluación de las entidades nombradas. En el mismo sentido la columna *CT* (Conteo Tripletas) enumera la cantidad de tripletas evaluadas correctamente, esto incluye a ambas entidades y la relación. Las columnas *PR* (Porcentaje Relación) y *PT* (Porcentaje Tripletas) describen los porcentajes de evaluación de cada uno de los métodos.

Tabla 3. Evaluación de relaciones para el experimento 1.

#	Caso	Total	Eval	CR	CT	PR	PT
1	Puestos de trabajo	17,073	200	193	193	96.5%	96.5%
2	Acrónimos	2,367	200	199	192	99.5%	96.0%
3	Se localiza en	343	200	184	172	92.0%	86.0%
4	Representado por	33	32	28	25	87.5%	78.1%
5	Dependencia <i>appos</i>	1,408	200	156	132	78.0%	66.0%
6	Dependencia <i>amod</i>	1,922	200	161	131	80.5%	65.5%
7	Sujeto e_1 Objeto e_2	44,479	200	126	111	63.0%	55.5%
8	Verbo entre entidades	7,896	200	124	102	62.0%	51.0%
9	Sujeto e_2 Objeto e_1	11,396	200	109	93	54.5%	46.5%
		86,917	1,632	1,280	1,151	78.4%	70.5%

5.2 Experimento 2

Para este experimento se seleccionaron aleatoriamente 300 documentos del corpus original de entidades nombradas, para posteriormente identificar y extraer relaciones. En este experimento del caso “Representado por” no se extrajo ninguna relación, por lo que no es presentado en los resultados. Por otro lado, se observa que en el caso “Se localiza en” (ambas columnas) y el caso “Dependencia *amod*” (primera columna) presentan el 100%. Sin embargo, las relaciones y tripletas evaluadas no son significativas, como se describe en la [Tabla 4](#).

Los métodos con el mayor número de tripletas evaluadas son “Sujeto e_1 Objeto e_2 ”, “Puestos de trabajo” y “Sujeto e_2 Objeto e_1 ” de la [Tabla 4](#) con un 55.25%, 91.46% y 56.31% respectivamente. Por otro lado, el método “Se localiza en” con muy pocas tripletas evaluadas presenta un 100%.

Tabla 4. Evaluación de relaciones para el experimento 2.

#	Caso	Total	Eval	CR	CT	PR	PT
1	Se localiza en	4	4	4	4	100.0%	100.0%
2	Acrónimo	80	80	77	75	96.25%	93.75%
3	Puestos de trabajo	334	328	304	300	92.68%	91.46%
4	Dependencia <i>amod</i>	16	13	13	8	100.0%	61.54%
5	Verbo entre entidades	76	70	50	43	71.43%	61.43%
6	Sujeto e_2 Objeto e_1	115	103	63	58	61.17%	56.31%
7	Sujeto e_1 Objeto e_2	435	362	252	200	69.61%	55.25%
8	Dependencia <i>appos</i>	18	16	11	8	68.75%	50.00%
		1,078	976	774	696	79.30%	71.31%

6 Conclusiones

En este trabajo se presentó un enfoque no supervisado para extraer relaciones entre dos entidades empleando árboles de dependencias. Para este fin se definió un conjunto de patrones, dependencias universales y verbos entre entidades, e identificando las dependencias de *sujeto*, *predicado*, y *objeto*. En los casos donde se definieron patrones también se establecieron las relaciones a identificar de forma manual, en cambio en los otros casos se identificaron y extrajeron relaciones de forma automática. En los resultados se observa que las relaciones con alta frecuencia son las que se definieron de forma manual. Sin embargo, se identificó un número mayor de relaciones con frases verbales, así como relaciones que indican parentesco.

La ventaja de emplear este método para identificar y extraer relaciones en dos entidades nombradas consiste en los árboles (grafos) de dependencia en sí, representando la oración en estos y permitiendo emplear los algoritmos básicos ya conocidos en la teoría de grafos para buscar el camino simple entre dos nodos, el nodo ancestro de dos nodos, y obtener los descendientes de un nodo específico. Cuando se conoce la relación a extraer (habiendo analizado los “patrones” de esta previamente), el proceso para identificar y extraer la relación definida se lleva a cabo de una forma ágil y sencilla, apoyándose de las formas

gramaticales presentes en las dependencias universales, así como de la etiqueta *POS* para restringir el espectro de búsqueda en la identificación de la relación.

La desventaja de este método recae en el análisis que se tiene que llevar a cabo para identificar relaciones potenciales. Analizar diversas oraciones y observar posibles “*patrones*” en los que ocurre una relación, esto con relaciones definidas de forma manual. Para el caso de extraer relaciones de forma automática se tienen que realizar aún más experimentos y análisis, además de analizar los resultados de los experimentos para *afinar* las relaciones a obtener, aplicando restricciones sobre ciertas dependencias, las entidades nombradas involucradas y tratar de *seleccionar* los tokens que den coherencia a la relación. Otra desventaja se observa en el conjunto de datos (entidades nombradas). Al estar desbalanceado se identifican y extraen relaciones con entidades nombradas más frecuentes.

Como trabajo a futuro, se planea analizar la informatividad presentada por las tripletas extraídas, evaluando su exactitud, en base a si presentan información crítica o se omite. Además, observar la información que aporta la relación de la tripleta, si es coherente, y se encuentra en un contexto adecuado a la oración de la que fue extraída. Revisar el algoritmo para identificar relaciones usando las dependencias universales, contemplar la negación como parte de una relación. Se pretende aplicar este enfoque a un conjunto de datos del estado del arte, para ello se deberá realizar un ajuste al algoritmo para adaptarse al idioma. O bien realizar la traducción del conjunto de datos al idioma español, con el objetivo de observar el rendimiento del enfoque propuesto.

Declaración de conflicto de intereses

Los autores declaran no tener conflicto de intereses con respecto a la investigación, autoría o publicación de este artículo.

Financiación

Los autores no recibieron apoyo financiero para la investigación, autoría y/o publicación de este artículo.

ORCID iD

Orlando Ramos-Flores  <https://orcid.org/0000-0002-8579-4123>

David Pinto  <https://orcid.org/0000-0002-8516-5925>

Referencias

- Afzal, N., Mitkov, R., & Farzindar, A. (2011). Unsupervised Relation Extraction Using Dependency Trees for Automatic Generation of Multiple-Choice Questions. In C. Butz & P. Lingras (Eds.), *Advances in Artificial Intelligence. Canadian AI 2011. Lecture Notes in Computer Science* (Vol. 6657). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21043-3_4
- Bunescu, R., & Mooney, R. (2005). Bunescu, Razvan, and Raymond Mooney. "A shortest path dependency kernel for relation extraction. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 724–731.
- Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12). <https://doi.org/10.1145/1409360.1409378>
- Fundel, K., Kuffner, R., & Zimmer, R. (2007). RelEx--Relation extraction using dependency parse trees. *Bioinformatics*, 23(3). <https://doi.org/10.1093/bioinformatics/btl616>
- Hasegawa, T., Sekine, S., & Grishman, R. (2004). Discovering relations among named entities from large corpora. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 415–422.
- Li, J., Luong, M.-T., Jurafsky, D., & Hovy, E. (2015). When Are Tree Structures Necessary for Deep Learning of Representations? *ArXiv Preprint*.

- Miwa, M., & Bansal, M. (2016). End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. *ArXiv Preprint*.
- Quan, C., Wang, M., & Ren, F. (2014). An Unsupervised Text Mining Method for Relation Extraction from Biomedical Literature. *PLoS ONE*, 9(7). <https://doi.org/10.1371/journal.pone.0102039>
- Vo, D.-T., & Bagheri, E. (2017). Open information extraction. In *World Scientific Encyclopedia with Semantic Computing and Robotic Intelligence | Semantic Computing*. World Scientific Publishing Co Pte Ltd. https://doi.org/10.1142/9789813227927_0001
- Wu, Y., Zhang, Q., Huang, X., & Wu, L. (2009). Phrase dependency parsing for opinion mining. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 1533–1541.
- Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., & Ishizuka, M. (2004). Unsupervised relation extraction by mining wikipedia texts using information from the web. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1021–1029.