Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Ingeniería del Lenguaje y del Conocimiento

Adquisición automática de hechos utilizando inferencia para la construcción de un grafo de conocimiento

presenta

Orlando Ramos Flores

en cumplimiento parcial de los requisitos

para obtener el grado académico de

Doctor en Ingeniería del Lenguaje y del Conocimiento

bajo la supervisión de

Dr. David Eduardo Pinto Avendaño

Benemérita Universidad Autónoma de Puebla

Dr. Manuel Montes y Gómez

Instituto Nacional de Astrofísica, Óptica y Electrónica

Puebla, Pue. Junio 2021

Dedico este trabajo de tesis a mi familia.

Agradecimientos

Agradezco el apoyo, guía y consejo de los directores de mi tesis: Dr. David Pinto Avendaño y Dr. Manuel Montes y Gómez.

También agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo brindado para la obtención de este grado.

Resumen

La adquisición de conocimiento empleando técnicas de Procesamiento de Lenguaje Natural (PLN) es una tarea que ha sido estudiada a través de los años, y continúa siendo relevante en la actualidad, debido a la gran cantidad de información contenida principalmente en documentos no estructurados. Aplicar técnicas basadas en la Extracción de Información (EI) conlleva a automatizar el proceso y en gran medida a ahorrar recursos para obtener conocimiento de textos no estructurados. En este trabajo se presenta un enfoque para obtener hechos de documentos no estructurados, siguiendo un flujo de trabajo que inicio con la definición del tipo de información que se deseaba conocer (entidades nombradas), la forma en que se relaciona (extracción de relaciones), usando inferencia lógica sobre esta información para obtener nuevos hechos, y finalmente almacenar los hechos obtenidos en una estructura (grafo de conocimiento).

En primer instancia, un conjunto de diecisiete entidades nombradas fue definido ampliando las clases usadas en trabajos del estado del arte. El conjunto de datos consiste de documentos en idioma español. Se etiquetaron doscientos cincuenta documentos, empleando el esquema Inside, Output, Begining, End, Single (IOBES) e IOB. Un modelo Named Entity Recognition (NER) fue desarrollado para identificar y reconocer a las entidades nombradas definidas. Se realizaron diferentes experimentos con el algoritmo Conditional Random Fields (CRF) y con redes neuronales con arquitectura Bidirectional Long Short-Term Memory (Bi-LSTM) incluyendo *Embeddings from Language Models* (ELMo) en idioma español. Además se realizaron experimentos con el conjunto de datos CoLNLL-2002 y con el conjunto de noticias etiquetado previamente, ambos conjuntos de datos bajo el esquema IOB.

En segundo lugar, un enfoque usando árboles de dependencia fue propuesto para identificar casos que permitan hacer frente a la extracción de relaciones entre dos entidades nombradas. Se analizaron casos basados en patrones, y los casos usando las *dependencias universales* y la etiqueta *Part-of-Speech* (POS). En los primeros casos se establecieron de forma manual las relaciones a identificar, y en los segundos casos se obtuvieron de forma automática. Los casos se aplicaron a oraciones de los documentos con dos entidades nombradas, la información

derivada de la oración que incluye las dos entidades es la única relevante para la extracción de la relación, y la relación es independiente del texto que precede o sigue a la oración. La estructura para las relaciones obtenidas es una tripleta (*entidad1*, *relación*, *entidad2*), generando conjunto de datos de tripletas, y se almacenaron en una base de datos para su posterior análisis. La evaluación de las relaciones fue de forma manual sobre dos conjuntos parciales de tripletas. El primer conjunto se tomó del corpus principal, y para el segundo se seleccionaron documentos de forma aleatoria y se extrajeron sus relaciones. La evaluación consistió en asignar un valor de verdadero o falso. Se evaluó la tripleta completa y a cada parte de la misma. En la primera evaluación, todos los elementos de la tripleta tienen que ser verdaderos para asignarle un valor de verdadero, y falso en caso contrario. En la segunda evaluación, el valor de la relación es el único contemplado sin importar a los otros elementos de la tripleta.

En tercer lugar, el conjunto de tripletas fue "*traducido*" a una base de hechos con sintaxis basada en Prolog. En la base de hechos un conjunto de *reglas lógicas genéricas* fue definido, es decir; reglas que al ser inferidas permitan obtener *nuevo conocimiento* sin contemplar de antemano un resultado esperado. En los experimentos se tomaron los dos conjuntos de datos evaluados en la etapa anterior y la base de hechos completa. Se realizaron experimentos con cada una de las reglas definidas sobre las tres bases de hechos. Además se desarrolló un análisis de forma manual sobre algunos de los resultados obtenidos. El análisis tuvo el objetivo de observar los hechos obtenidos de una regla, se observó la oración de procedencia del hecho, así como el documento de donde procede dicha oración. En este proceso se asumió como *"descubrir nuevo conocimiento"* a la forma en que se relacionan los hechos obtenidos, primero siendo parte de diferentes documentos y oraciones, y la forma que tienen dichos hechos de proporcionar información y un contexto entre sí. Este proceso no podría obtenerse de forma trivial, sobretodo cuando se posee un gran volumen de documentos.

Finalmente, los hechos obtenidos en la inferencia lógica son transformados siguiendo los lineamientos de la W3C para construir un grafo de conocimiento. Estableciendo previamente los esquemas necesarios para definir cada uno de los hechos obtenidos. Donde las entidades nombradas se definen como *nodos* y las relaciones como *aristas*, las *aristas* tienen dirección, partiendo del nodo *sujeto* (entidad1) y tienen como objetivo al nodo *objeto* (entidad2), por lo que el grafo de conocimiento es dirigido. Con este flujo de trabajo propuesto se ha podido comprobar que es posible extraer conocimiento de documentos no estructurados, así como descubrir nuevos hechos al observar su procedencia y como en su conjunto aportan información nueva o adicional.

Abstract

Knowledge Acquisition using Natural Language Processing techniques is a task that has been studied for years, continuing been relevant nowadays due to the high amount of data contained in unstructured documents mainly. Applying techniques based on Information Extraction leads to automating the entire process and saving resources greatly to obtain knowledge in unstructured texts. In this work, an approach to get facts from unstructured documents is introduced, following a pipeline starting with the definition of the kind of information (Named Entity Recognition), how this is related (Relation Extraction), using logic inference on these data to get new facts, and finally, storing the collected facts in a structure (Knowledge Graph).

Firstly, seventeen named entities group was settled increasing the number of the classes used in the state-of-the-art. The dataset is made up of politician-news documents in the Spanish language. A Named Entities Recognition (NER) model was developed to identify and recognize the named entities set before. Also, the named entities tagged process was made semi-manually assist by a web system for two hundred fifty documents using the schema Inside, Outside, Begining, End, and Start (IOBES) aims a get better results. Several experiments were carried up with Conditional Random Fields (CRF) and the Bidirectional Long Short-Term Memory (BiLSTM) Neural Network architecture adding Embeddings from Language Models (ELMo) in the Spanish language. Moreover, several experiments were made to the CoNLL-2002 dataset and the tagged news dataset setting the IOB schema for both.

Secondly, an approach using dependency trees was proposed to identify cases that enable to cope with the relationships extraction between two named entities. The defined cases are those pattern-based and those that use universal dependency relationships and the dependency tree Part-of-Speech label. In the pattern-based cases, a relationship set was defined manually, and In the other cases, the relationship was got automatically. The cases were applied to sentences with two entities the sentence-derived information that includes the two entities is the only relevant to extract their relationship, and the before wards and afterward text is independent of the relationship. The structure for all the relationships collected is a triplet (first entity, relationship,

second entity), so a new triplet dataset is built. At the same timing, the sentence and their triplet are stored in a database for later analysis. The relationships assessment was carried out on two datasets previously chosen. From the main dataset, the first two hundred triplets sorted by their frequency in an ascendant way were taken for each case to form the first dataset to assess. The second dataset was formed by extracting triplets from three hundred documents chosen randomly. The assessment consisted of assign a true or false value. The full triplet and each part of it were evaluated. In the first assessment, all triplet items had to be true for a true assign value and false otherwise. In the second assessment, the relationship value was the focus no matter the other items inside the triplet.

Thirdly, the triplet dataset was "*translated*" to a fact-base with the Prolog syntax. Inside the fact-base, a generic logic rule group was defined, i.e. by inferring these rules allow getting *new knowledge* non-regarding beforehand an expected result. The two previous evaluated datasets were used as well as the full triplets dataset for experiments, making experiments for each rule settled and for each fact-base. Manually was developed an analysis about some got results, aiming to observe the got specific rule facts. The fact's sentence source was observed as well as the document's sentence source. In this process, "new knowledge discover" was assumed in how obtained results are related themselves, first, belonging to different sentences and documents, and secondly, how they can provide a context and information to each other. This process could not be achieved trivially, especially on a gigantic document set.

Finally, facts got in the logic inference are transformed into a structure following the W3C standards aim to build a knowledge graph (KG). Previously, a schemas group was set to define each fact got. In the KG, named entities are defined as nodes and the relationships as edges. Edges have direction, starting from the subject (first entity) node toward the object node (second entity), so the graph is directed. With the pipeline proposed above is possible to extract knowledge from unstructured documents as well as new knowledge discover observing their source and how as a whole provide new or additional information.

Contenido

Lista de Figuras

Lista de tablas

Nomenclatura

Acrónimos / Abreviaciones

Bi-LSTM Bidirectional Long Short-Term Memory

CNN Convolutional Neural Network

CoNLL-2002 Conference on Natural Language Learning 2002

CRF Conditional Random Fields

DS Distant Supervision

EI Extracción de Información

ELMo Embeddings from Language Models

HMM Hidden Markov model

HTML HyperText Markup Language

IOB Inside, Output, Begining

IOBES Inside, Output, Begining, End, Single

IRI Internationalized Resource Identifier

KB Knowledge Base

KG Knowledge Graph

MEMM Maximum-entropy Markov Model

Mx-news Corpus de noticias

NER Named Entity Recognition

OWL Web Ontology Language

PLN Procesamiento de Lenguaje Natural

POS Part-of-Speech

Q&A Question Answering

RDF Resource Description Framework

RDFS Resource Description Framework Schema

RNN Recurrent Neural Network

RSS Really Simple Syndication

SQL Structured Query Language

SVM Support Vector Machines

URL Uniform Resource Locator

W3C World Wide Web Consortium

XML Extensible Markup Language

Capítulo 1

Introducción

Los datos son flujos de hechos crudos que representan eventos, y por sí mismos no tienen ningún significado. La información es la acumulación de datos que han sido formados en un contexto adquiriendo un significado. La información que está estructurada y organizada como resultado del procesamiento cognitivo y la validación se convierte en conocimiento (Cooper, 2017; Kendal and Creen, 2007).

La información está contenida en grandes colecciones de datos, estos datos pueden ser estructurados, semi-estructurados y no estructurados. Los datos estructurados poseen información con un alto grado de organización, ajustándose a los modelos de datos asociados con bases de datos relacionales u otras formas de tablas de datos, de modo que la inclusión de estos en una base de datos relacional, es transparente y fácil de acceder mediante un motor de búsqueda de una forma simple y directa. En contraste, los datos no estructurados son lo opuesto y suelen encontrarse en documentos de texto plano. Los datos semi-estructurados contienen etiquetas u otros marcadores para separar elementos semánticos, y hacer cumplir las jerarquías de registros y campos, por ejemplo, los documentos XML (Bărbulescu et al., 2013).

La adquisición de conocimiento de fuentes de datos es una tarea que ha sido estudiada a través de los años, y actualmente se sigue una fuerte tendencia de investigación sobre esta área, con la intención de estructurar el conocimiento sobre conceptos, entidades nombradas (personas, lugares, organizaciones, fechas, etc.), atributos y propiedades así como las relaciones existentes entre conceptos o entidades nombradas. Cuando el conocimiento está estructurado, la disposición de este conocimiento proporciona hechos relevantes de una o de múltiples temáticas de conocimiento de una manera fácil y práctica para desarrolladores, investigadores y usuarios finales. Actualmente existen trabajos relacionados con la adquisición del conocimiento sobre grandes volúmenes de información como son las bases/grafos de conocimiento Cyc (Lenat,

1995), Freebase (Bollacker et al., 2008a), Google (Singhal, 2012a), YAGO (Hoffart et al., 2013), Wikidata (Vrandečić and Krötzsch, 2014), DBPedia (Lehmann et al., 2015), entre otros, estos toman como fuente principal de conocimiento a Wikipedia, aunque principalmente de datos no estructurados como la Web. El principal problema que enfrentan las bases de conocimiento que han sido construidas de forma automática y no por especialistas (de forma manual), son las inconsistencias, y errores en las relaciones de sus conceptos. La mayoría de los enfoques existentes para la extracción de conocimiento de la Web se basan en la redundancia de datos web. Por ejemplo, muchos sistemas de extracción de relaciones extraen datos relacionales al descubrir patrones que se expresan con frecuencia en textos web. Estos proyectos se centran en recolectar una gran cantidad de conocimiento general. Sin embargo cuando se trata de un dominio específico, los documentos web relacionados con el dominio son relativamente escasos, lo que dificulta el uso de métodos basados en patrones. Por lo tanto, es importante investigar cómo extraer entidades y relaciones para un dominio específico Yan et al. (2018a).

1.1 Planteamiento del problema

Cuando se desea obtener conocimiento relevante de fuentes de datos, existen muchas formas de extraer la información: de bases de datos, utilizando lenguajes de consulta como SQL, para obtener información de datos estructurados. Si se desea obtener información sobre algún tema o persona, se puede realizar una búsqueda en la Web, y obtener esta información de Wikipedia que contiene información semi-estructurada. Los motores de búsqueda web tradicionalmente utilizaban únicamente un enfoque de Recuperación de Información, que consiste en obtener los documentos más relevantes a una consulta de una enorme cantidad de documentos (páginas web), es decir de datos no estructurados.

Sin embargo con el constante crecimiento de datos, principalmente en la Web, ha sido necesario identificar y proponer nuevas formas de adquisición de conocimiento por parte de empresas y la academia, estos enfoques no necesariamente han sido nuevos, tal como la web semántica propuesta por Berners-Lee et al. (2001) y formalizada por la W3C[1], dónde se describen bases de conocimiento, que son estructuras con conceptos, atributos, entidades y las relaciones entre ellas y reglas para organizar y consultar el conocimiento, como son las ontologías. De este modo las compañías de búsqueda web como Google, Microsoft y Yahoo actualmente cuentan con una base de conocimiento estructurada, que Google (Singhal, 2012a)

[1]https://www.w3.org

popularizo con el nombre de grafo de conocimiento, dónde los nodos son entidades o conceptos y las aristas son las relaciones semánticas entre los nodos.

Paulheim (2017a) menciona que los enfoques para la construcción de los grafos de conocimiento es a través de: curadores (personas con conocimiento profesional o experto) que realizan un trabajo manual que requiere mucho tiempo, y es problemática ya que la enorme cantidad de conceptos y relaciones es complicada para ser formalizada. Otro enfoque consiste en la edición por multitud de expertos (crowd). El enfoque para extracción de datos semi-estructurados de fuentes de bases de conocimiento como Wikipedia o DBPedia. A través de la extracción de información semi-estructurada o no estructurada, generalmente de la Web. Sin embargo cualquiera que sea el enfoque para la construcción del grafo de conocimiento, el resultado no sera perfecto. Además es poco probable, en particular cuando se aplican métodos heurísticos para lograr que el grafo de conocimiento sea totalmente correcto; generalmente hay una compensación entre la cobertura y la corrección, que se aborda de manera diferente en cada gráfico de conocimiento.

Pujara et al. (2013) argumentan que el grafo de conocimiento a menudo es incorrecto, con errores tales como nodos y aristas falsos y faltantes, y etiquetas de nodo faltantes o inexactas. Otro problema común tiene que ver con la extracción de entidades, ya que muchas referencias textuales pueden referirse a la misma entidad del mundo real y son agregadas al grafo de conocimiento sin ser analizadas. Para (Yan et al., 2018a) las técnicas de construcción de grafos de conocimiento, están fuertemente relacionadas con el PLN en si mismo. Por lo que los métodos para la construcción de grafos de conocimiento en inglés, no se pueden aplicar directamente a grafos de conocimiento en otros idiomas, es decir; las reglas y patrones de extracción que funcionan bien para el idioma inglés no funcionan en otro idioma. Por lo tanto, se debe dedicar mucha investigación al mapeo de entidades a sus clases correspondientes y al desarrollo de toda la taxonomía.

Por tal motivo, se propone el diseño de un modelo computacional para la adquisición automática de conocimiento en idioma español, sobre datos no estructurados en sitios web verticales (sitios web enfocados en una temática, por ejemplo: programación, servicios de viajes, noticias, etc.) en español. Al realizar una correcta identificación y extracción de entidades nombradas, así como identificar y extraer las palabras que relacionan ambas entidades. Utilizar un mecanismo de inferencia robusto para la validación y descubrimiento de nuevos hechos que no se observan de forma intrínseca. Finalmente, utilizar los hechos obtenidos para construcción de un grafo de conocimiento.

1.2 Objetivos

A continuación se presenta el objetivo de este trabajo de tesis así como los objetivos particulares que lo componen.

1.2.1 Objetivo general

Construcción de un modelo computacional para la adquisición automática de una base de hechos, para su uso en la inferencia de nuevos hechos y su integración con un grafo de conocimiento.

1.2.2 Objetivos particulares

- Generar un corpus de entidades nombradas sobre política, para adquirir y descubrir conocimiento de forma automática sobre documentos no estructurados en idioma español de México.
- Proponer un método para identificar y extraer relaciones semánticas entre dos entidades nombradas de forma automática.
- Definir un mecanismo de inferencia para descubrir nuevos hechos y generar una base de hechos de forma automática.
- Proponer la construcción de un grafo de conocimiento a partir de los hechos obtenidos en la inferencia.

1.3 Preguntas de investigación

- ¿Cómo se pueden identificar y extraer relaciones entre entidades nombradas de forma automática?
- ¿Es posible descubrir nuevos hechos por medio de inferencia lógica?
- ¿Se puede recuperar automáticamente conocimiento a partir de datos no estructurados?

1.4 Hipótesis

- Es posible inferir nuevo conocimiento, a partir de un conjunto de hechos extraídos de forma automática sobre documentos no estructurados.

1.5 Contribuciones

Las contribuciones de este trabajo se listan a continuación, incluyen los artículos generados a partir de esta investigación.

- Un conjunto de datos con doscientos cincuenta documentos etiquetados de forma manual, con diecisiete clases de entidades nombradas en el dominio político.
- Un método no supervisado para identificar y extraer relaciones de forma automática, usando árboles de dependencia.
- Una base de hechos y un conjunto de reglas definidas de forma genérica para descubrir nuevos hechos.
- Ramos-Flores, O., & Pinto A., D. E.. (2019). "Propuesta para la adquisición automática de hechos utilizando inferencia para la construcción de una grafo de conocimiento". En Habla, mente e ingeniería: Aplicaciones de la ingeniería del lenguaje y del conocimiento.
- Ramos Flores, O., & Pinto, D. (2019). "Reconocimiento de entidades nombradas enfocado en noticias políticas". En Avances en tecnologías del lenguaje y el conocimiento.
- Ramos Flores, O., & Pinto, D. (2020). "Proposal for Named Entities Recognition and Classification (NERC) and the Automatic Generation of Rules on Mexican News". Computación y Sistemas. Vol. 24, No. 2, pp 533-538. DOI: 10.13053/CyS-24-2-3377
- Ramos-Flores, O., Pinto, D., Montes-y-Gómez, M., & Vázquez, A. (2020). "Probabilistic vs deep learning based approaches for narrow domain NER in Spanish". Journal of Intelligent & Fuzzy Systems. Vol. 39, No. 2, pp 2015-2025. DOI: 10.3233/JIFS-179868

1.6 Distribución de la Tesis

En el Capítulo 1 se presenta este trabajo de tesis, los antecedentes de los que se parte, el planteamiento del programa y los objetivos. Además se presentan las contribuciones hechas sobre este trabajo y los artículos publicados.

El marco teórico es descrito en el segundo Capítulo 2. Se definen los esquemas de etiquetado IOB e IOBES, los modelos empleados para el Reconocimiento de Entidades Nombradas (NER, por sus siglas en inglés, árboles de dependencia universal, sintaxis de Prolog y el grafo de conocimiento (KG, por sus siglas en inglés).

Los trabajos relacionados se muestran en el Capítulo 3, se presenta literatura sobre la tarea NER empleando el algoritmo de Campos Aleatorios Condicionales (CRF, por sus siglas en inglés), y la arquitectura de Redes Neuronales Recurrentes (RNN, por sus siglas en inglés) para la Memoria Bidireccional a Largo Plazo (Bi-LSTM, por sus siglas en inglés). Así como trabajos relacionados con la Extracción de Relaciones, la generación de Reglas Lógicas y Grafos de conocimiento.

El Capítulo 4 describe la metodología general propuesta, así como cada etapa en el flujo de trabajo. Iniciando con la metodología para la tarea NER, describiendo el corpus de noticias (Mx-news) y el corpus CoNLL-2002. La segunda etapa para extraer relaciones de forma automática, donde se presenta el método empleando árboles de dependencia para identificar y extraer las relaciones. En la etapa tres, se presenta el proceso para transformar las relaciones obtenidas a una base de hechos. En la etapa final, se definen los esquemas para la construcción del grafo de conocimiento.

Los experimentos y resultados obtenidos son presentados en el Capítulo 5. Se detallan los experimentos realizados en cada una de las etapas, así como los resultados obtenidos en cada una de ellas. Finalmente, las conclusiones y trabajo futuro se discuten en el Capítulo 6.

Capítulo 2

Marco Teórico

El capítulo describe los conceptos, herramientas y recursos utilizados para la adquisición automática de conocimiento, como es el reconocimiento de entidades nombradas y extracción de relaciones entre entidades.

2.1 Modelos utilizados

En esta sección se describen los modelos utilizados con enfoques probabilístico y de aprendizaje profundo, además de los esquemas de etiquetado utilizados en los conjuntos de datos.

2.2 Esquemas de etiquetado IOB e IOBES

Los esquemas de etiquetado utilizados en la tarea de fragmentación de texto (text chunking) o el análisis superficial (shallow parsing), que consiste en encontrar frases no recursivas en una oración dada del texto en lenguaje natural. Estos esquemas de etiquetado son implementados en los corpus CoNLL-2002 y el corpus Mx-news, para el primer conjunto de datos se utilizó el esquema IOB (Inside/Output/Begining), y para el segundo conjunto de datos se utilizó el esquema de etiquetado IOBES (Inside/Output/Biginign/Single).

El trabajo de Ramshaw and Marcus (1995) presenta un método donde los fragmentos de frases nominales se codificaron como etiquetas IOB en palabras, donde palabras marcadas con la etiqueta I están dentro de un fragmento, aquellas marcadas con O están fuera, y la etiqueta B es usada para marcar al elemento más a la izquierda del fragmento, quienes inmediatamente siguen otro fragmento.

Tabla 2.1 Etiquetado de entidades nombradas sobre los esquemas IOB e IOBES.

No.	**Palabra**	**Entidad**	**Esquemas**							
			I	**O**	**B**	**I**	**O**	**B**	**E**	**S**
1	Sheinbaum	PER (persona)	-	-	✓	-	-	-	-	✓
2	la	-	-	✓	-	-	✓	-	-	-
3	jefa	TIT (título)	-	-	✓	-	-	✓	-	-
4	de	TIT (título)	✓	-	-	✓	-	-	-	-
5	gobierno	TIT (título)	✓	-	-	✓	-	-	-	-
6	electa	TIT (título)	✓	-	-	-	-	-	✓	-
7	mantuvo	-	-	✓	-	-	✓	-	-	-
8	una	-	-	✓	-	-	✓	-	-	-
9	reunión	-	-	✓	-	-	✓	-	-	-
10	con	-	-	✓	-	-	✓	-	-	-
11	AMLO	PER (persona)	-	-	✓	-	-	-	-	✓

Más tarde en el trabajo de Uchimoto et al. (2000) dividen las entidades nombradas en cuatro tipos de "*subetiquetas*" que representaban el inicio, las que están dentro, al final y aquellas que están compuestas por una sola entidad. Shen and Sarkar (2005) retoman este tipo de "*subetiquetas*" para etiquetar entidades nombradas, las definen como IOBES: la palabra inicial marcada con la etiqueta B, I para las palabras dentro del fragmento, el final es indicado por la etiqueta E y para entidades de una única palabra la etiqueta S es usada.

Para mostrar ambos etiquetados se presenta el siguiente ejemplo para la oración: "*Sheinbaum la jefa de gobierno electa mantuvo una reunión con AMLO*" el etiquetado sobre ambos esquemas se muestra en la Tabla 2.1, donde se describe la forma de anotar (etiquetar) las palabras (tokens) de una oración, en ambos esquemas la etiqueta "O" indica que el token en cuestión no es una entidad nombrada. En el esquema IOB la primera palabra es marcada con la etiqueta "B" y las siguientes con la etiqueta "I". En el esquema IOBES la primera palabra es marcada con la etiqueta "B", las palabras intermedias son marcadas con la etiqueta "I" y la palabra final con la etiqueta "E", cuando la entidad está conformada por sólo una palabra se marca con la etiqueta "S".

2.2.1 Campos Aleatorios Condicionales

El modelo de CRF es un framework de modelado de secuencias desarrollado por Lafferty et al. (2001), que posee todas las ventajas de los Modelos de Máxima Entropía de Markov (MEMM, por sus siglas en inglés) y resuelve el problema de sesgo (bias) de la etiqueta. El

problema del sesgo de la etiqueta esencialmente es que las observaciones futuras no pueden afectar la distribución posterior sobre los estados anteriores (Sutton and McCallum, 2012). La principal diferencia utilizada por los CRF es un modelo exponencial único para la probabilidad conjunta $p(x|y)$ de la secuencia completa de etiquetas dada la secuencia de observación. Se definen dos variables aleatorias; $X = x_1, \ldots, x_T$ es la variable sobre las secuencias de datos de las observaciones (tokens) que deben etiquetarse, la segunda es $Y = y_1, \ldots, y_T$ que es la variable sobre las secuencias de etiquetas correspondientes (entidades nombradas) (Lafferty et al., 2001; Sutton and McCallum, 2012). Formalmente un CRF de cadena lineal se puede definir como:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp\left(\sum_{k}^{K} \theta_k f_k(y_{t-1}, y_t, x_t)\right) \tag{2.1}$$

donde f_k denota una de las funciones del indicador binario (o característica) de K, cada una ponderada por $\theta_k \in \mathbb{R}$ y Z es un término de normalización, que itera sobre todas las asignaciones posibles.

$$Z(x) = \sum_{y} exp\left(\sum_{k}^{K} \theta_k f_k(y_{t-1}, y_t, x_t)\right) \tag{2.2}$$

La Figura 2.1 representa un ejemplo de la estructura de un CRF, donde la oración X es: "*López Obrador viaja a Puebla*", y las secuencias de etiquetas correspondientes Y son: "*PER-B, PER-E, O, O, GPE-S*". Donde las entradas y salidas están conectadas directamente a diferencia de otros modelos como las redes neuronales.

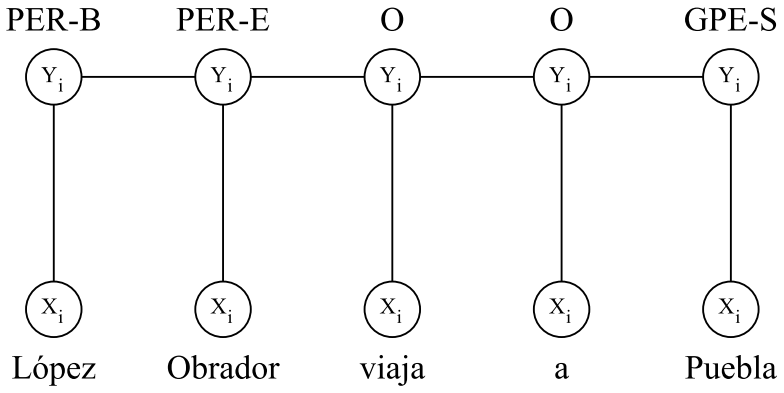

Fig. 2.1 Estructura de un CRF. Adaptado de (Huang et al., 2015; Lafferty et al., 2001)

2.2.2 Redes Neuronales

El modelo de red neuronal utilizado es un tipo especial llamado RNN, que posee una arquitectura de Memoría Bidireccional a Largo Plazo (Bi-LSTM, por sus siglas en inglés) propuesta por Graves and Schmidhuber (2005); Hochreiter and Schmidhuber (1997). Las RNN proporcionan una forma muy elegante de tratar con datos secuenciales (tiempo), que incorporan correlaciones entre puntos de datos que están cerca de la secuencia (Schuster and Paliwal, 1997). Las redes recurrentes pueden, en principio, usar sus conexiones de retro-alimentación para almacenar representaciones de eventos de entrada recientes en forma de activaciones ("memoria a corto plazo", a diferencia de "memoria a largo plazo" representada por pesos que cambian lentamente) Hochreiter and Schmidhuber (1997).

La red de Memoria a Largo Plazo (LSTM) que consiste en un conjunto de bloques recurrentemente conectados, cada uno contiene una o más células de memoria conectadas de forma recurrente, y tiene la capacidad de eliminar o agregar información al estado de la celda regulado por tres unidades multiplicativas: entradas, salidas y puertas de olvido. Dada una secuencia de entrada $x = (x_1, \ldots, x_T)$ una RNN estándar calcula la secuencia vectorial oculta $h = (h_1, \ldots, h_T)$ y el vector de secuencias de salida $y = (y_1, \ldots, y_T)$ mediante la iteración de las ecuaciones siguientes de $t = 1$ a T. Para la tarea NER x e y representan la entrada de características y etiquetas respectivamente. La red LSTM contiene tres compuertas, que son funciones de la entrada actual x_t, del estado oculto h_t: compuerta de entrada i_t compuerta de olvido f_t y compuerta de salida o_t (Graves et al., 2013; Huang et al., 2015). La Figura 2.2 ilustra una simple célula de la red LSTM que es implementado por las siguientes ecuaciones:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2.3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2.4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \tag{2.5}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.6}$$

$$h_t = o_t tanh(c_t) \tag{2.7}$$

donde σ es la función sigmoide logística, los términos W denotan las matrices de peso utilizadas, para asignar la entrada de la capa oculta a tres puertas y el estado de la celda de entrada, b_i, b_f, b_o, b_c son vectores de sesgo (bias) y c son los vectores de activación celular, en

los cuales todos tienen el mismo tamaño que el vector oculto h (Graves et al., 2013; Huang et al., 2015).

La Figura 2.3 muestra una red LSTM bidireccional (Bi-LSTM) en la que se accede al contexto de largo alcance en ambas direcciones, esto es, se puede entrenar una red Bi-LSTM utilizando toda la información de entrada disponible en el pasado (estados hacia adelante), y en el futuro (estados hacia atrás) de un secuencia de tiempo específico, a diferencia de la red LSTM que solo puede hacer uso del contexto anterior. En la Figura 2.3, los cuadros rellenos representan la célula LSTM (también se conoce como la capa oculta de una red LSTM). La Bi-LSTM conecta dos capas ocultas a una sola capa de salida, tanto la secuencia oculta del avance hacia adelante ($\overrightarrow{h}_t$) como la salida oculta de la secuencia hacia atrás ($\overleftarrow{h}_t$) se calculan utilizando las ecuaciones de la red LSTM (2.3-2.7) iterando la capa hacia adelante de $t = 1$ hasta T y la capa hacia atrás desde $t = T$ hasta 1 (Graves et al., 2013; Graves and Schmidhuber, 2005).

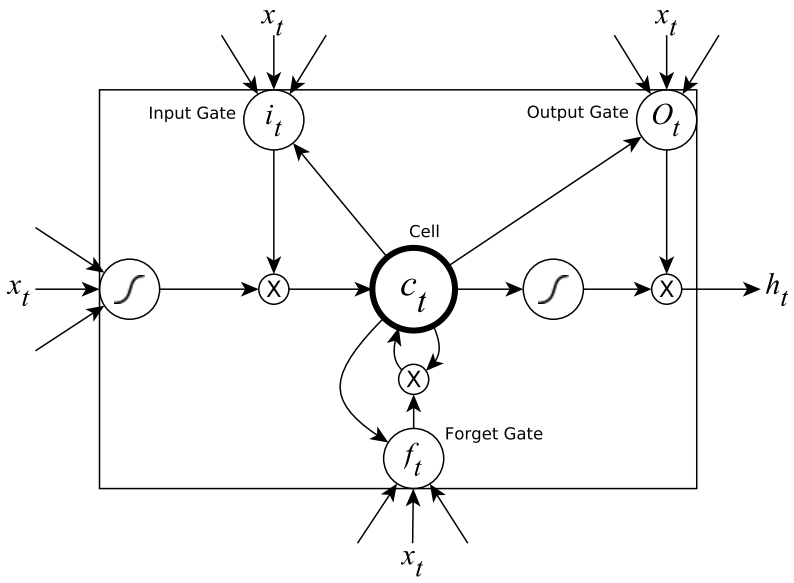

Fig. 2.2 Célula de una red LSTM. Fuente (Graves et al., 2013)

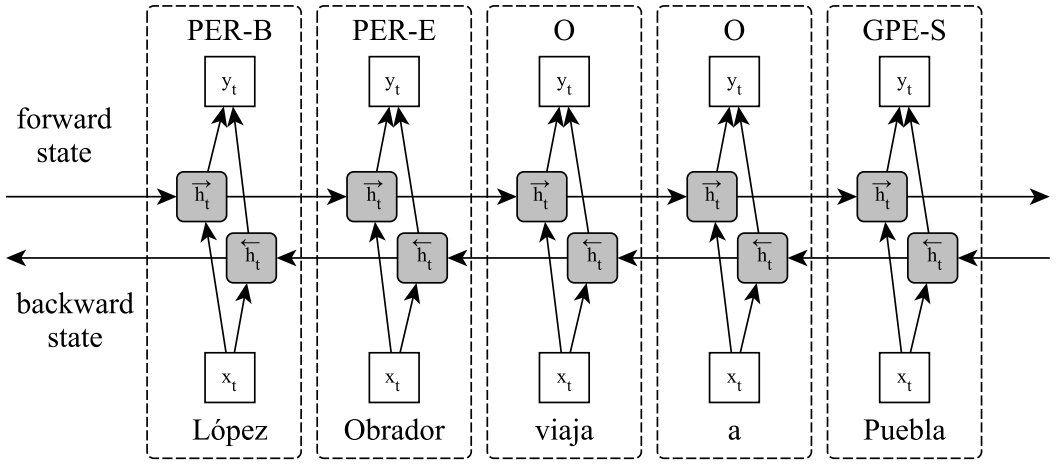

Fig. 2.3 Estructura de una red LSTM bidireccional. Adaptada de (Huang et al., 2015; Schuster and Paliwal, 1997)

2.3 Análisis de dependencias universales

Las Dependencias Universales[1] son un marco para la anotación coherente de la gramática (partes del habla, características morfológicas y dependencias sintácticas) en diferentes lenguajes humanos. La representación tipada de las dependencias fue diseñada originalmente por de Marneffe and Manning (2008).

Para el análisis de dependencia se hace uso de la biblioteca de Python Spacy[2] empleando un modelo en idioma español [3]. La herramienta permite crear el árbol de dependencias de un texto, así como etiquetar cada uno de las palabras con POS y con relaciones de dependencia universal entre palabras.

2.3.1 Relaciones de Dependencia Universal

Las relaciones de dependencia universal usadas por la biblioteca Spacy para ser identificadas y etiquetar oraciones son *"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* y *xcomp"*.

[1] https://universaldependencies.org/
[2] https://spacy.io
[3] https://spacy.io/models/es

Para desplegar de forma gráfica el árbol de dependencias se usa la biblioteca de Python Networkx [4] para transformar a grafo los datos resultantes y la biblioteca Graphviz [5] para dibujar el árbol.

La Figura 2.4 describe de forma gráfica el árbol de dependencias generado a partir de la oración: *Sheinbaum_PER la jefa_de_gobierno_electa_TIT mantuvo una reunión con AMLO_PER* . Previamente se han etiquetado entidades nombradas dentro de la oración. Del tipo Título (TIT) y Persona (PER). Como se observa la herramienta no descompone la entidad en tokens (palabras y símbolos) como ocurre con el resto de la oración. Para ello se reemplazan los espacios entre las palabras de la entidad por guiones bajos.

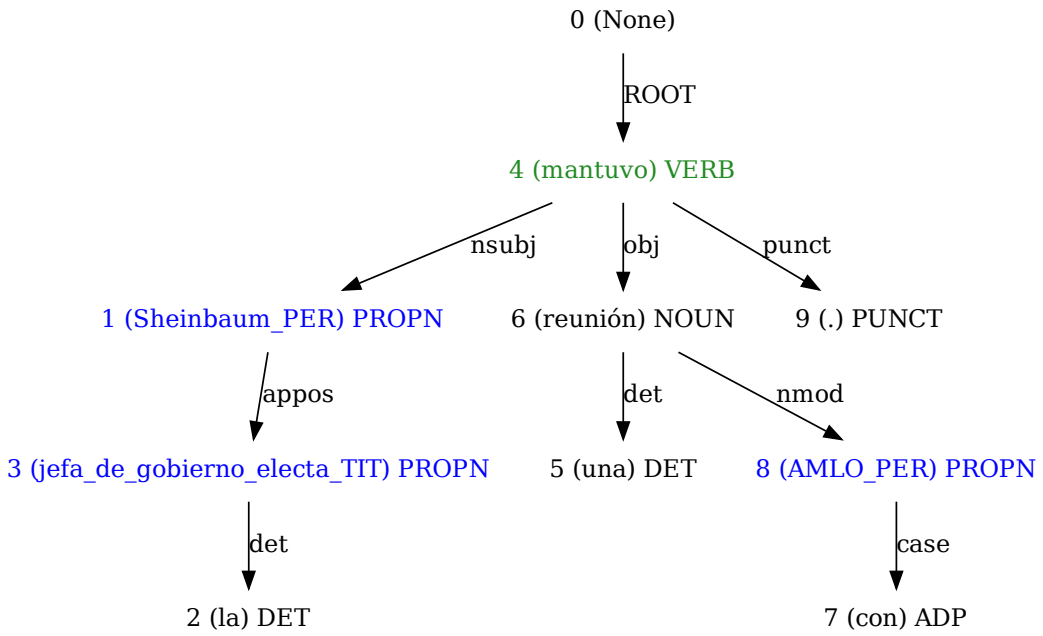

Fig. 2.4 Árbol de dependencias generado por el modelo Spacy e ilustrado con Graphviz.

El árbol de la Figura 2.4 inicia en la raíz (ROOT). Las palabras representan a los nodos y las aristas a las relaciones de dependencia universal. A la izquierda de cada palabra se encuentra su ID, entre paréntesis la palabra o signo de puntuación y al lado derecho la etiqueta POS.

[4]https://networkx.org/documentation/stable/
[5]https://graphviz.readthedocs.io/en/stable/manual.html

2.4 Prolog

Prolog es un lenguaje de programación lógico declarativo de propósito general. Además es considerado como herramienta en el PLN por Pereira and Shieber (2002) "*el lenguaje de programación lógica Prolog cuya columna vertebral es el formalismo de cláusulas definidas, como una herramienta para implementar los componentes básicos de los sistemas de procesamiento del lenguaje natural*".

Prolog cuenta con mecanismos integrados de búsqueda y unificación que lo convierten en un candidato ideal para implementar modelos formales de lingüística. Asimismo, es de naturaleza declarativa que permite centrarse en la resolución de problemas. Las reglas declarativas son una forma autodocumentada simple pero poderosa de representar y modelar el conocimiento del dominio (Bitter et al., 2010).

Los programas de Prolog están estructurados en términos de relaciones, expresadas entre entidades. Las llamadas a funciones corresponden a consultas sobre si una relación particular se cumple o no y en qué condiciones. Esta diferencia tiene tremendas ramificaciones. Significa que las variables juegan un papel completamente diferente en Prolog que en los lenguajes convencionales (Liu, 2018).

2.4.1 Sintaxis de Prolog

Los lenguajes de programación lógica contienen tres construcciones posibles: *datos, reglas, predicados* y *consultas*. Los datos permiten representar información que se considera incondicional, mientras que las reglas se utilizan para inferir o deducir información a partir de los datos (o de otras reglas), los predicados son datos que presentan una relación entre sus argumentos, y las consultas permiten cuestionar o consultar al programa para conocer si la información es falsa o verdadera, esto se infiere a partir de los datos y las reglas del programa (Dahl and García, 2018).

Datos

Los *datos* también pueden ser nombrados como *hechos* o *axiomas*. Además, un *hecho* es una regla que no contiene hipótesis por lo que se denota como aserción incondicional. A continuación se listan algunos *hechos* (Dahl and García, 2018).

```
madre("maria").
genero("femenino").
fecha("miercoles").
organizacion("secretaria de educacion publica").
lugar_nacimiento(P,_lugar).
```

La sintaxis que compone a los hechos está compuesta por **términos**. Donde un **término** puede ser una *variable*, un *átomo*, un *número* o un *término compuesto*.

- **Variables**. Se denotan como una secuencia de letras, dígitos o el símbolo "_" (guión bajo). Los lineamientos establecen que una variable debe iniciar con una letra mayúscula o con "_". Ejemplo: *P, X, _lugar, _otra_variable*.
- **Átomos**. Estos se denotan como una secuencia de símbolos entre comillas, una secuencia de letras, dígitos o con el símbolo "_". Obligatoriamente deben comenzar con una letra minúscula. Ejemplo: *x, y, madre, "femenino", _lugar_de_nacimiento*.
- **Números**. Se incluyen los enteros y los reales de punto flotante. Ejemplo: *2, -2, 2.5, -2.5*.
- **Término compuesto**. Es una expresión que tiene un nombre, usualmente llamado *functor*. Este debe ser un átomo y una serie de términos (*argumentos*) separados por comas, estos términos deben estar entre paréntesis. Además un *término compuesto* también puede estar como argumento. Ejemplo: *lugar_nacimiento(P, _lugar), nieto_de(X,Y), p(a,p(a,X))*.

Reglas

Las reglas permiten establecer aserciones condicionales. De la regla mostrada abajo, la *aserción_0* a la izquierda del símbolo ":-" es llamada **conclusión** de la regla, en cambio las aserciones a la derecha son llamadas **hipótesis** o **premisas**. Todas las reglas y hechos deben terminar con el símbolo "." (punto). El símbolo "," (coma) indica la *conjunción* y el símbolo ";" (punto y coma) la *disyunción* entre aserciones de las *premisas* (Dahl and García, 2018; Liu, 2018). A continuación se presenta la forma de una regla en general así como algunos ejemplos.

```
aserción_0 :- aserción_1, aserción_2, ..., aserción_k.
madre(X, Y) :- hija(Y,X); hijo(Y,X).
presidente(X, Y) :- pais(Y), persona(X).
```

Predicados

Los *predicados* son *relaciones* o *propiedades* que presenta un término. Los valores estructurados se pueden capturar fácilmente usando relaciones. Esta abstracción de datos de alto nivel permite que se capturen fácilmente sin detalles de implementación de bajo nivel (Liu, 2018). A continuación se describen algunos ejemplos de predicados. El símbolo "%" (porcentaje) se usa para definir comentarios en Prolog.

```
fundacion_BUAP(14,04,1578) % Fecha (dd-mm-yyy) de fundación de la BUAP
genero(masculino) % Es de genero masculino
municipio("cholula","puebla") % El municipio de Cholula pertenece al estado de Puebla
```

Consultas

Las consultas también llamadas *preguntas* son una secuencia de predicados separados por comas (*conjunción*) , utilizadas para interrogar o consultar si cierta información puede inferirse de un programa. Una consulta puede tener variables o no. Cuando la consulta tiene variables se llama *modo averiguación* donde el interprete de Prolog devuelve información en las variables establecidas, en caso contrario cuando no se usan variables se llama *modo verificación* el interprete de Prolog arrojara como respuesta "*true*" o "*false*" (Dahl and García, 2018). A continuación se describen algunos ejemplos sobre los dos modos de consulta.

```
%% Definición de los datos
fundacion_BUAP(14,04,1578).
fundacion_ciudad("puebla de zaragoza", puebla).
```

Consultas en **modo verificación** sobre el interprete de Prolog

```
?- fundacion_BUAP(14,04,1578).
true.

?- fundacion_BUAP(14,04,1500).
false.
```

Consultas en **modo averiguación** sobre el interprete de Prolog

```
?- fundacion_BUAP(D,M,Y).
D = 14,
M = 4,
Y = 1578.

?- fundacion_BUAP(_,_,Y).
Y = 1578.
```

2.5 Grafo de conocimiento

La Web semántica ha promovido la representación gráfica del conocimiento al definir e impulsar el uso de Marcos de Descripción de Recursos (RDF, por sus siglas en inglés), Esquemas de Marcos de Descripción de Recursos (RDFS, por sus siglas en inglés), Lenguaje de Ontología Web (OWL, por sus siglas en inglés), etc. en cuya representación gráfica las *entidades* que son los *nodos*, están conectadas por *relaciones* que son las *aristas* de la representación gráfica, donde las entidades pueden tener tipos, denotados como relaciones "*is a*" (por ejemplo: López Obrador *es una* persona) y atributos definidos con tipos de datos (por ejemplo: cadena, número entero, número flotante, etc.) (Paulheim, 2017a).

Los KG permiten estructurar entidades como por ejemplo: *organizaciones*, *personas* o *lugares* de los que se tenga el conocimiento, además de poder obtener al instante información relevante para su consulta Singhal (2012b). La Figura 2.5 ilustra el modelo de datos basado sobre un grafo RDF, con dos nodos (*sujeto* y *objeto*) y una arista que los conecta (*predicado*).

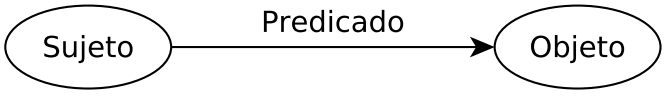

Fig. 2.5 Estructura gráfica básica de un grafo RDF. Fuente[6].

[6]https://www.w3.org/TR/rdf11-concepts/ *Recuperado:25-Mayo-2020*

2.5.1 Construcción del grafo de conocimiento

Para la construcción se emplea un Identificador de Recursos Internacionalizado (**IRI**, por sus siglas en inglés) o **literal** y denota cualquier "*cosa*". Esas "*cosas*" son llamadas **recursos** (*entidades*), por lo que cualquier "*cosa*" puede ser un **recurso**, por ejemplo: *documentos*, *conceptos abstractos*, *números* y *cadenas*. El **recurso** denotado por un **IRI** se llama *referente* y el **recurso** denotado por una **literal** se llama *valor literal*.

Las **literales** tienen tipos de datos que definen el rango de valores posibles, como *cadenas*, *números* y *fechas*. Tipos especiales de **literales**, como son las cadenas con etiquetas de idioma, denotan cadenas de texto sin formato en un lenguaje natural.

De este modo un *predicado* mantiene una relación entre el **recurso** denotado por un *sujeto* y un *objeto*. Esta definición de una tripleta RDF es conocida como **declaración RDF**, donde el *predicado* por si mismo es un **IRI** y denota una **propiedad**, es decir, un recurso que puede considerarse como una relación binaria.

2.5.2 Vocabularios RDF y espacios de nombres IRI

Un **vocabulario RDF** es una colección de IRI destinados a ser utilizados en grafos RDF. Los IRI en un vocabulario RDF regularmente comienzan con una subcadena común conocida como IRI de espacio de nombres (*namespace IRI*). Algunos *namespace IRI* están asociados por convención con un nombre corto conocido como **prefijo de espacio de nombres** (*Namespace prefix*).

Tabla 2.2 Ejemplo de espacios de nombre (*namespaces*).

Prefijo	**IRI**	**Vocabulario RDF**
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Vocabulario RDF incorporado en el IRI
rdfs	http://www.w3.org/2000/01/rdf-schema#	Vocabulario del esquema RDF
xsd	http://www.w3.org/2001/XMLSchema#	Los tipos XSD compatibles con RDF
foaf	http://xmlns.com/foaf/0.1/	Vocabulario del esquema FOAF para vincular personas e información a través de la Web

La Tabla 2.2 describe ejemplos del uso de *namespace prefix* (primer columna) usados para abreviar el *namespace IRI* (segunda columna) de esquemas con vocabularios RDF definidos para su uso en el grafo.

Ejemplo de grafo RDF escrito en XML, se tiene la tripleta <*Benito Juárez, fue presidente de, México*> compuesta por dos entidades, el sujeto es de tipo *persona* y el objeto es de tipo *lugar* vinculadas por el predicado *fue presidente*. El código RDF para generar el grafo correspondiente se describe a continuación.

```
1:  <?xml version="1.0"?>
2:  <rdf:RDF
3:    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:    xmlns:foaf="http://xmlns.com/foaf/0.1/"
5:    xmlns:lke="http://lke.buap.mx/0.1/">
6:    <!-- definición nodo sujeto -->
7:    <rdf:Description rdf:about="http://lke.buap.mx/0.1/per/Benito_Juarez">
8:      <rdf:type rdf:resource="http://dbpedia.org/ontology/Person"/>
9:     <foaf:name xml:lang="es">Benito Juárez</foaf:name>
10:     <!-- definición de la relación -->
11:     <lke:fue_presidente_de rdf:resource="http://lke.buap.mx/0.1/geo/Mexico"/>
12:   </rdf:Description>
13:   <!-- definición nodo objeto -->
14:    <rdf:Description rdf:about="http://lke.buap.mx/0.1/geo/Mexico">
15:      <rdf:type rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
16:     <foaf:name xml:lang="es">México</foaf:name>
19:   </rdf:Description>
20: </rdf:RDF>
```

Fig. 2.6 Grafo RDF en lenguaje XML de ejemplo.

Los *namespaces* son los primeros en definir, así como su *prefijo* (rdf, foaf y lke). A continuación se describe el primer nodo llamado *Benito Juárez* que contiene su IRI correspondiente (*http://lke.buap.mx/0.1/Benito_Juarez*), así mismo se define el *tipo/clase* a la que corresponde este nodo que es *persona* y está definido en el IRI de *foaf* por lo que se establece el recurso como *http://xmlns.com/foaf/0.1/Person*. Además se define el *nombre* del recurso (nodo/entidad) como una *literal* de tipo cadena (<*foaf:name xml:lang="es"*>) y estableciendo el idioma del recurso. Por último se define el *predicado* que se vincula con el recurso de *México* (<*lke:fue_presidente_de rdf:resource="http://example.org/Mexico"/*>).

De forma similar se define el recurso de *México*, en este caso el recurso es de tipo/clase *lugar* y se asocia al IRI de DBPedia para lugares poblados (*http://dbpedia.org/ontology/PopulatedPlace*).

La Figura 2.7 muestra el grafo RDF en forma gráfica, donde los *nodos* son las elipses, las *literales* son los rectángulos y los *predicados* son las aristas. Existen en la web sitios

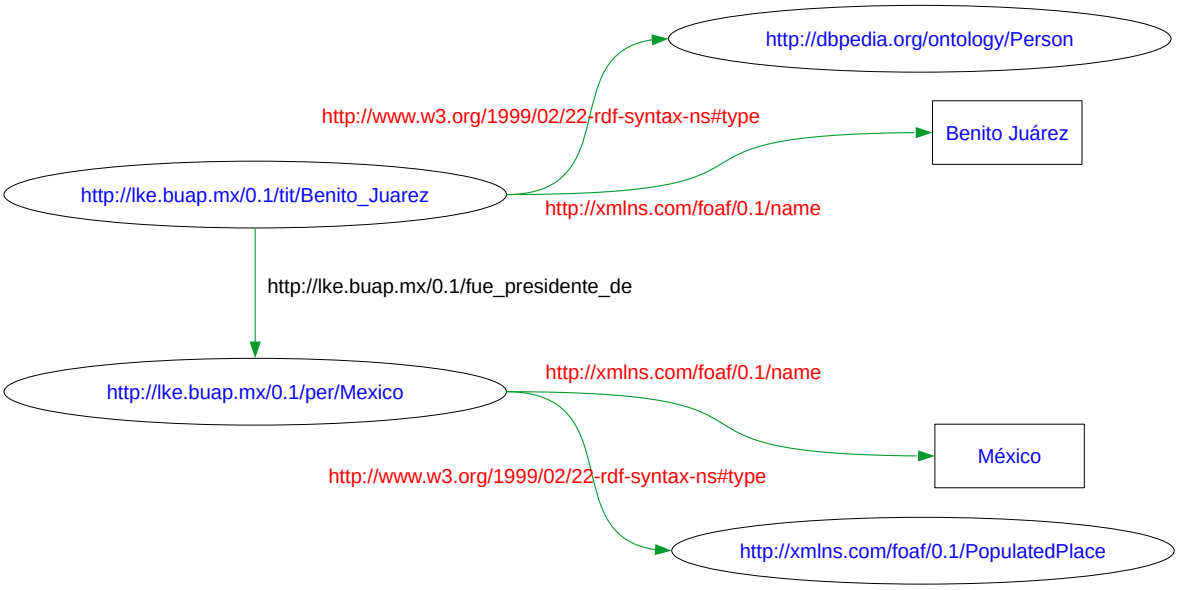

Fig. 2.7 Representación gráfica de un KG.

validadores[7] de código RDF donde se puede verificar la sintaxis del grafo, como resultado se obtiene una tabla con todas las tripletas presentes en el grafo y además permite generar la representación en forma gráfica.

[7]https://www.w3.org/RDF/Validator/

Capítulo 3

Revisión del Estado del Arte

Este capítulo se describen trabajos sobre el reconocimiento de entidades nombradas enfocado a los métodos probabilísticos y aprendizaje profundo, la extracción de relaciones entre entidades, generación de reglas lógicas y la construcción de grafos de conocimiento.

3.1 Reconocimiento de Entidades Nombradas

El Reconocimiento de Entidades Nombradas es una subtarea de la Extracción de Información, el término "*Entidad Nombrada*" fue acuñado en la Conferencia de Comprensión de Mensajes 6 (MUC-6), donde se presentó la tarea para identificar los *nombres de todas las personas, organizaciones y ubicaciones geográficas* en un texto. Una subtarea adicional involucró también identificar entidades de *tiempo, monetarias y expresiones de porcentajes* (Grishman and Sundheim, 1996). Algunos trabajos han sido enfocados en los Modelos Ocultos de Markov (HMM) (Bikel et al., 1997; Todorović et al., 2011), así como Modelos de Máxima Entropía de Markov (MEMM) (McCallum et al., 2000; Todorović et al., 2011; Uchimoto et al., 2000), Máquinas de Vectores de Soporte (SVM) (Asahara and Matsumoto, 2003; Chesney et al., 2017; Kudo and Matsumoto, 2001) para identificar entidades nombradas (chunks) en idioma Inglés y Jápones. En la competencia de CoNLL-2002 (Conference on Computational Natural Language Learning) para la extracción de entidades nombradas en idioma Español y Holandés, se presentó el trabajo de Carreras et al. (2002) usando el algoritmo binario Adaptive Boosting (AdaBoost), combinando varios árboles de decisión pequeños de profundidad fija. Los modelos más populares para la tarea NER son los Campos Aleatorios Condicionales ya que pueden producir una mayor precisión de etiquetado general.

3.1.1 Campos Aleatorios Condicionales

Los Campos Aleatorios Condicionales (CRF) fueron descritos por Lafferty et al. (2001) ya que poseen todas las ventajas de los MEMM y resuelven el problema de sesgo (bias) de la etiqueta. En el trabajo de Sha and Pereira (2003) usaron el modelo CRF para abordar la tarea de etiquetado de secuencias de frases nominales, utilizando funciones de análisis superficiales con etiqueta POS (Part-of-speech) bajo el esquema IOB (Inside, Output Begining). En el trabajo de McCallum and Li (2003) realizan sus experimentos sobre dos corpus del CoNLL-2003 (Inglés y Alemán), presentan un método de inducción de características para los CRF, que consiste en obtener las semillas de los datos etiquetados y usar la Web para aumentar significativamente los datos en los lexicones, además de usar un conjunto de características que consisten en pruebas singleton y conjunciones binarias entre pruebas y con características actuales en el modelo.

Esta tarea es abordada por Mozharova and Loukachevitch (2017) en idioma ruso para identificar nombre de personas, organizaciones y lugares. Además de haber etiquetado entidades de tipo organización, nombre de organizaciones de medios, lugares, y entidades geopolíticas. Para el etiquetado emplearon el esquema IOB. En sus experimentos utilizaron un clasificador CRF para dos conjuntos de datos, bajo el esquema IO e IOB. Para el modelo CRF utilizaron características de los tokens, sobre su contexto y basadas en lexicones, alcanzando un F-Score de 0.96 y 0.81 en los experimentos.

Para reconocer entidades en registros médicos electrónicos Liu et al. (2017) en idioma chino emplearon un conjunto de datos con 830 mil registros. El clasificador empleado es un modelo basado en CRF, donde las características empleadas fueron bolsa de caracteres, parte de la oración (POS), características de diccionarios anotados por profesionales, características de agrupamiento (basados en la relación entre las palabras) y características léxicas. En sus experimentos alcanzaron un F-Score de 88.82%.

3.1.2 Aprendizaje profundo

Con el surgimiento del Deep Learning (Aprendizaje profundo) se han desarrollado diferentes enfoques para resolver el reconocimiento de entidades nombradas, utilizando modelos con incrustaciones de vectores *Word2Vec, Glove, Senna y FastText* (Sienčnik, 2015). Además se han utilizando redes neuronales convolucionales (CNN) y recurrentes (RNN), el trabajo de Huang et al. (2015) fue uno de los primeros en usar una arquitectura Bi-LSTM (Bidirectional Long Short-Term Memory) de una RNN y una capa adicional que usa Campos Aleatorios Condicionales (CRF: Conditional Random Fields), en su trabajo realizan diferentes

experimentos con diferentes combinaciones (LSTM, BI-LSTM, CRF, LSTM-CRF, y BI-LSTM-CRF), donde la red neuronal BI-LSTM-CRF es la mejor evaluada. En el trabajo de Ma and Hovy (2016) utilizan una combinación de CNN y RNN, usan incrustaciones a nivel de carácter de cada palabra como entrada para una CNN, su salida alimenta una Bi-LSTM (Bidirectional Long Short-Term Memory), finalmente la salida de esta red alimenta una capa de la red principal basada en un CRF (Conditional Random Fields) para clasificar la mejor secuencia de etiquetas. Ademas en sus experimentos también utilizan como entrada embeddings (incrustaciones de palabras en vectores) aleatorios, Senna, Word2Vec y GloVe así como dos diferentes arquitecturas de redes (LSTM-CNN y LSTM-CNN-CRF).

El reconocimiento de entidades es una tarea de la EI que continua siendo estudiada hoy en día, debido a que suelen aplicarse métodos para dominios específicos y en diferentes idiomas. Los métodos más populares son los basados en CRF, porque se utilizan para predecir las secuencias de información contextual, para agregar información que será utilizada por el modelo para hacer una predicción correcta. Por la misma razón se emplean redes neuronales recurrentes, además de apoyarse en embeddings para incrementar la precisión y recall en el reconocimiento de entidades. Aunque no existen embeddings en diferentes idiomas, existen herramientas que facilitan la creación de un conjunto de embeddings apropiado al lenguaje. Los algoritmos CRF son los más adecuados cuando se tiene un conjunto de entrenamiento pequeño, a diferencia de las redes neuronales que necesitan grandes cantidades de datos de entrenamiento para obtener buenos resultados.

3.2 Extracción de Relaciones Semánticas

En la Tabla 3.1 se muestran trabajos sobre el reconocimiento y extracción de relaciones semánticas. Girju et al. (2007) presentaron la Tarea 4 del SemEval-2007 para la clasificación de relaciones entre nominales definiendo siete relaciones. Principalmente se basan en identificar y extraer hiponimia e hiperonimia para identificar relaciones del tipo *is-a* or *is-kind-of*, además de proponer una serie de patrones léxico-sintácticos utilizando los hipónimos e hiperónimos (por ejemplo: $< hipónimo, is-a/is-kind-of, hiperónimo >$) como lo hace Galicia-Haro and Gelbukh (2014). Además de estas relaciones mencionadas antes, Ta and Thi (2016) definen otras relaciones semánticas como *part-of, made-of*, etc. Otra técnica empleada por Punuru and Chen (2012) es utilizando un enfoque estadístico para formar relaciones semánticas, dónde el elemento principal es el verbo que une la tripleta, obtienen los verbos modificando la medida TF-IDF y la llaman VF-IDF, de este modo rankean y obtienen los mejores candidatos, además de

utilizar herramientas de procesamiento de lenguaje natural para obtener POS y frases nominales. En el trabajo de Borzì et al. (2014) identifican relaciones semánticas entre conceptos de forma automática usando información estadística extraída de la Web, para construir una red asociativa ponderada a partir del léxico de WordNet en inglés, aumentado con las entidades enciclopédicas de Wikipedia.

Un enfoque para la clasificación de relaciones utilizando modelos de lenguaje pre-entrenados es el de Wu and He (2019), argumenta que este tipo de modelos han mostrado ser efectivos para mejorar muchas tareas de PLN. El modelo que propone es llamado R-BERT, para hacer que el módulo BERT (Devlin et al., 2018) capture la información de ubicación de las dos entidades, tanto al principio como al final de las dos entidades, insertando un token especial que rodea la primera entidad ($) y otro token que rodea la segunda entidad (#), además el inicio de la oración es marcado con "[CLS]", por ejemplo: "*[CLS] The $ kitchen $ is the last renovated part of the # house #* ." para la relación *Component-Whole(kitchen, house)*. De los vectores de salida del entrenamiento con BERT, toma los vectores correspondientes de cada entidad, se obtiene el promedio de los vectores para obtener una representación vectorial, después de una operación de activación (*tanh*) se agrega una capa completamente conectada a cada uno de los vectores, incluyendo el vector del primer token. Finalmente concatenan los tres vectores y agregan una capa completamente conectada y una capa *softmax*. Con esta propuesta evaluada sobre el conjunto de datos del SemEval-2010 Task 8, obtienen un Macro F1 de 89.25, lo que los coloca por encima de trabajos similares para la clasificación de relaciones.

3.2.1 Supervisión a distancia

Los métodos supervisados para la extracción de relaciones necesitan datos etiquetados para el entrenamiento, esto implica un gran costo, y por lo tanto, limitados en cantidad. La Supervisión a Distancia (DS) es un método efectivo para generar datos etiquetados a gran escala para la extracción de relaciones por esta razón Mintz et al. (2009) presentan un método de Supervisión a Distancia utilizando Freebase (Bollacker et al., 2008b). El objetivo de este método propuesto es obtener el mayor número de oraciones posibles, siempre y cuando la oración contenga un par de entidades (nombres de personas, organizaciones y lugares) y estas tengan una relación conocida dentro de Freebase. Esto con la intención de obtener un gran cantidad de características (potencialmente ruidosas), para ser usadas en un clasificador logístico multiclase optimizado usando L-BFGS con regularización gaussiana. La evaluación es *held-out*; se realiza de forma automática manteniendo parte de las instancias durante el entrenamiento, y comparando las

Tabla 3.1 Trabajos relacionados sobre la extracción de relaciones semánticas.

Autor	**Algoritmos**	**Evaluación**
Girju et al. (2007)	Patrones léxico-sintácticos. Roles gramaticales. Características basadas en WordNet. Información sintáctica superficial y profunda. Clusters semánticos basados en semejanza de sustantivos. SVM.	Cause-Effect P:69.5%, R:100.0%, F:82.0%. Instrument-Agency P:76.9%, R:78.9%, F:77.9%. Product-Producer P:80.6%, R:87.1%, F:83.7%. Origin-Entity P:70.6%, R:66.7%, F:68.6%. Theme-Tool P:69.0%, R:69.0%, F:69.0%. Part-Whole P:72.4%, R:80.8%, F:76.4%. Content-Container P:93.1%, R:71.1%, F:80.6%.
Ben Abacha and Zweigenbaum (2011)	Enfoque basado en reglas. Patrones léxicos. Reconocimiento de Entidades Médicas. UMLS Metathesaurus.	Precision:74.21%, Boundary Error:27.10%, Type Error Rate:12.23%.
Punuru and Chen (2012)	Tripletas SVO (Sujeto, Verbo, Objeto), VF*ICF, log-likelihood ratio, POS, identificación de Frases Nominales, medida Above Expectation (AE).	VF * ICF: cada uno de los verbos se clasifica manualmente como relevante o no. Precisión de los resultados reducidos (etiquetado de relaciones correcta).
Galicia-Haro and Gelbukh (2014)	TF-IDF. Coheficiente de Correlación de Pearson. C-value, definición de patrones lingüísticos para términos en español. NC-value. Hipónimos e Hiperónimos. Patrones léxico-sintácticos (Hearst). Motor de búsqueda de Google.	Medida de Cimiano para rankear pares (hipónimo, hiperónimo). Precision y Recall. Evaluación de términos de una y múltiples palabras [manuales de lavadoras y 3 evaluadores]. Relaciones semánticas [WordNet, BabelNet, MCR].
Mirrezaei et al. (2015)	Plantillas de oraciones (<subject; relation;object>). NER, POS, resolución de la correferencia, expresiones regulares, convierten fechas a cadenas, heurísticas (full match, synset match, partial match), similitud semántica.	Framework de Bronzi (un enfoque para evaluar los extractores de información medida por verbos automáticamente) fue ampliado para aceptar tripletas mediadas por sustantivos. Además se comparó con un gold estándar. Las métricas usadas: precision, recall, F1 y PMI.
Colhon and Cristea (2016)	Patrones morfo-sintácticos [usados para detectar relaciones anafóricas (referenciales), afectivas (sentimientos o meciones), parentesco (familiares, árbol genealógico) y sociales (jearquía del trabajo o rangos sociales)]. Patrones de dependencia léxica. Árboles de análisis de dependencia.	Evaluan las cuatro relaciones semánticas con las métricas: Precison, Recall, F-measure
Ta and Thi (2016)	Presentan un enfoque híbrido (NLP y estadístico) para extraer relaciones semánticas (sinónimos, hipónimos e hiperónimos, IS-A, PART-OF, MADE-OF, DELIMITED-BY, TAKES-PLACE-IN, ATTRIBUTE-OF, RESULT-OF, AFFECTS)	Las métricas de evaluación son precision, recall y F-Measure.
Barzegar et al. (2018)	NERM (Neural Entity / Relation Model). LSTMN (Long Short Term Memory Networks). Word Embeddings (W2V: Wod2Vector). DNA (Distributional Navigational Algorithm) usa modelos semánticos distributivos como una heurística basada en la relevancia. SemEval 2010 task 8 Corpus. ConceptNet.	Random Model P:0.0220, R:0.0160. Unigram Model P:0.0043, R:0.0270. Bigram Model P:0.2944, R:0.2613. Random Forest P:0.3663, R:0.2476. NERM P:0.3281, R:0.3281.

instancias de relaciones recién descubiertas con las instancias mantenidas. Si bien la evaluación *held-out* sufre de falsos negativos, proporciona una medida aproximada de precisión sin requerir una evaluación humana costosa. En sus resultados obtuvieron 68% de precisión.

Otros trabajos como el de Zeng et al. (2015) abordan los dos problemas que existen en la extracción de relaciones usando el método DS. El primero es la utilización de una base de conocimiento (KB) alineada heurísticamente con los textos, y los resultados son tratados como datos etiquetados. En el segundo, típicamente han aplicado modelos supervisados a características diseñadas de manera elaborada, cuando se obtienen los datos etiquetados a través de supervisión a distancia, además del uso de herramientas de PLN para derivar estas características conduce a la propagación o acumulación de errores. Para resolver estos problemas utilizan la Red Neuronal Convulocional (CNN) con el aprendizaje para múltiples instancias, la entrada de la red recibe embeddings entrenados con Word2Vec.

En el trabajo de Wang et al. (2018) presentan un método de Supervisión a Distancia sin etiqueta (Label-Free DS), cuyo objetivo es evitar el ruido en las etiquetas típicamente introducidas por lo métodos DS. Utilizan la información de la entidad y la ley de traducción en Grafos de Conocimiento (KG) considerando el modelo TransE (Bordes et al., 2013). TransE codifica tripletas ($< h,r,t >$) en un espacio continuo de baja dimensión con la ley de traducción $h + r \approx t$, donde h,r,t describe la entidad inicial (cabeza), la relación y la entidad final (cola) respectivamente. TransE ha demostrado un buen rendimiento en la predicción de la entidad final cuando se le da una entidad inicial y una relación. En este trabajo hacen uso de CNN con una capa de atención, el maxpooling empleado es el presentado por Zeng et al. (2015), los experimentos fueron realizados con el corpus de noticias New York Times (NTY), con la técnica de evaluación *held-out* alcanzando 88% de precisión.

Para la Extracción de Relaciones también se han enfocado en modelos de Aprendizaje de Refuerzo Profundo (RL: Reinforcement Learning) como es descrito por Qin et al. (2018). Diseñan un agente RL cuyo objetivo es aprender a elegir si retiene o elimina la instancia candidata de la supervisión a distancia, en función del cambio de rendimiento del clasificador de relaciones. Emplean *word embedding* y *position embedding* para convertir las oraciones en vectores. El conjunto de datos es NYT y la evaluación es *held-out*, que puede proporcionar una medida aproximada de la capacidad de clasificación sin una evaluación humana costosa.

Para Jiang et al. (2016) la DS adolece de dos puntos primordiales, el primero utiliza el supuesto expresado al menos una vez para la generación de datos etiquetados, que establece que: "*si dos entidades participan en una relación, al menos una oración que menciona estas dos entidades expresará esa relación*", por esta razón solo se selecciona la oración más probable

para cada par de entidades en el entrenamiento y prueba, por lo que argumentan que este supuesto es muy fuerte, y la selección de una sola oración definitivamente perderá la rica información contenida en otras oraciones. El segundo punto trata la Extracción de Relaciones DS como un problema de aprendizaje de etiqueta única, y selecciona para cada par de entidades una etiqueta de relación única, ignorando el hecho de que podría haber múltiples relaciones entre el mismo par de entidades (traslape de relaciones). Proponen una arquitectura de red neuronal convolucional de etiquetas múltiples (MIMLCNN) para abordar los dos problemas descritos anteriormente, definidos en tres pasos: (1) extracción de características a nivel de oración, (2) agrupación máxima de oraciones cruzadas (max-pooling) y (3) modelado de relación de etiquetas múltiples. El conjunto de datos utilizado es NYT10 (Riedel et al., 2010) alineado con relaciones de Freebase. La evaluación fue held-out, se comparan con otros trabajos y muestran sus resultados con métricas de la curva *Precision-Recall* y P@N. Los resultados que obtienen superan trabajos previos del estado del arte.

3.2.2 Enfoque no supervisado

Un enfoque que consiste en la agrupación (clustering) basada en el contexto de pares de entidades, es el propuesto por Hasegawa et al. (2004), establecen que dos entidades nombradas se consideran coexistentes (co-ocurrencia), si aparecen dentro de la misma oración y están separadas por, como máximo, *cinco* palabras intermedias. El proceso consiste en etiquetar las entidades nombradas con un etiquetador, para después recopilar todas las instancias entre el par *"A"* y *"B"* que ocurren dentro de una cierta distancia una de la otra. Las palabras del contexto que interviene entre *"A"* y *"B"* se pasan a su forma básica (stemmed) y se almacenan. El mismo proceso se realiza entre el par de entidades *"C"* y *"D"*, si el conjunto de contextos de *"A"* y *"B"* y aquellos de *"C"* y *"D"* son similares, entonces los dos pares de entidades son colocados en el mismo grupo (clúster). Para calcular la similitud del contexto entre pares de entidades, hacen uso de un vector formado por todas las palabras que intervienen en todas las co-ocurrencias del par de entidades, y son pesadas usando *tf*idf*, emplean la similitud coseno como métrica. En el siguiente paso construyen grupos (clusters) de pares de entidades basados en su similitud, los grupos consistieron de al menos 30 pares de entidades. El paso final consiste en etiquetar los grupos resultantes, para ello las palabras comunes y más frecuentes en un grupo (clúster) se convierten en la etiqueta del grupo. Los experimentos fueron realizados sobre noticias periodísticas del New York Times de 1995, la evaluación se llevo a cabo sobre

los dominios de pares de entidades (*PERSON-GPE*) y (*COMPANY-COMPANY*), las métricas empleadas para evaluar los clusters fueron *precision*, *recall* y *F1-measure*.

Un sistema llamado URES (Unsupervised Relation Extraction System) es presentado por Feldman and Rosenfeld (2006) que aprende patrones de texto sin etiquetar, utilizando descripciones breves de las relaciones objetivo y sus atributos. El sistema necesita como entrada la definición de relaciones objetivo, compuestas por el esquema de la relación y algunas palabras clave que permiten recopilar frases relevantes. Las palabras clave se utilizan para recopilar oraciones de la Web, y para crear instancias de los patrones genéricos para la generación de semillas, estas semillas (10 instancias para cada relación objetivo) se pasan como entrada al modulo de aprendizaje de patrones, que utiliza las semillas para aprender patrones probables de ocurrencias de relaciones. Opcionalmente, estas instancias pueden ser filtradas por un NER. En la etapa final, se clasifican la lista de todas las instancias extraídas. En lugar de un clasificador binario, hacen uso de una función de confianza con valor real, mapeando el conjunto de instancias extraídas en el segmento [0, 1]. Utilizan un modelo de regresión para estimar la función de confianza para cada coincidencia de patrones. En sus experimentos utilizan cinco tipos de relaciones *"Acquisition, Merger, CEO_Of, MayorOf e InventorOf"*. Para la evaluación se empleó la *precision* sobre las relaciones a identificar.

En el trabajo de Yan et al. (2009) proponen un método que agrupa pares de conceptos en muchos grupos (clusters) basados en la similitud de sus contextos. Los contextos son colocados como patrones de dos tipos: patrones dependientes del análisis de dependencia (dependency parsing) de las oraciones de documentos de Wikipedia, y patrones superficiales generados de información altamente redundante de la Web. Los pares de conceptos son el *título del artículo* de Wikipedia y *conceptos relacionados* que aparecen en los vínculos. Hacen uso de un motor de búsqueda (Google) para obtener términos relacionales entre los conceptos: 1) utilizan una ventana circundante al par de conceptos identificando los verbos y sustantivos para seleccionar los términos, después de la recolección se combinan todos los términos y se clasifican (ranking) usando un algoritmo basado en entropía, y seleccionan una palabra clave de cada par de conceptos. 2) definen patrones superficiales empleando los dos conceptos y la palabra clave. Los patrones de dependencia son definidos como subrutas de la ruta de dependencia más corta entre un par de conceptos. Para la clasificación hacen uso del algoritmo K-means para agrupar el par de conceptos en dos fases, en la primera se agrupan pares de conceptos en grupos con buena precisión usando los patrones de dependencia, para después mejorar la cobertura de los grupos empleando patrones superficiales. Los experimentos se realizaron sobre dos categorías de Wikipedia, *"Directores ejecutivos estadounidenses"* y *"Empresas"*. Para medir la extracción

de las relaciones emplearon las medidas de *precision* y *coverage* (evalúa todos los conceptos pares extraídos de forma correcta).

Un framework que integra una base de conocimiento y enfoques de minería de textos es desarrollado por Alicante et al. (2016). El framework está dividido en tres etapas, en la primera se dedica a la identificación y clasificación de entidades médicas, en la segunda aplican un algoritmo de agrupamiento (spherical K-means), y la tercera parte que se encarga de asociar una etiqueta a cada grupo (clúster). EL pre-procesamiento de los documentos de texto incluye tokenización, dividir el texto en oraciones, PoS tagging ya que la etiqueta Pos se asocia a cada token, de la misma forma que el lema de cada token. Para la identificación de entidades se define un conjunto de patrones basados en etiquetas PoS, se buscan las coincidencias en diccionarios especializados, además utilizan el esquema de etiquetado IOB (Inside, Out, Begining) para las entidades extraídas. En la segunda etapa, cada par de entidades que se encuentran en la misma oración es identificada como una potencial relación, para después aplicar el algoritmo de agrupamiento. Se construye un vector de características (*"tipo de la entidad"*, *"unigramas, bigramas, y trigramas de palabras"* así como *"características barrera"*) asociado al par de entidades. Realizan pruebas con diferentes medidas de similitud, como son *Manhattan*, *Binary* y *Cosine*. El algoritmo de K-means fue fijado con un máximo de 10 iteraciones, de la misma forma que el número de clusters fue 10. En otros experimento utilizaron el algoritmo spherical K-means con un máximo de 12 iteraciones. Para la evaluación adoptaron la métrica *silhouette*, que tiene en cuenta dos aspectos relevantes para juzgar una agrupación: el primer aspecto da una idea de cuán compacto es un clúster, mientras que el segundo evalúa qué tan diferente del otro es.

Un framework para la extracción no supervisada de conceptos clínicos a través de la composición semántica es presentado por Tulkens et al. (2019). Utilizan embeddings de FastText, para las palabras OOV (Out of Vocabulary) generan embeddings con Word2vec. Emplean la composición sobre palabras sin tener en cuenta las dependencias sintácticas. En primera instancia extraen todas las frases nominales (*Noun Phrases*) con cTakes parser (Savova et al., 2010) de los documentos como candidatos, después con el conjunto de candidatos crean vectores usando la composición aritmética sobre las representaciones vectoriales (embeddings) de las palabras en cada frase. Para cada candidato extraen todas las palabras dentro de la *frase*, y una ventana de *n* palabras de contexto a cada lado de la *frase* y la *frase* misma, obteniendo tres secuencias separadas de palabras, se le asigna un peso a cada palabra en ambas ventanas de contexto, usando el recíproco de su distancia a la palabra de enfoque, aplican la media (*element-wise mean*) a cada secuencia, y a su vez aplican la media a las tres secuencias para obtener un

solo vector. Además de extraer *frases* también se extraen *conceptos* del meta-tesauros UMLS (Unified Medical Language System[1]), para cada *concepto* extraído aplican las mismas métricas *element-wise mean* (con excepción del recíproco). A cada *concepto* se le asigna una etiqueta del conjunto de etiquetas del corpus I2b2-2010[2]. A los dos grupos de vectores de *frases* y *conceptos* les calculan la similitud coseno, así el *concepto* del conjunto de *conceptos* con la mayor similitud es entonces asignado a una *frase*. En sus experimentos evalúan los vectores de *frases* comparando con las etiquetas estándar *"gold"* del corpus I2b2-2010, las métricas usadas fueron *precision*, *recall* y *F1-score* empleando el clasificador kNN (*k Nearest Neighbors*), para cada *frase* se verificó si se superpone con alguno de los fragmentos (*chunks*) estándar *"gold"* del corpus i2b2.

La extracción de relaciones es una de las tareas ampliamente usada para la adquisición de conocimiento, para tareas como Question & Answering, construir bases de conocimiento, resúmenes automáticos, etc. La mayoría de trabajos se enfoca en obtener relaciones básicas definidas en el estado del arte, como son *is a* y *part of*, así como relaciones en el dominio médico. Estos trabajos están presentes en métodos supervisados, donde comparan su rendimiento con otros conjuntos de datos. Sin embargo, pocos trabajos abordan la identificación y extracción de relaciones no definidas, es decir, aquellas que se descubren de forma intrínseca en la oración, para dar un significado a un par de entidades nombradas.

3.3 Inferencia para obtener nuevos hechos

El desarrollo de un sistema para escalar información textual de la Web es propuesto por Schoenmackers et al. (2008). Realizan inferencias probabilísticas escalables sobre afirmaciones básicas extraídas de la Web. Como entrada usan una consulta conjuntiva, un conjunto de reglas de inferencia expresadas como cláusulas Horn, y conjuntos grandes de afirmaciones base extraídas de la Web, WordNet y otras bases de conocimiento. Establecen seis reglas independientes del dominio para preservar el significado para las afirmaciones: observadas en el corpus, sinónimos, generalización y transitividad de los merónimos de las partes de ellos. El sistema realiza un encadenamiento hacia atrás desde la consulta, utilizando las reglas de inferencia para construir un bosque de árboles de prueba a partir de las afirmaciones básicas. Para ello, emplean redes de Markov y evalúan los resultados usando inferencia probabilística aproximada.

[1]https://www.nlm.nih.gov/research/umls/index.html
[2]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3168320/

Las Redes Lógicas de Markov (MLN, por sus siglas en inglés) son usadas por Jiang et al. (2012) para eliminar el ruido en la construcción de bases de conocimiento. El método que proponen realiza inferencias probabilísticas conjuntas sobre hechos candidatos. Emplean una representación compacta de nombres de categorías y relaciones en predicados de segundo orden, además hacen uso de restricciones ontológicas en MLN. Realizan diversos experimentos y se comparan con el proyecto Never-Ending Language Learner (NELL) (Carlson et al., 2010), alcanzando un F1-measure de 0.836 en su mejor resultado.

Para la construcción de una base de conocimiento con entidades, hechos y reglas extraídos en una escala web, Chen and Wang (2013) presentan un enfoque probabilístico usando MLN, que son una extensión de la lógica de primer orden que aumenta un peso a cada cláusula. Para obtener escalabilidad diseñan un modelo relacional con todos los hechos y reglas en una base de datos. Hacen uso de un conjunto de reglas Horn extraídas previamente por Schoenmackers et al. (2010). En tiempo de ejecución construyen una consulta SQL para cada regla individual, identificando seis patrones de reglas en el conjunto de reglas. El resultado es un grafo (red de Markov) que codifica la distribución de probabilidad, para mejorar el algoritmo hace uso del motor de inferencia GraphLab para realizar la inferencia. En sus experimentos se comparan con otras bases de conocimiento con el tiempo de ejecución y escalabilidad.

En el trabajo de Pujara et al. (2013) emplean la inferencia del framework de Lógica Suave Probabilística (PSL, por sus siglas en inglés) para la construcción de un grafo de conocimiento, plantean como hechos a las entidades, sus etiquetas y las relaciones entre ellas. El objetivo es eliminar el ruido, inferir la información que falta y determinar que hechos candidatos deben incluirse. Al combinar las reglas lógicas con átomos (hechos candidatos), los resultados que obtienen de la inferencia PSL es la interpretación más probable para la elección de hechos. Hacen uso del conjunto de datos MusicBrainz para sus experimentos y se comparan con el proyecto NELL (Carlson et al., 2010). En sus resultados superan con 0.823 de F1-measure a lo realizado por NELL.

Un conjunto de reglas de inferencia genéricas son establecidas por Bast and Haussmann (2014), para el proceso de extracción en un sistema de Extracción de Información Abierto (OIE, por sus siglas en inglés). El conjunto de reglas es genérico e independiente de la superficie textual exacta de una relación. El proceso consiste en identificar el sujeto, predicado y objeto de la tripleta en la oración, después se clasifica el predicado en una de las relaciones semánticas establecidas. Dependiendo de la relación asignada previamente, se aplica un conjunto de reglas de inferencia para inferir nuevas tripletas. Como paso final, se eliminan aquellas tripletas que no presentan suficiente información, dependiendo de si se usaron y cómo se usaron para derivar

nuevas tripletas. El conjunto de datos usado comprende doscientas oraciones obtenidas de forma aleatoria de Wikipedia. Evalúan la exactitud y la informatividad de las tripletas, asignando de forma manual una etiqueta (*yes* o *no*) para cada tipo de evaluación. Las métricas aplicadas son la precisión para calcular tripletas correctas y recall para obtener el número de tripletas correctas.

Un método de inferencia para extraer hechos relevantes es presentado por Sena et al. (2017). Su enfoque se encuentra en obtener entidades nombradas y sus relaciones en idioma portugués. Proponen un conjunto de oraciones con relaciones transitivas y simétricas anotadas manualmente, y por medio de la inferencia de estas incrementar la cantidad de hechos sin determinar el tipo de relación previamente. Para la inferencia usan Máquinas de Vector de Soporte (SVM, por sus siglas en inglés) obteniendo una precisión de 0.83. Para sus experimentos hacen una comparación con ReVerb (Fader et al., 2011) y DepOE (Gamallo et al., 2012) en base a la cantidad de hechos extraídos y la precisión. En sus resultados su método propuesto obtuvo una mayor cantidad de hechos válidos, y una precisión de 0.82, 0.92, 0.70 respectivamente.

La Tabla 3.2 presenta los motores de inferencia o razonadores empleados para deducir nuevos hechos o asociaciones de información existente presentados por (Singh and Karwayun, 2010), y menciona que un razonador semántico, motor de razonamiento, motor de reglas, o simplemente un razonador, es una pieza de software capaz de inferir consecuencias lógicas de un conjunto de hechos o axiomas afirmados. Las reglas de inferencia se especifican comúnmente mediante un lenguaje de ontología y, a menudo, un lenguaje de descripción.

Los trabajos que abordan la inferencia de nuevos hechos a partir de un conjunto existente emplean diferentes métodos. Las redes lógicas de Markov y la Programación Lógica Suave son los principales algoritmos para abordar la inferencia, los hechos son tripletas que están compuestas por *sujeto, predicado* y *objeto* para construir grandes bases de conocimiento. Además, en sus evaluaciones se comparan con trabajos del estado del arte, como bases de conocimiento construidas sin un método de inferencia. Sin embargo, las relaciones presentes en las tripletas son definidas y se enfocan en el idioma inglés. Solo un trabajo emplea un conjunto pequeño de reglas lógicas genéricas para la inferencia de nuevos hechos.

3.4 Grafo de Conocimiento

Paulheim (2017b) menciona que el término de grafo de conocimiento (KG: Knowledge Graph, en inglés) fue popularizado por Google en el año 2012, refiriéndose a su uso del conocimiento semántico en la búsqueda web ("*cosas, no cadenas*"), y recientemente también se utiliza para

Tabla 3.2 Motores de inferencia para deducir nuevos hechos o asociaciones de información existente.

Motor	**Lenguaje soportado**	**Observaciones**
Jess	Reglas declarativas	Es pequeño, ligero y uno de los motores más rápidos disponibles.
Hoolet	Web Ontology Language - Description Logic, Resource Description Framework	Se implementa utilizando la API WonderWeb OWL para analizar y procesar OWL, y la lógica de Vampire para fines de razonamiento.
Pellet	Web Ontology Language (OWL) - Description Logic	Proporciona servicios de razonamiento para ontologías OWL. Es un componente central de las aplicaciones de gestión de datos basadas en ontologías.
SHER	Web Ontology Language	No hace ninguna inferencia sobre la carga. Tolera inconsistencias lógicas en los datos y señala rápidamente esas inconsistencias.
KAON2	Web Ontology Language - Description Logic, Semantic Web Rule Language y Frame Logic	Admite la respuesta a consultas conjuntivas expresadas en SPARQL Protocol and RDF Query Language (SPARQL) y RDF.
RacerPro	Web Ontology Language - Description Logic, Web Ontology Language Lite	Proporciona razonamiento algebraico, restricciones mínimas / máximas sobre los enteros, ecuaciones polinomiales lineales sobre los reales o cardinales con relaciones de orden, igualdades y desigualdades de cadenas.
Jena	Web Ontology Language - Description Logic	Proporciona una API para extraer datos y escribir en grafos RDF. Soporta consultas en lenguaje SPARQL. Habilidad para actualizar y borrar datos RDF.
FaCT	Description Logic	Incluye los razonadores SHF y SHIQ. Posee una lógica expresiva, es compatible con el razonamiento con bases de conocimiento arbitrarias y posee una arquitectura cliente-servidor basada en Common Object Request Broker Architecture (CORBA).
FaCT++	Web Ontology Language - Description Logic	Está implementado en C ++ y muestra un rendimiento excepcional en ontologías expresivas.
SweetRules	Rule Markup/Modeling Language (RuleML), Semantic Web Rule Language, Web Ontology Language	Incluyen traducción e interoperabilidad que preservan la semántica entre una variedad de lenguajes de reglas y ontologías, como: XSB Prolog, reglas de producción de Jess e IBM CommomRules.
OWLIM	Web Ontology Language Lite, RDFS	Es un repositorio semántico escalable con conjuntos de reglas personalizables e implementado como una capa de almacenamiento e inferencia para el marco Sesame OpenRDF.
F-OWL	Web Ontology Language	Extrae triples RDF y se convierten al formato del marco de F-OWL, y se introducen en el motor F-OWL. Utiliza reglas de flora definidas en el lenguaje flora-2 para verificar la consistencia de la ontología y extraer el conocimiento oculto a través de la resolución.
BaseVISor	Rule Markup/Modeling Language (RuleML), Web Ontology Language	Es un motor de inferencia de encadenamiento hacia adelante basado en una red Rete optimizada para el procesamiento de triples RDF.

referirse a las bases de conocimiento de la web semántica como DBpedia[3] o YAGO[4]. Desde una perspectiva más amplia, cualquier representación de algún conocimiento basado en grafos podría considerarse un KG (esto incluiría cualquier tipo de conjunto de datos RDF, así como las ontologías lógicas de descripción).

Pan et al. (2017) argumenta que un KG es una forma gráfica para representar el conocimiento sobre entidades y las relaciones entre las entidades. KG no es realmente un concepto completamente nuevo. Las ideas básicas fueron propuestas en el formalismo de representación del conocimiento, el KG es heredado de formalismos clásicos de representación del conocimiento, en particular de las redes semánticas. En el 2004 los investigadores de la representación del conocimiento y razonamiento, abordaron algunos problemas conocidos como las redes semánticas al estandarizar la versión moderna de la red semántica o grafo RDF (Resource Description Framework, en inglés), que utiliza OWL (Web Ontology Language, en inglés) para definir su esquema. OWL ofrece diferentes niveles de poder expresivo a los modeladores para definir esquemas livianos o esquemas pesados para sus grafos de conocimiento. En general, un grafo de conocimiento puede verse como una ontología con una vista centrada en la entidad, que consiste en un conjunto de entidades tipadas interconectadas y sus atributos, así como algunos axiomas de esquema para definir el vocabulario (terminologías) utilizado en el KG. La unidad básica de un KG es una entidad singular. Cada entidad puede tener varios atributos; por ejemplo, los atributos de una persona incluyen nombre, fecha de nacimiento e información familiar. Además, las entidades están conectadas entre sí por relaciones; por ejemplo, algunas entidades tipo *Persona* como algunas entidades tipo *Producto*, que están a la venta en algunas entidades de *Tienda*.

Singhal (2012b) menciona que el KG de Google permite buscar cosas, personas o lugares de los que se tenga el conocimiento, por ejemplo: lugares muy conocidos, celebridades, ciudades, equipos deportivos, edificios, características geográficas, películas, objetos celestiales, obras de arte y más. Además de poder obtener al instante información relevante para su consulta. Este es un primer paso crítico hacia la construcción de la próxima generación de búsquedas, que aprovecha la inteligencia colectiva de la web y entiende el mundo un poco más como lo hace la gente.

La construcción de grafos de conocimiento de forma automática es presentado por Rospocher et al. (2016), el grafo de conocimiento está centrado en eventos sobre noticias

[3]http://wiki.dbpedia.org Acceso:Mayo-2018

[4]https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/ Acceso:Mayo-2018

en cuatro idiomas (inglés, español, italiano y holandés). Los datos se encuentran en documentos no estructurados, para obtener los eventos hacen uso de una secuencia lineal especializada de módulos con técnicas de PLN (Reconocimiento de entidades, Desambiguación de entidades, Minería de opinión, Analizador sintáctico, Etiquetador de roles semánticos, Resolución de correferencia, Extracción de relaciones y Detección de hechos). El grafo de conocimiento es vinculado con otras bases de conocimiento, algunas son DBpedia, PropBank[5], VerbNet[6], FrameNet[7] y WordNet. Los datos son representados usando GAF (Grounded Annotation Framework) y SEM[8] (Simple Event Model) además de RDF (Resource Description Framework), RDFS (RDF Schema) y OWL (Web Ontology Language). Para su estudio definen cuatro grafos de conocimiento (casas de estudio): WikiNews, Fifa WorldCup, Cars y Airbus Corpus. Para la evaluación de la calidad de la construccion de los grafos, confiaron en el juicio humano para evaluar las tripletas con muestras aleatorias.

Para evaluar la "*exactitud*" (*accuracy*) de un grafo de conocimiento Gao et al. (2019) proporcionan un framework de evaluación iterativo que garantiza una estimación de "*exactitud*" de alta calidad con una fuerte consistencia estadística con ayuda de anotadores. Donde se puede especificar un error vinculado en el resultado de la estimación, y el framework lo muestra y estima iterativamente. Se detiene tan pronto como el error de estimación es inferior al umbral requerido sin sobremuestreo y evaluaciones manuales innecesarias. Además aplican el muestreo de clusters con una teoría de probabilidad desigual que permite evaluaciones manuales eficientes. El tamaño óptimo de la unidad de muestreo en el grafo es derivado cuantitativamente al asociarlo con el costo de evaluación aproximado. Los clusters se forman con el *sujeto* de la tripleta, con el objetivo de reducir el costo de evaluación, al evitar identificar el *sujeto* en múltiples ocasiones.

Yan et al. (2018b) han realizado una recopilación de los principales grafos de conocimiento de la literatura, la Tabla 3.3 muestra dicha comparación. En la Tabla 3.3 se observa el surgimiento de las bases de conocimiento a través de los años, así como los proyectos de donde han surgido, el contenido del que disponen y el tipo de almacenamiento.

[5]https://propbank.github.io/
[6]https://verbs.colorado.edu/verbnet/
[7]https://framenet.icsi.berkeley.edu/fndrupal/
[8]https://semanticweb.cs.vu.nl/2009/11/sem/

Tabla 3.3 Principales grafos de conocimiento. Clasificación hecha por (Yan et al., 2018c)

Proyecto	**Año**	**Contenido**	**Objetos de datos**	**Tipo almacenamiento**
CYC	1984	120K+ concepts	Ontology, Facts	CycL
WordNet	1985	117K synsets	Ontology, Facts	OWL/RDF
FrameNet	1997	10 000 word senses and 170 000 annotated sentences	Ontology, Facts	XML/OWL
ConceptNet	1999	1.6M assertions	700K Facts	JSON
HowNet	2000	800 sememes	6,000 Chinese characters	Named sememes
KnowItAll	2005	54 753 facts	XML	Pattern / Wrapper / Rule
TextRunner	2007	500M assertions	Facts	RDF
Freebase	2007	22M+	Ontology, Facts	RDF
DBpedia	2007	320 classes, 0.7M Wiki types, 3.6M entities, 247M triples	Ontology, Facts	RDF
YAGO	2007	10M+ entities, 120M+ facts	Ontology, Facts	RDF
Kylin	2008	Classified documents and sentences	Ontology, Facts	RDF
OpenCalais	2008	Semantic metadata	N/A	OWL/RDF
Satori	2009	400M entities	Ontology, Facts	RDF
NELL	2010	123 categories and 55 relations	Ontology, Facts	TSV File
Probase	2011	2.7M concepts	Is-A pair	RDF
EntityCube/Renlifang(C)	2011	Entity-Relation Graph	Facts	RDF
Google Knowledge Graph(M)	2012	570M+ objects, 18B facts	Ontology, Facts	RDF Triple
Sogou Zhilifang(C)	2012	100M+ entities, 1M+ relations	Ontology, Facts	N/A
Baidu Zhixin(C)	2012	N/A	Ontology, Facts	N/A
Facebook Graph Search	2013	User's network of friends, Bing's search engine	Facts	GraphML

Los métodos para la construcción de bases de conocimiento o grafos de conocimiento, se enfocan en la construcción de ellos en una escala web. Emplean el lenguaje XML basado en RDF según los lineamientos de la W3C. Están constituidos por entidades o conceptos que son representados en la estructura como nodos, y las relaciones que existen entre ellos son las aristas. También poseen diferentes atributos para identificar el tipo o la clase a la que pertenece dicho nodo, además en sus atributos presentan vínculos hacia otros grafos de conocimiento.

Capítulo 4

Adquisición y representación automática de conocimiento

La adquisición de conocimiento de documentos no estructurados, esencialmente de documentos de noticias web, se realiza a través del reconocimiento y clasificación de entidades nombradas, a su vez estas entidades son utilizadas para la extracción de relaciones entre entidades. Las entidades nombradas y las relaciones extraídas se emplean para construir una base de hechos, manualmente se definen *"reglas genéricas"*, esta información es usada para generar nuevos hechos utilizando un motor de inferencia lógica. La representación de conocimiento es establecida en un grafo de conocimiento, que se construye con vértices (entidades nombradas) y aristas (relaciones) entre ellos.

4.1 Reconocimiento y clasificación de entidades nombradas

La metodología propuesta para el reconocimiento y clasificación de entidades nombradas se describe en este capítulo, así como los conjuntos de datos utilizados para los experimentos.

4.1.1 Recolección de Datos

Para la recolección de noticias se utiliza Scrapy[1] que es una biblioteca escrita en Python para la extracción de datos en sitios web. Específicamente se utilizan *Scrapy* y *Spider* (araña o rastreador), la primera se utiliza para recuperar URLs de noticias en fuentes RSS y la segunda

[1]https://scrapy.org/

para extraer la información de los documentos HTML de las URLs previamente extraídas como se muestra en la Figura 4.1. Se cuenta con al menos de una fuente de noticias RSS por cada estado de la República Mexicana con un total de 258 periódicos digitales (fuentes RSS) de los que diariamente se extraen datos para la construcción del corpus de noticias.

Crawler

El conjunto de datos ha sido recopilado a partir de documentos web de los principales periódicos de la República Mexicana, por medio de un crawler y una base de datos.

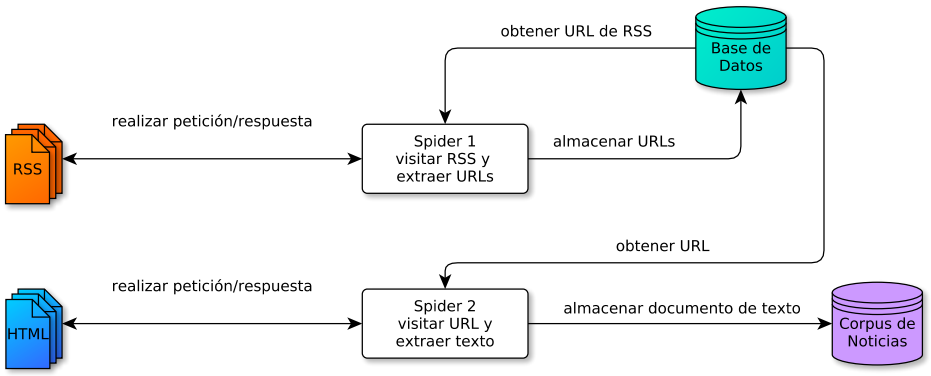

Fig. 4.1 Estructura general del crawler de noticias.

Los documentos de noticias utilizados están enfocados en el dominio de política y cuenta con alrededor de 36 mil documentos. La Figura 4.2 representa la cantidad de noticias descargadas a través de los meses, iniciando el día 11 de Julio de 2018 con 85 noticias descargadas, la fecha final de descarga fue el día 10 de Junio de 2019.

4.1.2 Conjunto de datos

Se utilizan dos corpus en idioma español para realizar experimentos sobre la tarea NERC (Reconocimiento y Clasificación de Entidades Nombradas), el corpus de Carreras et al. (2002) en idioma español para la competencia del CoNLL-2002 etiquetado bajo el esquema IOB (Inside, Begining, Output) para cuatro entidades. El segundo conjunto de datos está construido por documentos de noticias políticas de la República Mexicana, en el cual un anotador etiquetó 250 documentos manualmente para diecisiete entidades nombradas usando el esquema IOBES (Inside, Output, Begining, End, Single).

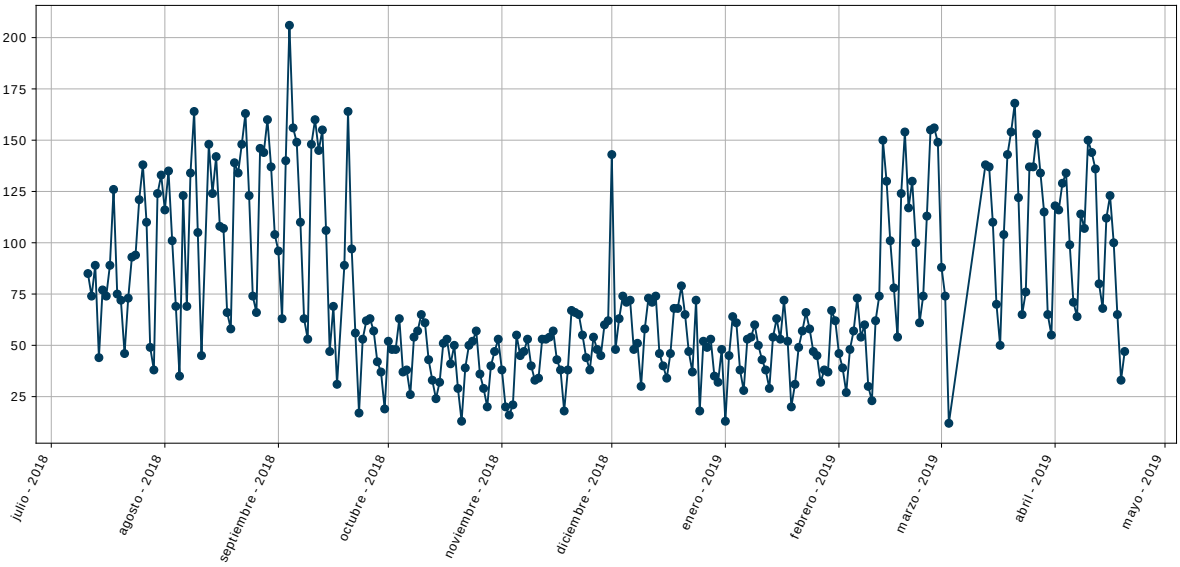

Fig. 4.2 Noticias políticas descargadas.

Corpus CoNLL-2002

El corpus CoNLL-2002 (Conference on Computational Natural Language Learning 2002)[2] se describe en la Tabla 4.1, donde el símbolo (∗) indica la longitud original (tokens) para cada oración, y el símbolo (†) indica que todas las oraciones se ajustaron a la longitud de 50 tokens. De este corpus se utilizó el conjunto de entrenamiento (train), el conjunto de prueba (test) A y B, así como la unión de todos (ensemble).

Tabla 4.1 Descripción del corpus CoNLL-2002.

	Train	**Test-A**	**Test-B**	**Ensemble**
Oraciones ∗	8,323	1,915	1,517	11,755
Oraciones †	9,947	2,177	1,848	13,972
Tokens	26,099	9,646	9,086	31,405
Etiquetas individuales	8	8	8	8
Esquema	IOB (Inside/Output/Begining)			

El corpus contiene cuatro entidades nombradas anotadas, la clase MISC (misceláneas) como se muestra en la Figura 4.3 contiene 7,380 entidades, de las cuales 2,957 marcadas con la etra “B” y con la letra “I” 4,423 entidades. La clase LOC (lugares) contiene 6,981

[2]https://www.clips.uantwerpen.be/conll2002/ner/

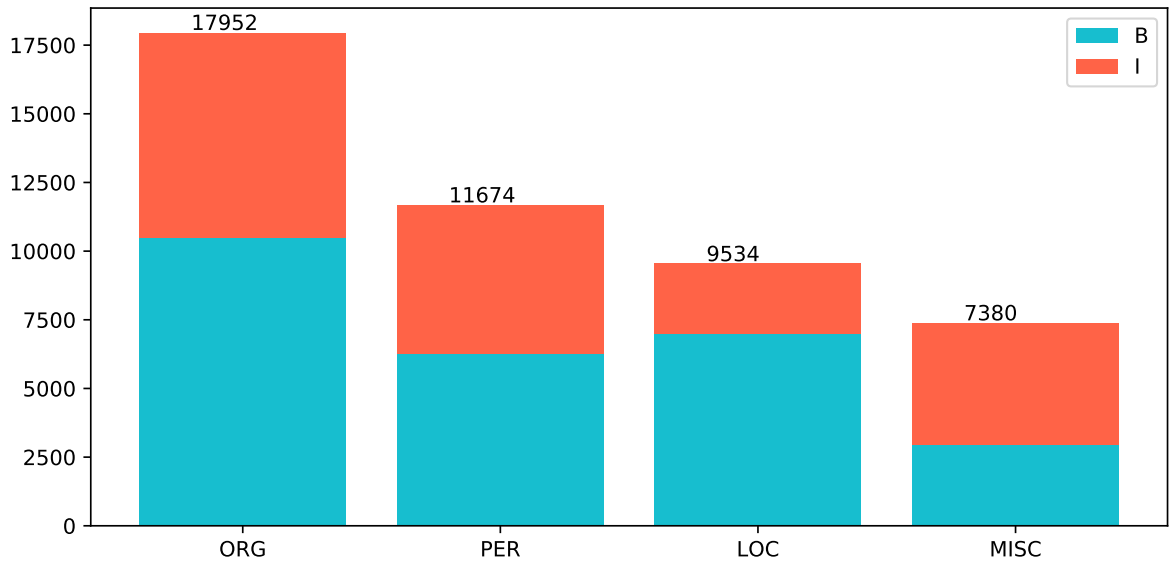

Fig. 4.3 Distribución de clases del corpus CoNLL-2002 bajo el esquema IOB.

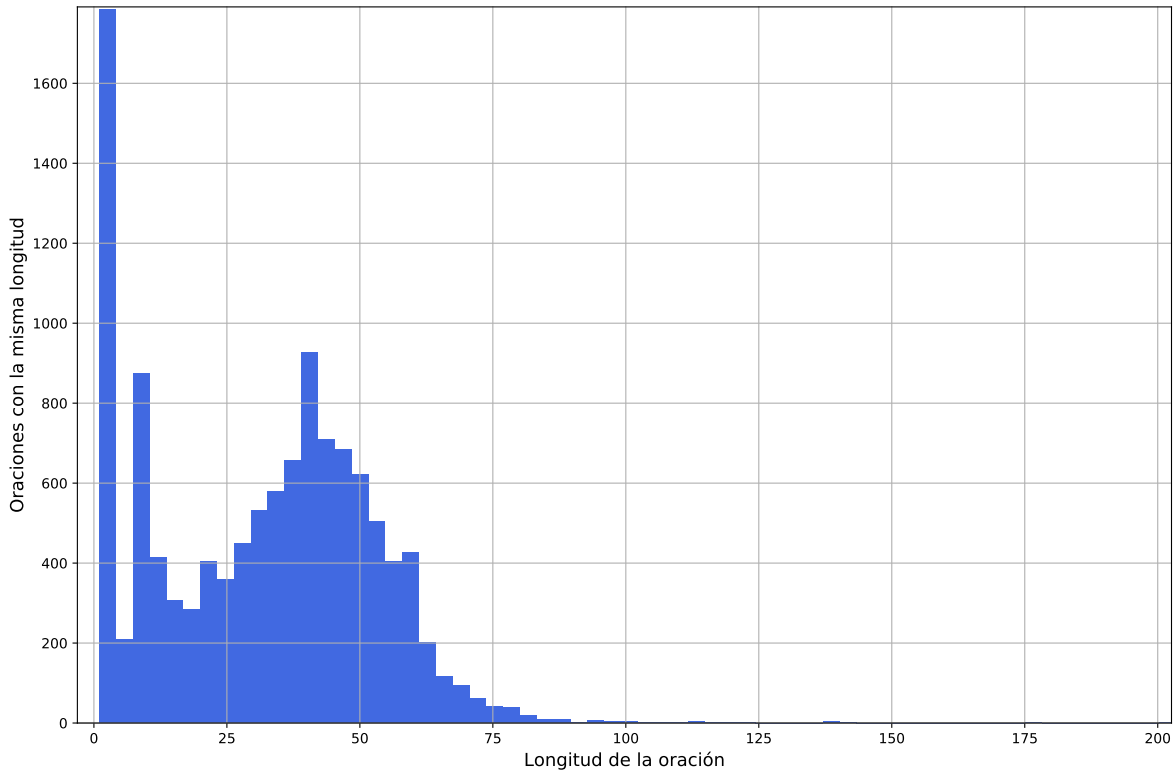

Fig. 4.4 Histograma de oraciones del corpus CoNLL-2002.

entidades marcadas con la letra "B" y 2,553 con la letra "I" con un total de 9,534 entidades. Para la clase PER (nombres de personas) se tienen 6,278 y 5,396 marcadas con la letra "B" e "I" respectivamente sumando 11,674 entidades. Finalmente la clase ORG (organizaciones) cuenta con 17,952 entidades conformadas por 10,490 y 74,62 marcadas con la letra "B" e "I" respectivamente.

En la Figura 4.4 se observa las sentencias contenidas en el corpus CoNLL-2002, el *eje x* muestra la longitud (cantidad de tokens) de las oraciones, en cambio el *eje y* enumera oraciones con la misma longitud. La visualización de las oraciones es usada para definir la longitud de las oraciones en los experimentos con modelos de redes neuronales, ya que estos modelos requieren oraciones (vectores) con la misma longitud para ser procesados.

Corpus de noticias mexicanas

Este corpus fue anotado de forma manual por un anotador, en el dominio de noticias políticas mexicanas (Mx-news). La Tabla 4.2 presenta el contenido de los datos anotados, donde el símbolo ($*$) indica la longitud (tokens) original de las oraciones, el símbolo (†) indica el ajuste de la longitud a 50 tokens de las oraciones.

Tabla 4.2 Descripción del corpus Mx-news.

	Split I	**Split II**	**Split III**	**Ensemble**
Oraciones $*$	1,295	1,295	1,297	3,888
Oraciones †	1,666	1,677	1,661	5,004
Tokens	7,628	7,726	7,664	13,273
Etiquetas individuales	63	63	63	65
Esquema	IOBES (Inside/Output/Begining/End/Single)			

La Figura 4.5 describe la distribución del etiquetado IOBES sobre las 17 clases, iniciando con la clase más densa hasta la más dispersa. También describe cada una de las etiquetas IOBES para cada clase, donde se puede observar que algunas clases no contienen las cuatro etiquetas IBES, como son PRC (porcentajes) que carece de etiquetas "S", DEM (demográfica) únicamente cuenta con una etiqueta "B" y "E" y carece de la etiqueta "I", en cuanto AGE (edad) carece de la etiqueta "S".

La Figura 4.6 describe la distribución de las oraciones original del corpus Mx-news, donde se puede apreciar una gran número de oraciones en el rango de 0 a 50 tokens de longitud. Por esta razón las oraciones mayores fueron recortadas a 50 tokens, el motivo de este recorte es

debido al uso de redes neuronales ya que requieren vectores (oraciones) de entrada de una misma longitud.

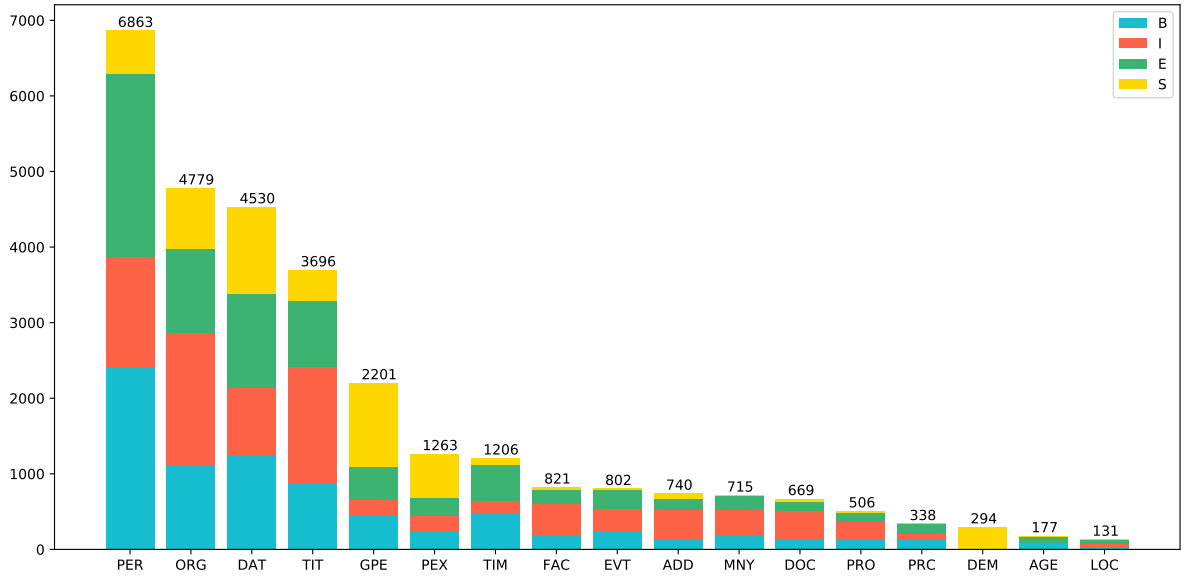

Fig. 4.5 Distribución de clases del corpus Mx-news bajo el esquema IOBES.

Finalmente, la Tabla 4.3 muestra las 17 clases usadas en el corpus Mx-news, están listadas de la la clase más densamente anotada (PER) hasta la clase mas dispersa (LOC). También se describen el significado de cada una de las clases y los conceptos sobre los que se realizaron las anotaciones.

4.1.3 Reconocimiento y Extracción de Entidades Nombradas

En esta sección se presentan los modelos sobre el reconocimiento y clasificación de entidades nombradas (NERC: Named Entity Recognition and Classification). Los modelos son tomados del estado del arte, el modelo probabilístico de campos aleatorios condicionales (CRF: Conditional Random Fields) y modelos basados en aprendizaje profundo (DL: Deep Learning).

Modelo de Campos Aleatorios Condicionales

Los modelos de Campos Aleatorios Condicionales (CRF) son frecuentemente usados en problemas NER, debido a que están diseñados para considerar el contexto en el proceso. Los experimentos realizados se describen a continuación.

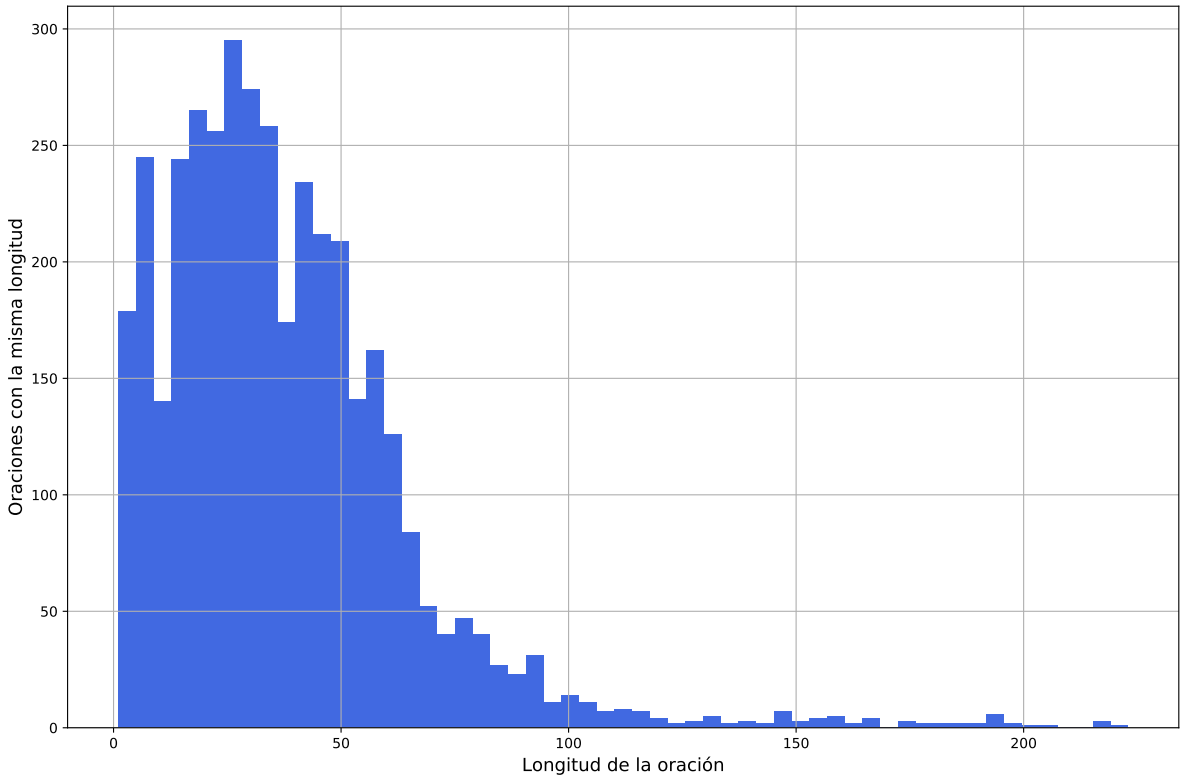

Fig. 4.6 Histograma de oraciones del corpus Mx-news.

Tabla 4.3 Clases usadas en el etiquetado manual del corpus Mx-news.

No.	Clase	Descripción
1	PER	Nombres de personas, alias y abreviaciones.
2	ORG	Organizaciones, instituciones.
3	DAT	Diferentes formatos de fechas.
4	TIT	Título o posición de personas dentro de una organización.
5	GPE	Nombres de países, estados, ciudades, municipios.
6	PEX	Partidos políticos, alias y abreviaciones.
7	TIM	Expresiones de tiempo.
8	FAC	Nombres de instalaciones.
9	EVT	Nombres de eventos.
10	ADD	Expresiones de direcciones, URLs y usuarios de Twitter.
11	MNY	Cantidades monetarias.
12	DOC	Documentos, leyes, reglamentos.
13	PRO	Nombres de productos, marcas, aplicaciones.
14	PRC	Expresiones de porcentaje.
15	DEM	Origen geográfico o racial de personas.
16	AGE	Edad de personas.
17	LOC	Lugares sobre regiones, ríos, lagos.

Características El conjunto de características utilizadas consiste de las partes de las palabras, etiquetas POS (Part-of-Speech) simplificadas, banderas que indican la forma de la palabra: minúsculas, mayúsculas, título o dígito. El inicio o fin de la oración y características de palabras cercanas.

Parámetros utilizados Para este modelo se utilizó CRFsuite[3] que es una biblioteca de Python que implementa Campos Aleatorios Condicionales, los parámetros usados fueron los siguientes:

- *algorithm* = *lbfgs* (Gradiente descendente usando el método L-BFGS)
- *c1* = 0.1 (Coeficiente de regularización para L1)
- *c2* = 0.1 (Coeficiente de regularización para L2)
- *max_iterations* = 50 (El número máximo de iteraciones para algoritmos de optimización).
- *all_possible_transitions* = *True* (CRFsuite genera funciones de transición que asocian todos los pares de etiquetas posibles)

[3]http://www.chokkan.org/software/crfsuite/

Modelos basados en Redes Neuronales

Dos modelos de redes neuronales se utilizaron para los experimentos, basados en la arquitectura de Larga Memoria a Corto Plazo bidireccional (Bi-LSTM) que pertenecen a las redes neuronales recurrentes. El primer modelo utiliza embeddings generados por los índices del vocabulario. El segundo modelo denominado Bi-LSTM-ELMo utiliza embeddings pre-entrenados con ELMo. En los modelos Bi-LSTM y Bi-LSTM-ELMo fue usada la biblioteca de alto nivel de Python Keras[4] 2.2.4, y TensorFlow 1.13.1.

Modelo LSTM bidireccional (Bi-LSTM)

La Figura 4.7 describe las capas que componen el modelo Bi-LSTM. La configuración utilizada para el modelo Bi-LSTM se detalla a continuación:

- **Capa de entrada**. Se introducen las oraciones previamente codificadas a vectores de longitud 50. Las oraciones mayores a 50 fueron divididas, y las oraciones menores a 50 fueron rellenadas con el token "<*pad*>".
- **Capa de Embeddings**. Esta capa entrena los vectores introducidos de longitud 50 obteniendo vectores con pesos de 300 por cada ítem. Los parámetros usados son:
  - *input_dim*. Es el tamaño del vocabulario, para el CoNLL-2002 y Mx-news respectivamente.
  - *output_dim* = 300
  - *input_length* = 50
- **Capa Bi-LSTM**. La capa LSTM es alimentada por la capa Embeddings, y está configurada:
  - *units* = 100
  - *dropout* = 0.01
  - *recurrent_dropout* = 0.3
  - *return_sequences* = True

[4]https://keras.io/

La función de activación es la tangente hiperbólica (*tanh*). Una capa envolvente *Bidirectional* es usada para aprender características de alto nivel en ambas direcciones (hacia adelante y hacia atrás) de la capa LSTM.

- **Capa de salida**. *TimeDistributed Dense* se utiliza para aplicar densidades totalmente conectadas en cada paso de tiempo para obtener la salida por separado mediante pasos de tiempo. Los parámetros usados son:
  - *units* = numero de clases individuales. Para CoNLL-2002 (8) y para Mx-news (65).
  - *activation* = función *softmax*.

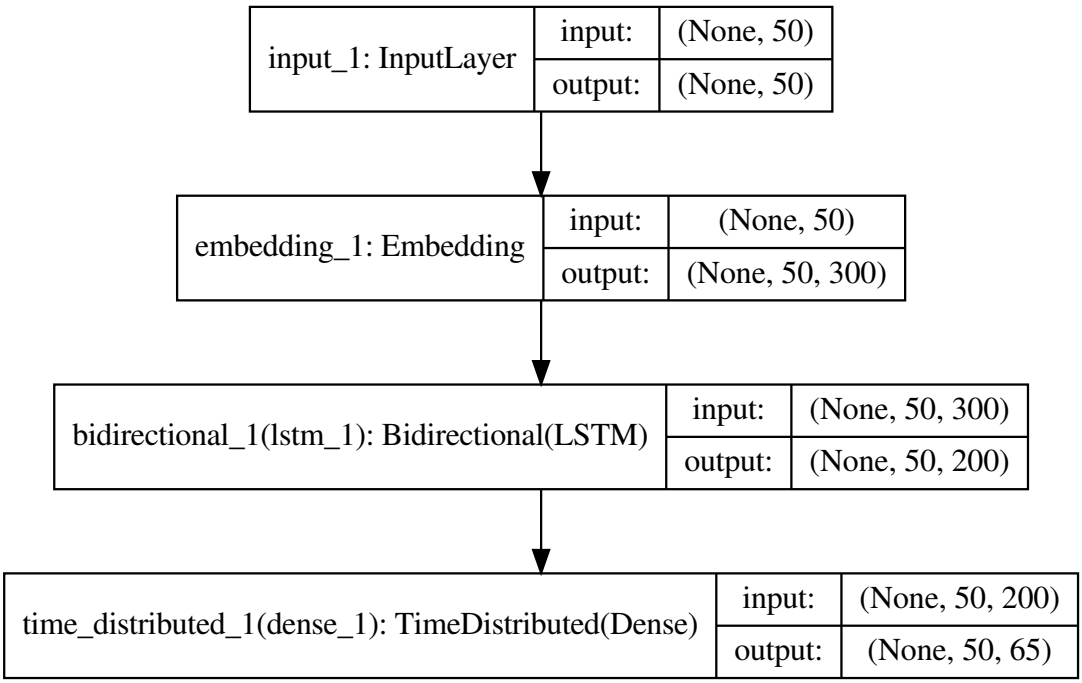

Fig. 4.7 Diagrama general del modelo Bi-LSTM.

Para el entrenamiento de la red neuronal se uso la siguiente configuración:

- *optimizer* = algoritmo RMSprop
- *learning rate* = 0.001
- *learning rate decay* = 0.0
- *metrics* = *accuracy* and *loss* = *categorical_crossentropy*

- *bach_size* = 50
- *epochs* = 20
- *validation_split* = 0.1
- *shuffle* = True

Modelo LSTM bidireccional con ELMo embeddings (Bi-LSTM-ELMo) Las capas utilizadas en el modelo Bi-LSTM-ELMo se describen de forma general en la Figura 4.8, y se detallan a continuación:

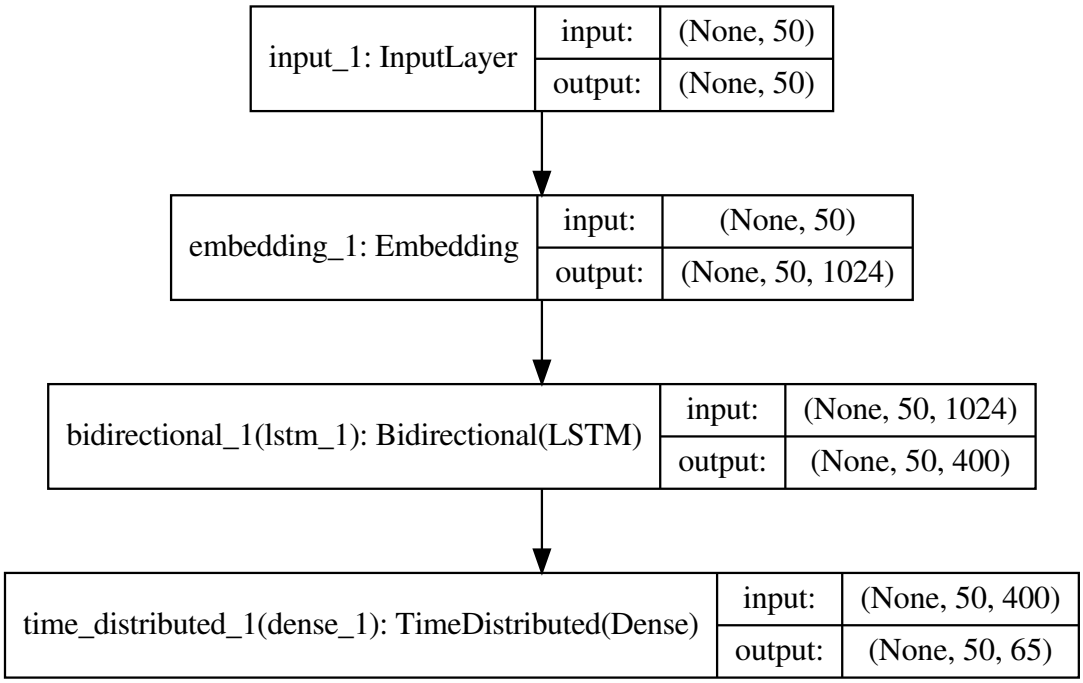

Fig. 4.8 Diagrama general del modelo Bi-LSTM-ELMo.

- **Capa de entrada**. Se introducen las oraciones previamente codificadas a vectores de longitud 50.
- **Capa de Embeddings**. Para esta capa se entrenaron embeddings en idioma español con ELMo (Peters et al., 2018), usando la biblioteca *elmoformanylangs*[5] *0.0.2* de Python

[5]https://github.com/HIT-SCIR/ELMoForManyLangs

basada en el trabajo de Che et al. (2018). Cada uno de los tokens del vocabulario de ambos corpus (CoNLL-2002 y Mx-news) son mapeados a vectores de longitud 1024. Los embeddings pre-entrenados se vuelven a entrenar en esta capa. Los parámetros usados son:

  - *input_dim*. Es el tamaño del vocabulario, para el CoNLL-2002 y Mx-news respectivamente.
  - *output_dim* = 1024
  - *input_length* = 50
  - *weights* = matriz de embeddings ELMo pre-entrenada
  - *trainable* = True

- **Capa Bi-LSTM**. La capa LSTM es alimentada por la capa Embeddings, y está configurada:

  - *units* = 200
  - *dropout* = 0.01
  - *recurrent_dropout* = 0.3
  - *return_sequences* = True

  La función de activación es la tangente hiperbólica (*tanh*). Una capa envolvente *Bidirectional* es usada para aprender características de alto nivel en ambas direcciones (hacia adelante y hacia atrás) de la capa LSTM.

- **Capa de salida**. *TimeDistributed Dense* se utiliza para aplicar densidades totalmente conectadas en cada paso de tiempo para obtener la salida por separado mediante pasos de tiempo. Los parámetros usados son:

  - *units* = numero de clases individuales. Para CoNLL-2002 (8) y para Mx-news (65).
  - *activation* = función *softmax*.

Para el entrenamiento de la red neuronal se uso la siguiente configuración:

- *optimizer* = algoritmo RMSprop
- *learning rate* = 0.001

- *learning rate decay* = 0.0
- *metrics* = *accuracy* and *loss* = *categorical_crossentropy*
- *bach_size* = 50
- *epochs* = 20
- *validation_split* = 0.1
- *shuffle* = True

Experimentos

Los experimentos consisten en analizar el comportamiento del modelo CRF con ambos corpus el CoNLL-2002 y Mx-news en dos procedimientos.

El primero procedimiento consiste en analizar únicamente la clase más dispersa MISC para el CoNLL-2002, y la clase LOC para el corpus Mx-news. En el experimento siguiente se agrega la segunda clase menos dispersa, LOC y AGE para los corpus CoNLL-2002 y Mx-news respectivamente. Este proceso se repite cada vez hasta alcanzar la clase más densa de cada corpus. Por lo que el número de experimentos es igual al número de clases. Este procedimiento va de la clase más dispersa a la más densa.

Para el segundo procedimiento el orden de análisis es inverso, de la clase más densa (ORG para CoNLL-2002 y PER para Mx-news) a la más dispersa (MISC para CoNLL-2002 y LOC para Mx-news). De igual forma se integra una clase cada vez hasta terminar el proceso.

Evaluación

Para la evaluación de los procedimientos se utilizan las métricas de *precision*, *recall* y *F1-score* así como la métrica de *Micro-Average* para las métricas mencionadas previamente.

$$precision = \frac{TP}{(TP+FP)} \tag{4.1}$$

$$recall = \frac{TP}{(TP+FN)} \tag{4.2}$$

$$F1-score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{4.3}$$

donde *TP* es el número de palabras correctamente detectadas, *FP* es el número de palabras erróneamente detectadas y *FN* es el número de palabras erróneamente detectadas como una clase otra "O".

$$Macro - Average \quad mesaure = \frac{\sum_{i=1}^{N} measure_i}{N} \tag{4.4}$$

En *Macro-Average measure*, *measure* corresponde a *precision*, *recall* y/o *F1-score* y es calculada primero localmente sobre cada categoría y luego se toma el promedio general de la NE. (Özgür et al., 2005). En la ecuación 4.4 *measure* para cada NE *i* es calculada, *N* es el número de NEs, y el *Macro-Average* es obtenido tomando el promedio de *measure* para cada NE.

Dos conjuntos de evaluaciones son usados para las evaluaciones de los experimentos sobre los dos esquemas de etiquetado (IOB, IOBES), y para ambos conjuntos de datos. Las evaluaciones son realizadas se presentan con ejemplos para una mejor comprensión.

Evaluación de etiquetas individuales Este conjunto se aplica sobre las etiquetas individuales para cada entidad nombrada (NE). Del siguiente ejemplo, *y_test* es el conjunto de prueba que contiene las etiquetas "verdaderas". Las etiquetas predecidas por el modelo en cuestion son representadas por *y_pred*.

Tabla 4.4 Ejemplo para evaluación de etiquetas individuales.

y_test	'O'	'B-LOC'	'I-LOC'	'I-LOC'	'O'	'B-PER'	'I-PER'	'O'
y_pred	'B-LOC'	'I-LOC'	'I-LOC'	'O'	'O'	'B-PER'	'I-PER'	'O'

Los resultados de aplicar las métricas de *precision*, *recall* y *F1-score* son descritos en la Tabla 4.5, así como el número de etiquetas que se usaron para realizar las evaluaciones por la columna *soporte*. En el último renglón se calcula el macro promedio de cada una de las métricas, que es la medida tomada como resultado para todos los experimentos.

Evaluación sobre entidades nombradas El segundo conjunto de evaluación es aplicado sobre una NE completa para ambos esquemas (IOB e IOBES), es decir compuesta por una o más etiquetas individuales. De la Tabla 4.4 que muestra un ejemplo a evaluar, únicamente se toman la etiquetas correspondiente a las clases y se omiten las etiquetas de los esquemas IOB e IOBES, como muestra la Tabla 4.6.

Tabla 4.5 Ejemplo de evaluación sobre etiquetas individuales.

Etiqueta	Precision	Recall	F1-score	Soporte
B-LOC	0.00	0.00	0.00	1
I-LOC	0.50	0.50	0.50	2
B-PER	1.00	1.00	1.00	1
I-PER	1.00	1.00	1.00	1
macro avg	0.62	0.62	0.62	5

Tabla 4.6 Ejemplo de evaluación sobre entidades nombradas.

Etiqueta	Precision	Recall	F1-score	Soporte
PER	1.00	1.00	1.00	1
LOC	0.00	0.00	0.00	1
macroavg	0.50	0.50	0.50	2

4.2 Extracción automática de relaciones

Esta sección comprende la *extracción de relaciones* de forma automática de dos entidades. Las entidades nombradas son obtenidas usando el modelo generado en la Sección 4.1.

La metodología se describe en la Figura 4.9. El proceso consiste utilizar el modelo NER para etiquetar documentos sobre noticias políticas. El modelo NER devuelve como salida documentos etiquetados con las entidades nombradas reconocidas y clasificadas. Estos documentos a su vez son pre-procesados. Consiste de dividir cada documento en oraciones, se realiza un filtro de entidades nombradas, en este caso se omiten las entidades de *dirección* (ADD) y *porcentaje* (PRC) de las 17 planteadas en la Sección 4.1. Además como parte del pre-procesamiento se crean oraciones con solo dos entidades nombradas. Si una oración contiene más de dos entidades se crea una ventana que contiene dos entidades, la ventana se mueve a través de la oración permitiendo solo dos entidades y en cada movimiento se crea una nueva oración. Estas oraciones se usaran en la extracción de relaciones. En este proceso la metodología se centra en las relaciones de dependencia entre tokens. La biblioteca de Python Spacy[6] es empleada para este propósito. Por lo cual se genera un árbol de dependencias por cada oración, y se aplican métodos para identificar y extraer la relación que de sentido a las entidades involucradas. Las relaciones extraídas y las entidades nombradas en cuestión forman tripletas, las cuales son almacenadas en una base de datos así como la oración involucrada.

[6]https://spacy.io/

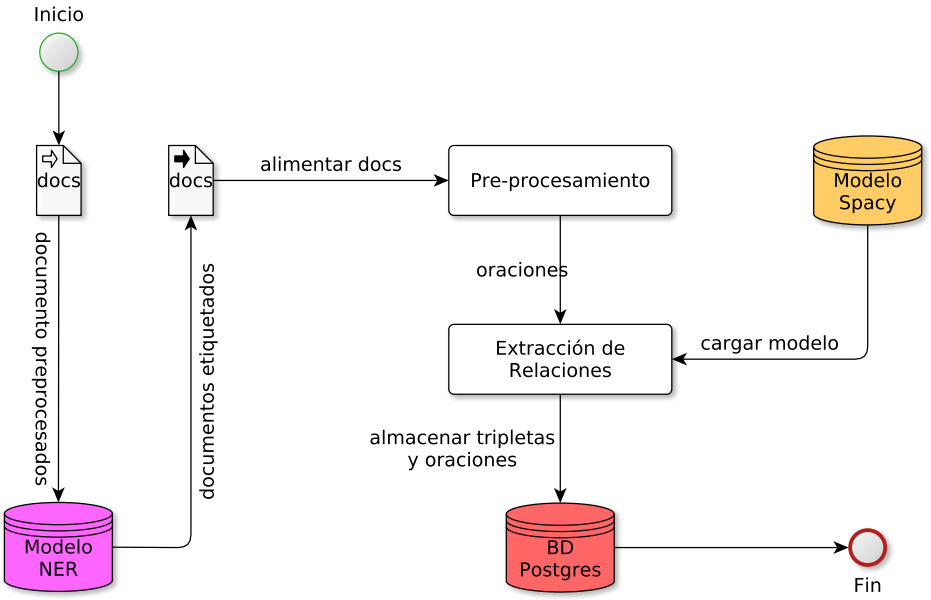

Fig. 4.9 Modelo para la Extracción de Relaciones.

4.2.1 Pre-procesamiento

En el pre-procesamiento se reciben como entrada los documentos etiquetados por el modelo NER. Cada documento es dividido en oraciones. El proceso verifica que al menos dos entidades nombradas estén incluidas dentro de la oración (en caso contrario se omite la oración), con el objetivo de encontrar una relación para las dos entidades que forman una tripleta (<*entidad1, relación, entidad2*>). Las oraciones con más de dos entidades son duplicadas n veces donde $n = E - 1, E = totalentidades$. Un ejemplo se describe en la siguiente oración.

La embajada de ***Estados Unidos (GPE)*** *informó que el* ***presidente (TIT) Donald Trump (PER)*** *realizó un cambio en la delegación que asistirá a la toma de posesión de* ***Andrés Manuel López Obrador (PER)*** .

La oración contiene cuatro entidades nombradas (GPE, TIT, PER, PER) lo que dará como resultado 3 oraciones. En cada oración se elige una ventana de dos entidades. Se remueven las etiquetas de las entidades fuera de la ventana seleccionada. Las tres oraciones contendrán únicamente pares de entidades. A continuación se listan las oraciones generadas:

1. La embajada de **Estados Unidos (GPE)** informó que el **presidente (TIT)** Donald Trump realizó un cambio en la delegación que asistirá a la toma de posesión de Andrés Manuel López Obrador .

2. La embajada de Estados Unidos informó que el **presidente (TIT) Donald Trump (PER)** realizó un cambio en la delegación que asistirá a la toma de posesión de Andrés Manuel López Obrador .

3. La embajada de Estados Unidos informó que el presidente **Donald Trump (PER)** realizó un cambio en la delegación que asistirá a la toma de posesión de **Andrés Manuel López Obrador (PER)** .

Este proceso es aplicado únicamente a oraciones con más de dos entidades. Las oraciones con exactamente dos entidades y la generadas son utilizadas en el proceso de extracción de relaciones.

4.2.2 Extracción de relaciones

Aquí se procesa cada oración de forma individual. Se utilizan los árboles de dependencias (véase Figura 4.11) entre los tokens (palabras, símbolos y entidades) que componen la oración. Una vez generado el árbol de la oración se transforma en un grafo dirigido. Los nodos están representados con círculos y contienen el *id* que representa la posición del token en el texto de la oración, la etiqueta POS (Part of Speech) y el token que puede ser una palabra, un símbolo o una entidad nombrada etiquetada. Las líneas son aristas que unen los nodos y representan la relación de dependencia entre dos tokens. El grafo es utilizado para trazar rutas simples, además para obtener los nodos descendentes y/o los ancestros de un nodo.

En este proceso se definieron algunos métodos para la identificación y extracción de la relación que existe para dos entidades nombradas. Una vez extraída la relación se forman tripletas con las entidades nombradas, y se almacenan en una base de datos así como la oración para futuros usos. Los métodos para la extracción son los siguientes:

Relaciones sobre puestos de trabajo

Este método esta enfocado principalmente en la identificación de relaciones sobre el puesto de trabajo (título) de una persona así como otras más. Para ello, se establecieron diez relaciones en base a la aparición de las entidades dentro del árbol de dependencias previamente generado.

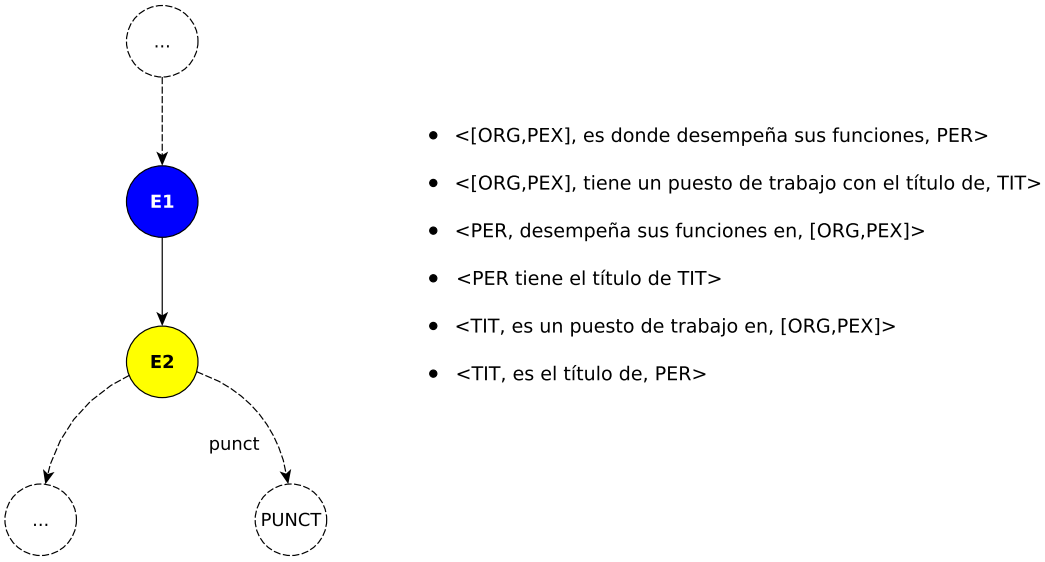

Fig. 4.10 Estructura de las relaciones sobre puestos de trabajo.

La *entidad 1* (E1) es la que aparece primero dentro del texto de la oración. La *entidad 2* (E2) es la que aparece después de E1 dentro del texto. La Figura 4.10 muestra parte de la estructura de un grafo para identificar la relación sobre puestos de trabajo. En donde E1 puede tener cualquier relación de dependencia con E2 (por ello no se especifica en la Figura 4.10) pero E2 debe de ser el descendiente directo de E1. Además E2 puede contener (línea punteada en la Figura 4.10) algún nodo con una etiqueta POS de tipo PUNCT (signo de puntuación) o cualquier otra etiqueta POS. PUNCT debe ser una *coma* (,), *punto y coma* (;) o *paréntesis que abre* (() o no contener ninguno de estos símbolos.

Para identificar relaciones sobre la temática de puestos de trabajo se establecieron patrones. Estos patrones se basan en la clase de la entidad nombrada. En la Figura 4.10 se describen las tripletas que se identifican siguiendo los patrones. Por ejemplo en la primera tripleta de la expresión [ORG,PEX] significa que en la tripleta E1 puede ser una entidad de clase *organización* (ORG) o de *partido político* (PEX) y E2 debe de ser una entidad de clase *persona* (PER). Así con la combinación de patrones de E1 ([ORG,PEX]) y E2 (PER) se obtendrán dos relaciones *"es donde desempeña sus actividades"*. De este modo cuando se cumplen las condiciones de la Figura 4.10 se crea la tripleta con la relación correspondiente que se han establecido previamente.

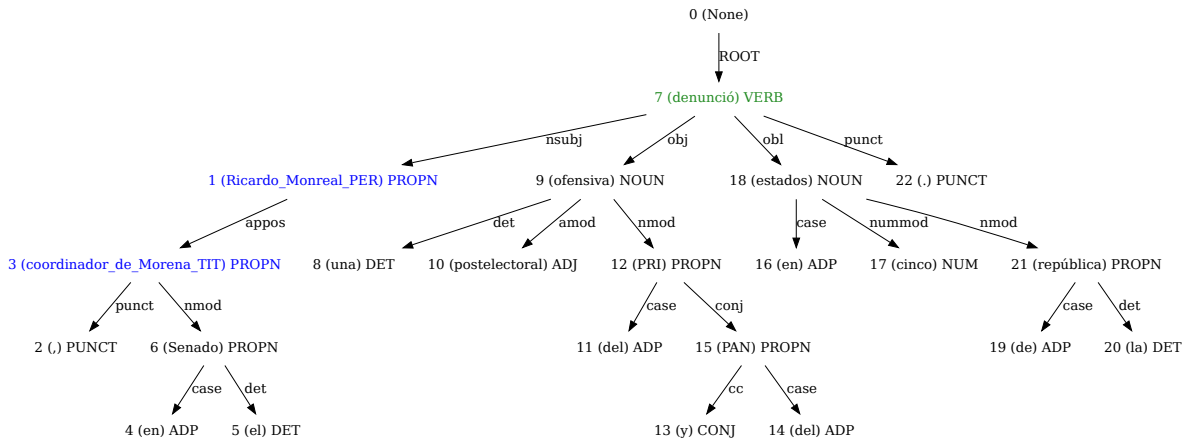

Fig. 4.11 Ejemplo de relación sobre puestos de trabajo.

A continuación se describe un ejemplo sobre la identificación de relaciones bajo este método. A la oración *"Ricardo_Monreal_PER , coordinador_de_Morena_TIT en el Senado denunció una ofensiva postelectoral del PRI y del PAN en cinco estados de la república ."* se le aplica el análisis de dependencias (*dependency parser*) y se obtiene el árbol de dependencias mostrado en la Figura 4.11. Del ejemplo, el primer paso consiste en verificar que la entidad nombrada E2 sea descendiente directo de E1. Además de analizar los descendientes de E2. Después se verifican las clases de las entidades en la lista de patrones. En este caso E1 es una PER y E2 es un TIT, el patrón que se obtiene es PER-TIT y cumple con el criterio sobre la cuarta tripleta de la Figura 4.10. De este modo se obtiene la relación *"tiene el título de"*. La tripleta resultante se almacena como <*"Ricardo_Monreal_PER", "tiene el título de", "coordinador_de_Morena_TIT"*>. Posteriormente se almacena la tripleta y la oración en la base de datos.

Relación: es acrónimo de

El método para identificar a relaciones de tipo *"acrónimo"* se describe en la Figura 4.12. E2 debe ser descendiente directo de E1 sin importar la relación de dependencia que contenga. Además los nodos descendientes de E2 deben de ser POS del tipo PUNCT así como contener uyna relación de dependencia *punct* con E2. Los signos de puntuación son paréntesis de apertura y cierre respectivamente. Para que la relación *"acrónimo"* se cumpla E1 y E2 deben de ser de la misma clase. Solo se permiten las entidades mostradas en la Figura 4.12.

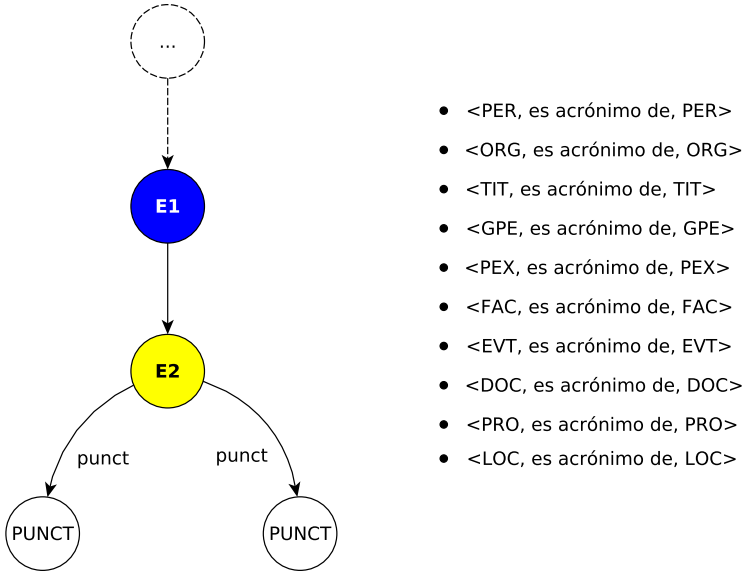

Fig. 4.12 Estructura para identificar la relación *acrónimo*.

Un ejemplo de la relación *"acrónimo"* es la oración: *"Fue entonces cuando el Partido del Trabajo notificó al Partido_de_la_Revolución_Democrática_PEX (PRD_PEX) que no firmarían con ellos sino con el PAN ."* El árbol de dependencias generado se muestra en la Figura 4.13. Para identificar la relación de *"acrónimo"* E1 debe tener como descendiente directo al nodo E2, a su vez E2 debe tener dos nodos descendientes POS del tipo PUNCT a través de la relación de dependencia *punct*. En el texto E2 debe encontrarse entre paréntesis y ambas entidades deben ser de la misma clase. El ejemplo pertenecen a la clase *partido político* como se describe en la Figura 4.13.

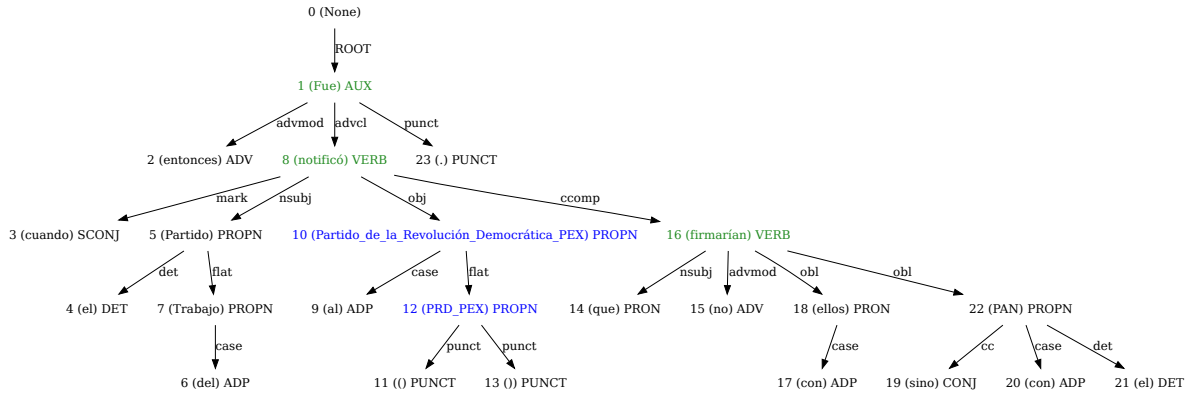

Fig. 4.13 Ejemplo del árbol de dependencias de la relación *acrónimo*.

La relación resultante como se observa en la Figura 4.13 es *“es acrónimo de”*. Las entidades nombradas se unen para formar la tripleta <*Partido_de_la_Revolución_Democrática_PEX, es acrónimo de, PRD_PEX*> que será almacenada al igual que la oración en la base de datos.

Relación: que pertenece a

El método describe la forma de identificar la relación *"se localiza en"* y la relación *"que pertenece a"*. En la Figura 4.14 E1 debe tener una relación de dependencia *"flat"*, *"nmod"* o *"appos"* con E2. E1 debe pertenecer a la clase FAC (instalación) u ORG (organización) y E2 debe pertenecer a la clase GPE (geopolítica). De manera opcional E2 puede tener nodos descendientes POS del tipo ADP (Adposición), DET (Determinante) y/o PUNCT (paréntesis de apertura y cierre). Estas nodos opcionales deben contener la relación de dependencia *case*, *det*, y/o *punct* respectivamente.

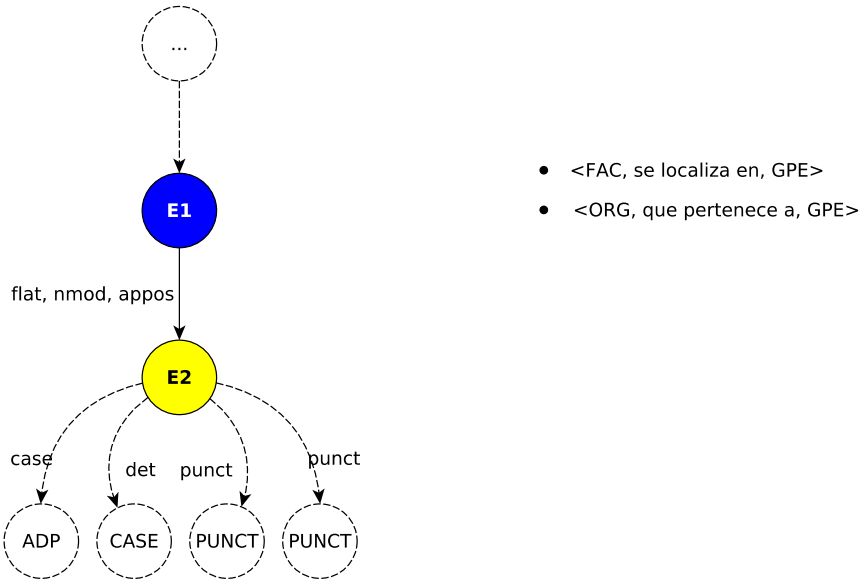

Fig. 4.14 Estructura para identificar la relación *que pertenece a*.

La oración de ejemplo: *“El Tribunal_de_Justicia_Electoral_ORG de Baja_California_GPE aprobó con dos votos a favor y uno en contra , que el siguiente período de Gobierno del Estado será de seis años .”* obtiene el árbol de dependencias que ilustra la Figura 4.15. Se observa que E1 pertenece a la clase ORG y tiene como descendiente directo con una relación de dependencia *nmod* a E2 con clase GPE. A su vez E2 tiene una relación de dependencia *case* con el nodo descendiente POS del tipo ADP. Como se cumplen todos

estos requerimientos la relación que se le asigna es *"que pertenece a"* y la tripleta resultante es: <*Tribunal_de_Justicia_Electoral_ORG, que pertenece a, Baja_California_GPE*>.

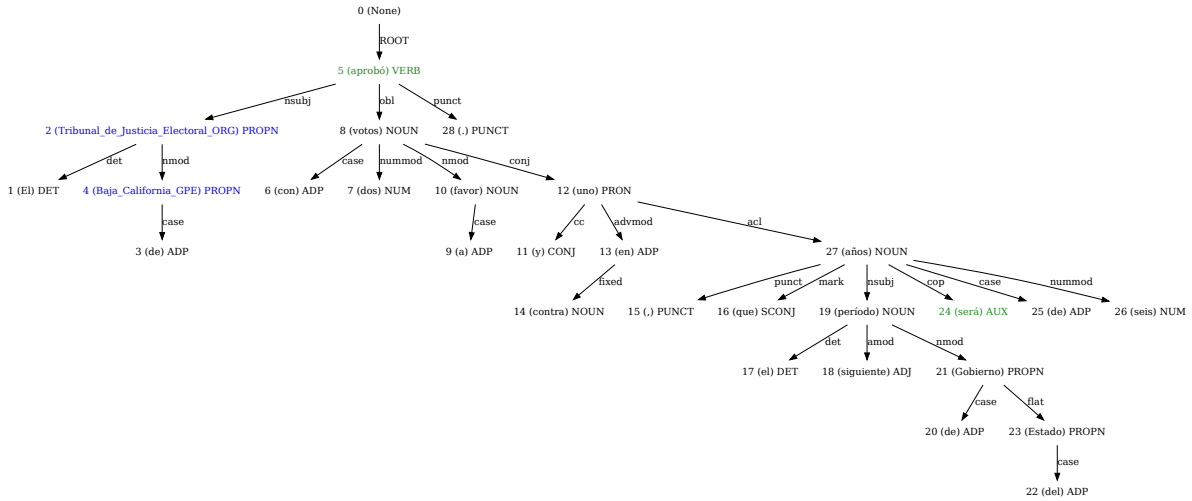

Fig. 4.15 Ejemplo del árbol de dependencias de la relación *que pertenece a*.

Relación: es representado por

El método para identificar la relación *"es representado por"* se describe en la Figura 4.16. La relación de dependencia entre E1 y E2 debe ser *flat* o *appos*. E2 debe ser descendiente directo de E1. E1 debe pertenecer a la clase GPE (geopolítica) y E2 pertener a la clase PER (persona). E2 debe tener un nodo descendiente POS del tipo PUNCT con la relación de dependencia *punct*. El símbolo del nodo PUNCT debe ser la coma (,).

El árbol de dependencias que se genera de la oración: *"Los gobernadores de Guanajuato , Diego Sinuhé , y de Jalisco_GPE , Enrique_Alfaro_PER , relataron para Milenio , que la comida fue muy amena ."* se puede observar en la Figura 4.17. En la Figura 4.17 se observa los requerimientos planteados para identificar la relación *"es representado por"*, donde la entidad E1 y E2 pertenecen a la clase GPE y PER respectivamente. Se observa que las entidades están ligadas con la relación de dependencia *flat*. E2 tiene como nodo descendiente POS del tipo PUNCT, con la relación de dependencia *punct*. El nodo PUNCT es el signo de puntuación coma (,).

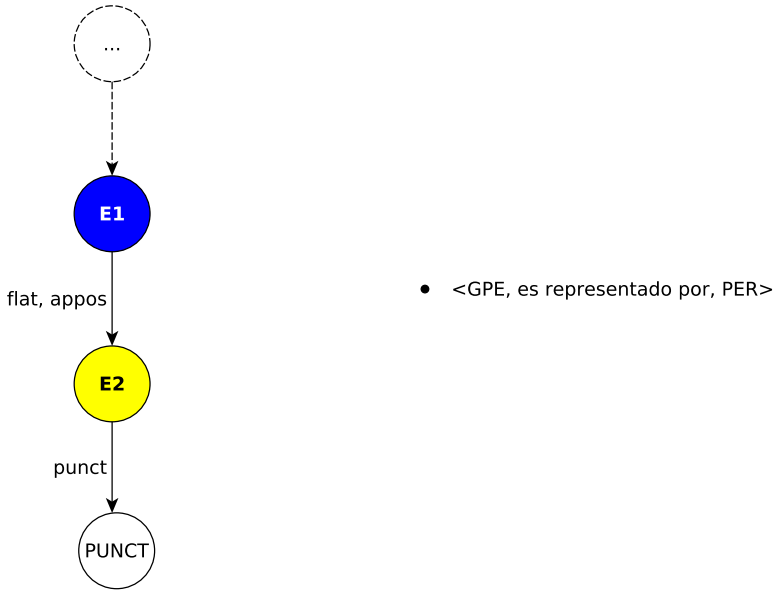

Fig. 4.16 Estructura para identificar la relación *es representado por*.

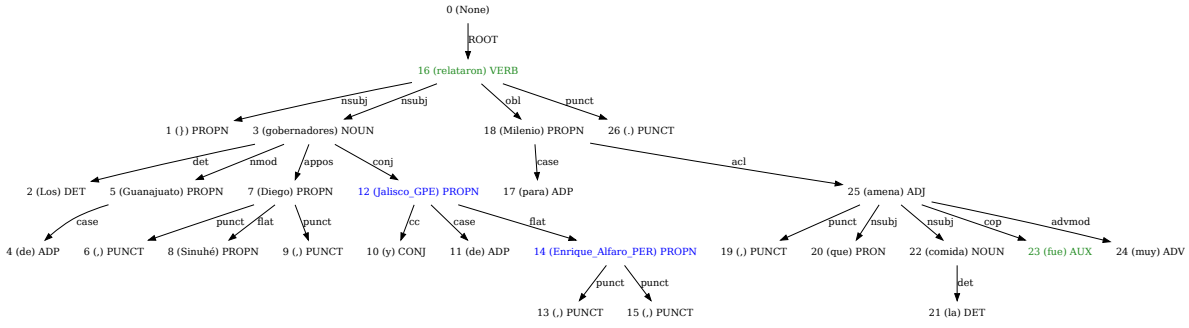

Fig. 4.17 Ejemplo del árbol de dependencias de la relación *es representado por*.

Relaciones usando: appos

Al usar la relación de dependencia *appos* se pueden identificar diferentes relaciones entre entidades nombradas, como son relaciones familiares, cargo o puesto o frases que se encuentran entre dos entidades nombradas. El método se describe en la Figura 4.18 así como en la Figura 4.19 se ilustran el complemento del método.

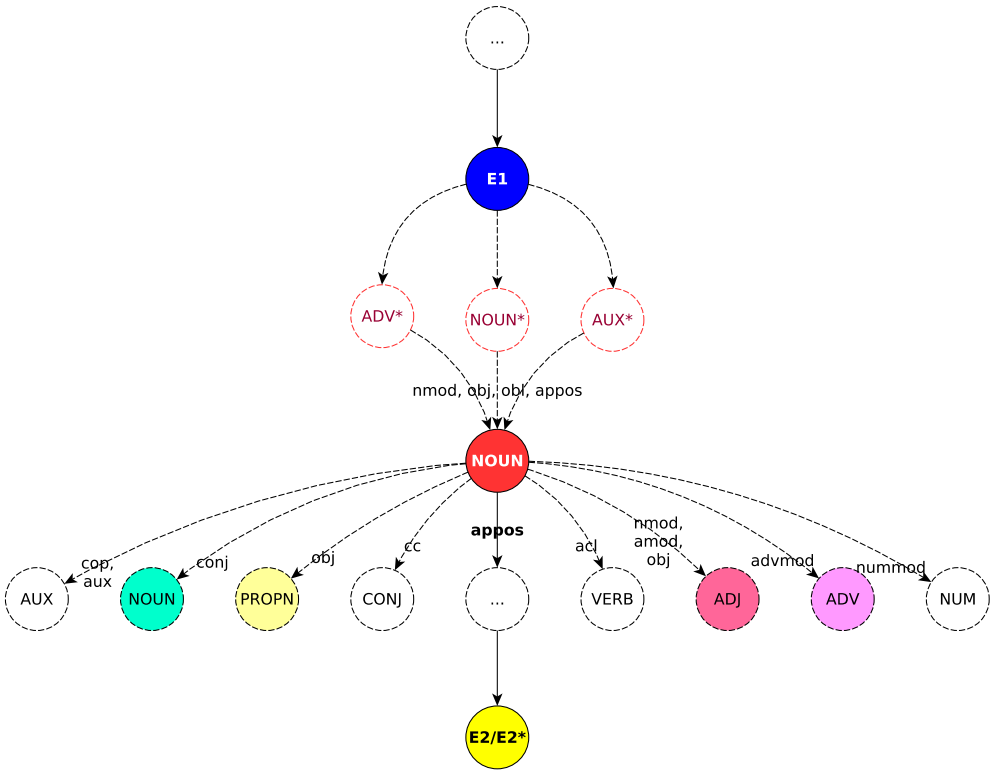

Fig. 4.18 Estructura para identificar relaciones usando *appos*.

El método consiste en buscar una ruta simple entre E1 y E2 dentro del grafo, identificar la relación de dependencia *"appos"* en donde tenga como padre a un nodo POS del tipo NOUN (sustantivo). En la Figura 4.18 se observa el nodo NOUN en color rojo. Es el nodo que sirve como punto de origen para identificar diferentes nodos POS opcionales como descendientes de el, que a su vez cada uno de esos nodos brindan opciones para poder ensamblar una relación que esté en contexto con las entidades nombradas involucradas. Además opcionalmente se puede reunir información del nodo NOUN observando su ancestro, en este caso el ancestro pueden ser ADJ (adjetivo), NOUN (sustantivo) o AUX (verbo auxiliar) y la relación de dependencia de

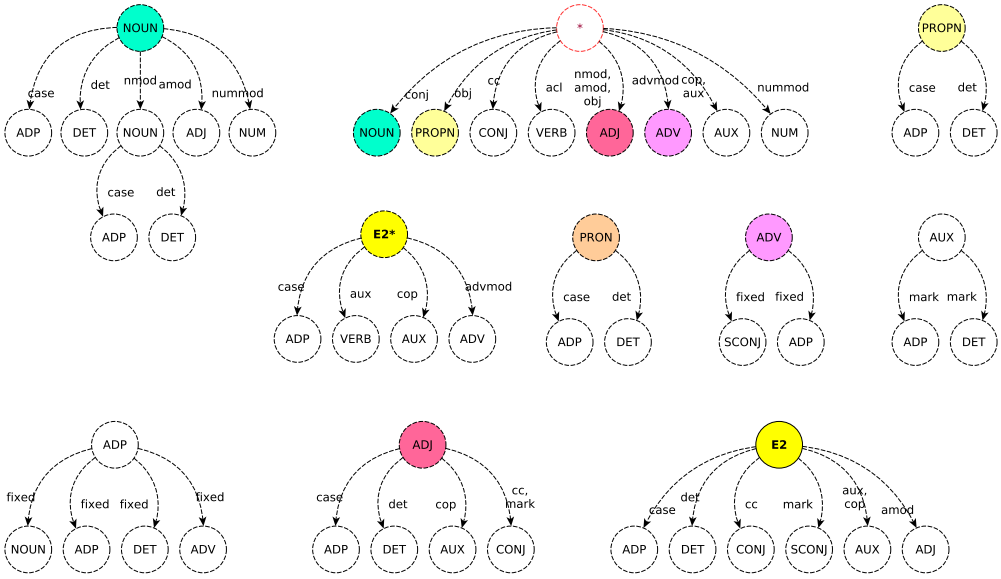

Fig. 4.19 Estructura complemento para identificar relaciones usando *appos*.

esos nodos hacia el nodo punto de referencia NOUN pueden ser cualquiera de las siguientes *nmod*, *obj*, *obl* o *appos* como se observa en la Figura 4.18.

La relación más básica que se puede encontrar en este método consiste del nodo E1, NOUN (color rojo) con una relación de dependencia *"appos"* hacia E2, aunque no necesariamente E2 sea un descendiente directo de NOUN, de la misma forma E1 puede o no ser un ancestro directo de NOUN. Los nodos y aristas con línea de puntos son opcionales. Es decir, puede tomarse uno o más nodos, o ninguno de ellos para identificar una relación entre entidades nombradas.

La Figura 4.19 describe el resto de la estructura de este método. Los colores están asociados a la estructura principal mostrada en la Figura 4.18. Como se observa en la Figura 4.19 existen diferentes opciones como son NOUN, (*) corresponde a los nodos ancestro del nodo tomado como punto de referencia, PROP (nombre propio), PRON (pronombre), ADV (adverbio), ADJ (adjetivo). Sobre los nodos ADP (adposición) y AUX (verbo auxiliar) se va a extraer información cuando se cumplan las relaciones de dependencia *fixed* y *mark* respectivamente.

E2 se analiza para conocer si sus descendientes pueden proporcionar información relevante. La Figura 4.19 muestra las relaciones de dependencia y los nodos que deben cumplir los descendientes de E2, para que pueda ser tomada en cuenta esta información.

A continuación se presentan algunos ejemplos para identificar y extraer relaciones entre entidades nombradas utilizando la relación de dependencia *"appos"*. Para el primer ejemplo se toma la oración *"López_Obrador_PER , mejor conocido por sus iniciales AMLO_PER ,*

busca desmarcarse de la clase política que ha gobernado México durante casi un siglo y se ha presentado como un adalid en la lucha contra la corrupción .” el árbol de dependencias parcial se muestra en la Figura 4.20. El primer paso es identificar una ruta simple entre las entidades nombradas (color azul). La ruta simple se busca iniciando por la entidad más profunda en el árbol, en este caso el nodo de partida es *AMLO_PER*, seguido del nodo *iniciales*, nodo *conocido* hasta llegar al nodo de la entidad en el nivel más alto *López_Obrador_PER*. Después se busca la relación de dependencia *“appos”* iniciando nuevamente con el nodo de la entidad más profunda. Una vez encontrada la relación de dependencia se verifica que el ancestro sea un sustantivo NOUN.

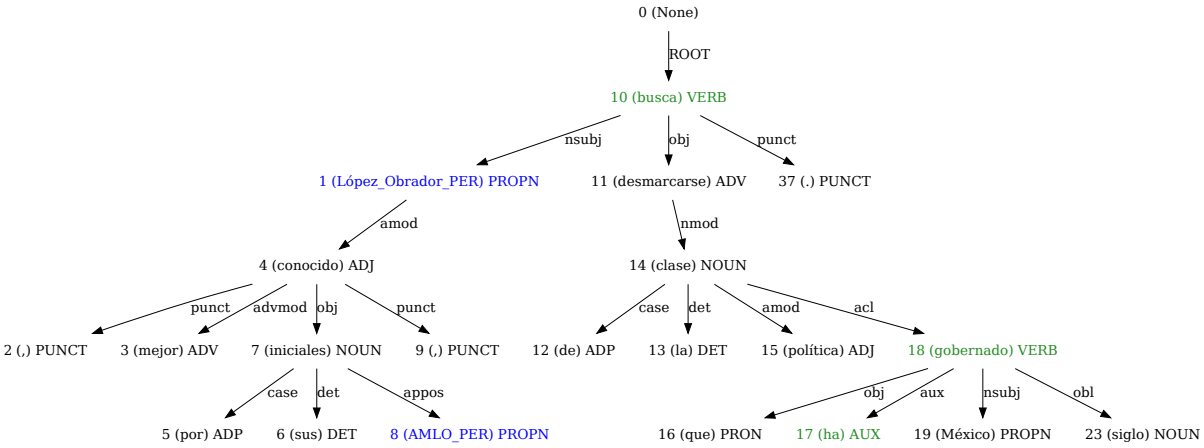

Fig. 4.20 Ejemplo 1 del árbol de dependencias usando la relación *appos*.

Así el nodo (*iniciales*) NOUN es el punto de partida para identificar y extraer la relación que existe entre ambas entidades. Se analiza el ancestro de NOUN que es el nodo (*conocido*) ADJ con una relación de dependencia *obj*. Esto satisface la regla establecida en la Figura 4.18 por lo que se procede a identificar a los ancestros del nodo (*conocido*) ADJ. En este caso no hay más opciones. Al analizar los descendientes del nodo (iniciales) NOUN, los nodos hijos (*por*) ADP y (*sus*) DET con las relaciones de dependencia *case* y *det* respectivamente satisfacen lo establecido en la Figura 4.19. De este modo se extraen todos los nodos identificados y se ordenan en base a la posición que ocupan en la oración. Entonces la relación entre ambas entidades es *“conocido por sus iniciales”*. La tripleta se forma con ambas entidades nombradas y la relación extraída (<*López_Obrador_PER, conocido por sus iniciales, AMLO_PER*>) para después ser almacenada.

El segundo ejemplo contempla la oración *“Aseguró que el futuro que está por venir es glorioso , de tranquilidad y bienestar para Veracruz , ya que se tiene la certeza de*

que este parte aguas histórico , el principal triunfo es haber logrado que dentro de unas horas Andrés_Manuel_López_Obrador_PER sea el Presidente_de_México_TIT .” el árbol de dependencias generado de forma parcial se ilustra en la Figura 4.21.

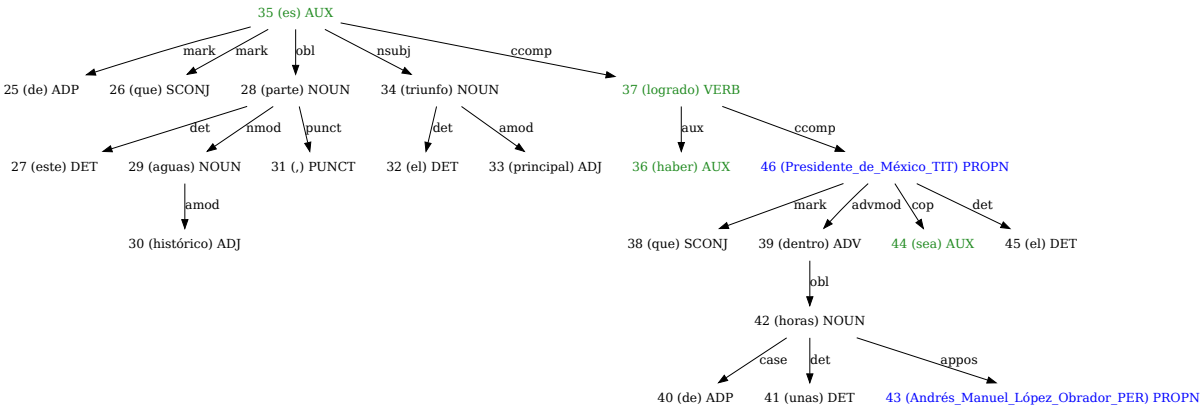

Fig. 4.21 Ejemplo 2 del árbol de dependencias usando la relación *appos*.

Siguiendo el método antes mencionado se encuentra la relación de dependencia *“appos”* y el nodo principal del que se realizara el análisis (*horas*) NOUN en este de la Figura 4.21. Los descendientes de este nodo son los nodos (*de*) ADP y (*unas*) DET. Este ejemplo muestra un caso particular y ocurre cuando E1 (primera entidad en el texto de la oración) aparece en los niveles más profundos y E2 en los niveles más altos. Esto se puede observar en el *id* de las entidades. En el ejemplo E1 tiene el *id* 43 y E2 el *id* 46. Cuando ocurre este caso se verifican los descendientes de E2 los cuales pueden ser nodos POS del tipo AUX con relación de dependencia *cop*, VERB con relación *aux*, ADP con la relación *case* y ADV con la relación *advmod* como lo describe E2* en la Figura 4.19. De este modo se identifican los descendientes de E2* que son el nodo (*dentro*) ADV con la relación de dependencia *advmod* y el nodo (*sea*) AUX con la relación de dependencia *cop*. La relación resultante es *“dentro de unas horas sea”*. Para formar la tripleta se sigue de forma estricta el orden (*id*) en que las entidades nombradas aparecen en el texto de la oración, la tripleta es <*Andrés_Manuel_López_Obrador_PER, dentro de unas horas sea, Presidente_de_México_TIT*>. Se almacena la tripleta y se continua con la siguiente oración.

Relaciones usando: amod

Este método describe el procedimiento para identificar la relación que existe para dos entidades nombradas usando la relación de dependencia *"amod"* como foco del análisis. La Figura 4.22 describe el proceso de forma gráfica.

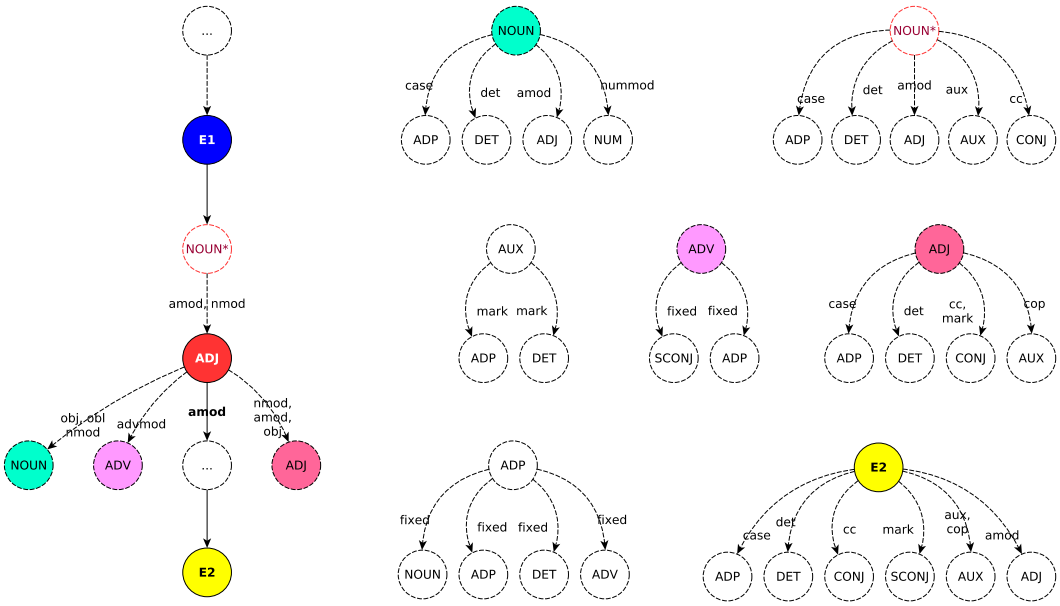

Fig. 4.22 Estructura para identificar relaciones usando *amod*.

En el primer paso se busca que exista una ruta simple entre ambas entidades y que exista al menos una relación de dependencia *"amod"*. A diferencia del método anterior aquí se busca la relación de dependencia *"amod"* iniciando en el nivel más alto hasta llegar a los niveles más profundos. Una vez encontrada la relación de dependencia se analiza. Se toma el nodo destino de la relación *"amod"*. Se verifica que el nodo POS elegido previamente (ancestro o descendiente) sea del tipo ADJ, este nodo sera el punto de partida para la identificación de la relación de ambas entidades nombradas. Cuando el nodo ADJ tenga como ancestro un nodo NOUN con una relación de dependencia *amod* o *nmod* se toma la información del nodo NOUN. En cambio los descendientes del nodo ADJ pueden ser los nodos NOUN, ADV, ADJ o ninguno de ellos, cada nodo con su respectiva relación de dependencia como se describe el la Figura 4.22.

Los ejemplos presentados a continuación ejemplifican de forma más clara el proceso. En el ejemplo 1 la oración *"Andrés Manuel López Obrador , virtual presidente electo , aceptó las críticas realizadas en torno a la designación de Manuel Bartlett como titular de la Comisión*

Federal de Electricidad (CFE) , en conferencia_de_prensa_EVT realizada en su oficina de la colonia_Roma_GPE .” tiene el árbol de dependencias parcial representado en la Figura 4.23.

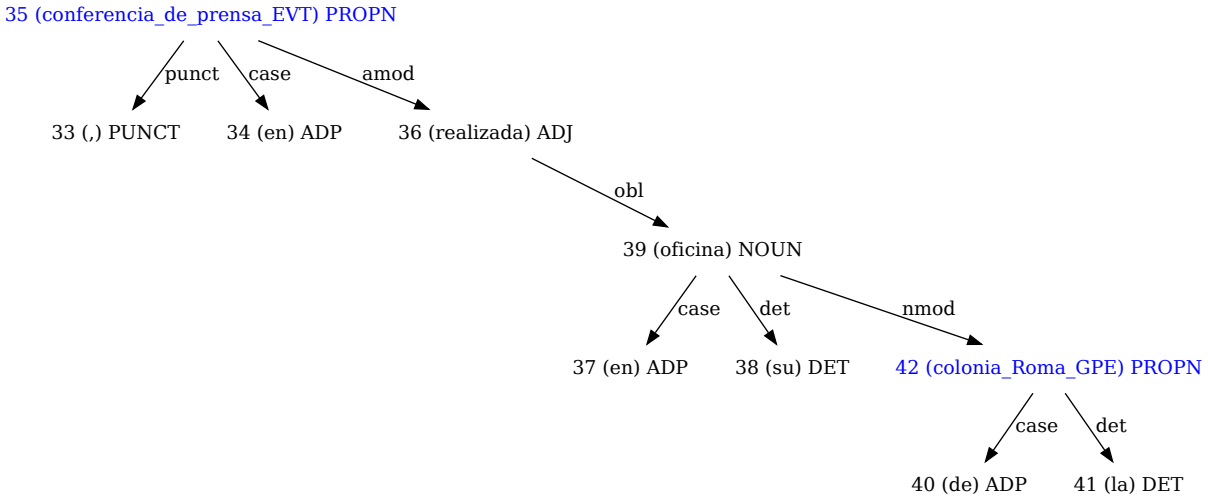

Fig. 4.23 Ejemplo 1 del árbol de dependencias usando la relación *amod*.

Del ejemplo se busca la relación de dependencia *“amod”* entre ambas entidades nombradas. Siguiendo el proceso se elige el nodo (*realizada*) ADJ. A continuación se analiza el nodo descendiente (*oficina*) NOUN que contiene la relación de dependencia *obl* cumpliendo así los requisitos establecidos. Del nodo (*oficina*) NOUN se obtienen sus descendientes. En el último paso se analizan los descendientes de E2. En este caso los nodos (*del*) ADP y (*la*) DET satisfacen el proceso de identificación mostrado en la Figura 4.23. Se extraen los nodos identificados y se forma la relación entre entidades: *“realizada en su oficina de la”*. La tripleta se forma con las entidades nombradas (<*conferencia_de_prensa_EVT, realizada en su oficina de la, colonia_Roma_GPE*>) y se almacena.

En el siguiente ejemplo para la oración *“Dado lo anterior , se deja firme la constancia_de_mayoría_DOC y validez expedida a favor de la candidata_suplente_TIT , Claudia Tello Espinosa , quien deberá asumir el cargo de diputada federal .”* se describe su árbol de dependencias de forma parcial en la Figura 4.24.

Una vez encontrada la relación de dependencia *“amod”* que es el nodo (*expedida*) ADJ. Se analiza el ancestro que es el nodo (*validez*) NOUN y cumple con la relación de dependencia *amod*. Del nodo (*validez*) NOUN se analizan sus descendientes, en este caso solo se tiene el nodo (*y*) CONJ. En el ejemplo se analizan los nodos descendientes de E2, en cuyo caso se extrae el nodo (*a*) ADP y a su vez cumple con los requisitos establecidos en la Figura 4.24, donde se describe las relaciones de dependencia *fixed*. Por esta razón los nodos (*favor*) NOUN

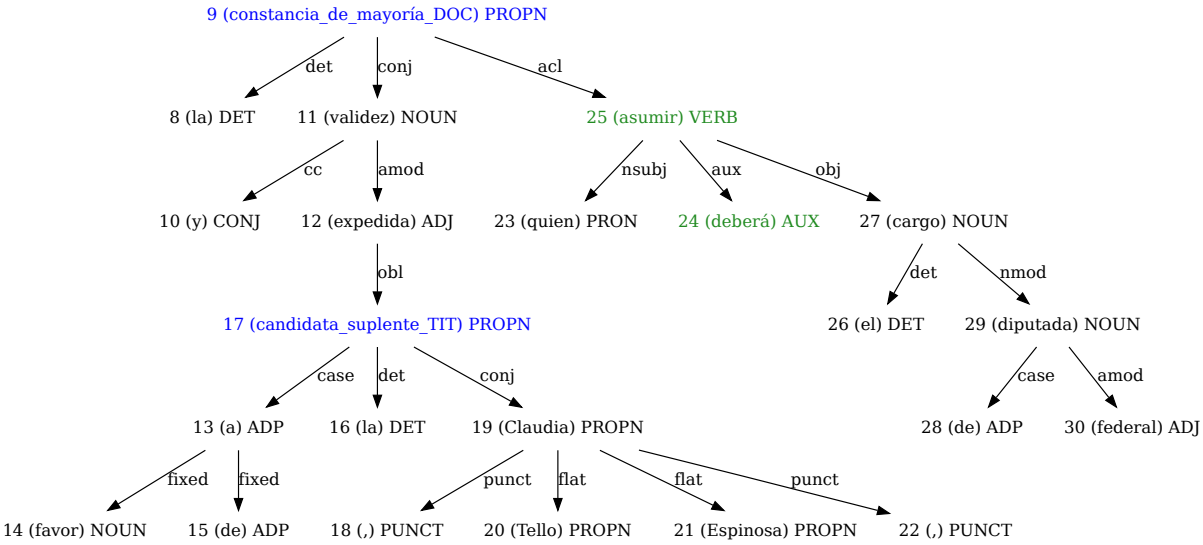

Fig. 4.24 Ejemplo 2 del árbol de dependencias usando la relación *amod*.

y (*de*) ADP son identificados para posteriormente ser extraídos. El nodo (*la*) DET con la relación de dependencia *det* también se identifica. Finalmente se extrae el texto de los nodos identificados para formar la relación entre las dos entidades: *"y validez expedida a favor de la"*. Por lo tanto la tripleta (<*constancia_de_mayoría_DOC, y validez expedida a favor de la, candidata_suplente_TIT*>) se forma y se almacena.

Relaciones con al menos un verbo entre ambas entidades

El procedimiento en este método consiste en identificar al menos un nodo POS del tipo VERB. Este nodo se busca entre la ruta simple dentro del grafo de las dos entidades nombradas. La búsqueda inicia en E2 hacia E1. Una vez encontrado el nodo VERB este se fijara como el nodo central del método. Como se muestra en la Figura 4.25 puede ocurrir que el ancestro del nodo principal sea también un nodo POS de tipo VERB*, se identifica el nodo siempre y cuando cumpla con alguna de las relaciones de dependencia *ccomp*, *xcomp* o *advcl*. Por otro lado cuando se analizan los nodos descendientes del nodo VERB, existen varias opciones que pueden aportar información.

A su vez la Figura 4.26 complementa los nodos y relaciones de dependencia que se deben cumplir para identificar más nodos, y de esta forma formar la relación que existe entre ambas entidades.

Se describen dos ejemplos para encontrar la relación entre dos entidades nombradas, esto enfocándose en el verbo que existe entre ambas entidades.

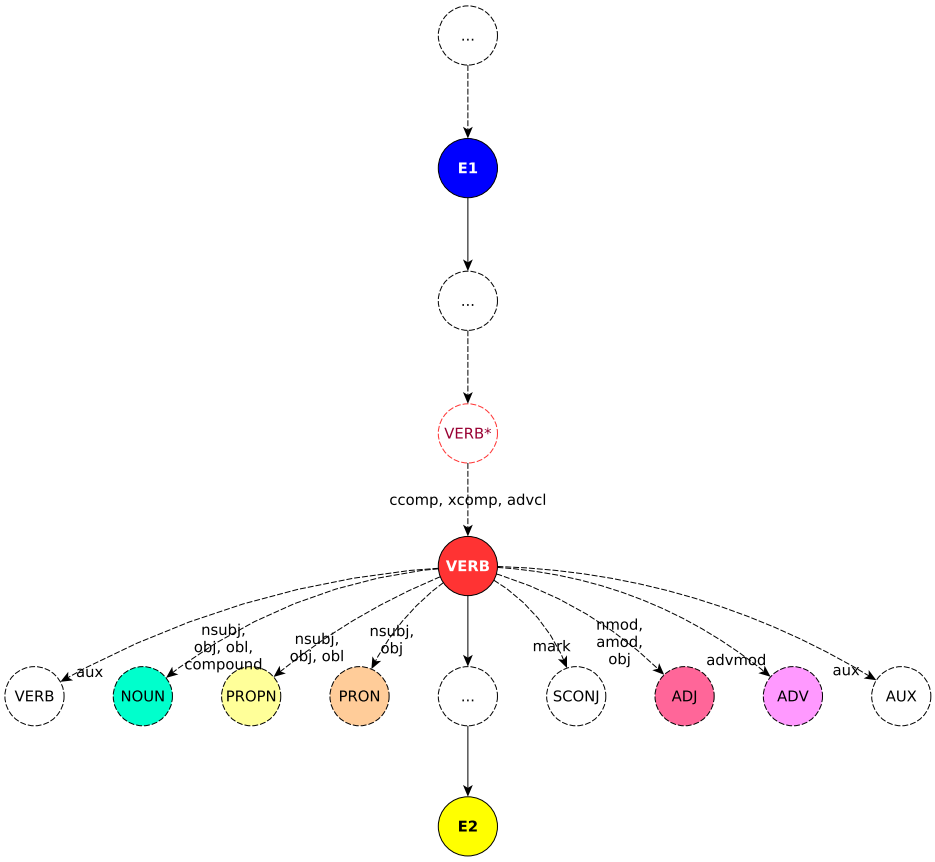

Fig. 4.25 Estructura para identificar relaciones con un verbo entre las entidades nombradas.

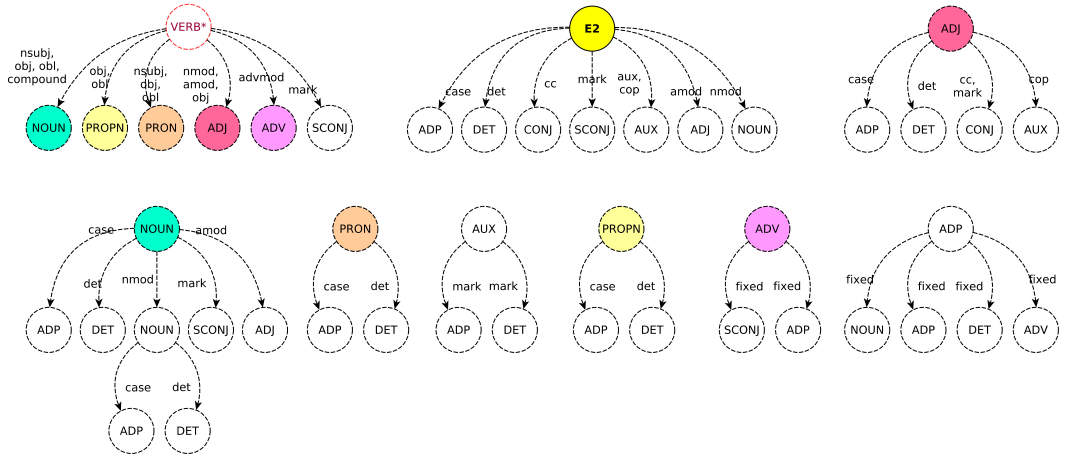

Fig. 4.26 Estructura complemento para identificar relaciones con un verbo entre las entidades nombradas.

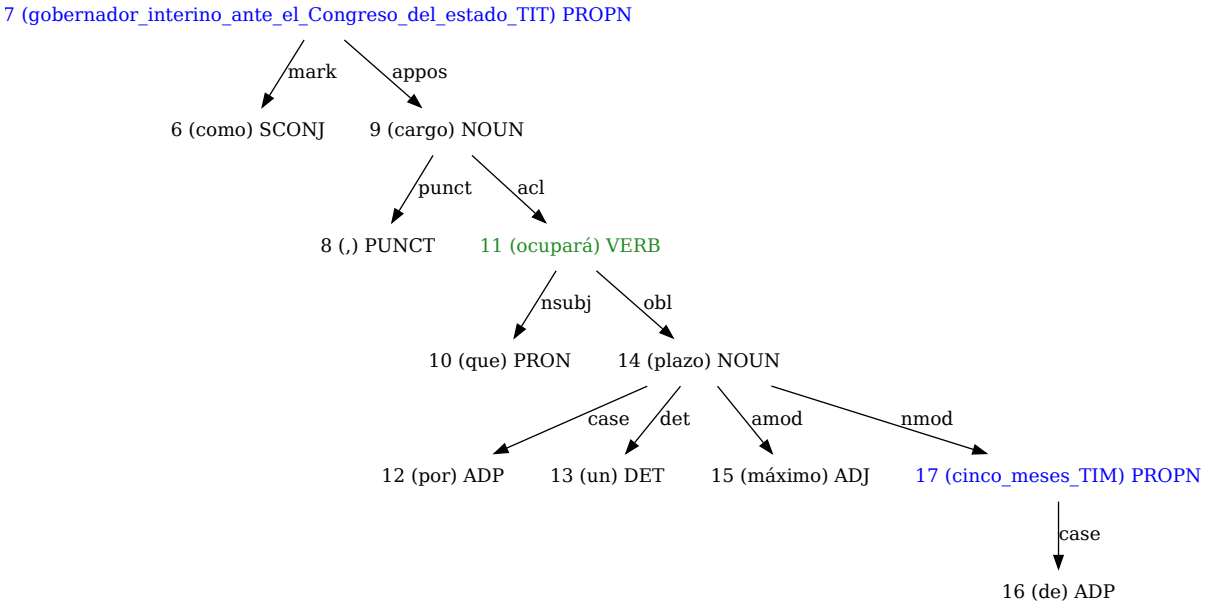

Fig. 4.27 Ejemplo 1 del árbol de dependencias parcial de relaciones con un verbo entre ellas.

El primer ejemplo consiste de la oración *"Guillermo Pacheco Pulido rindió protesta como gobernador_interino_ante_el_Congreso_del_estado_TIT , cargo que ocupará por un plazo máximo de cinco_meses_TIM ."* cuyo árbol parcial de dependencias se puede observar en la Figura 4.27, donde se observa el nodo POS del tipo VERB de color verde. Siguiendo el procedimiento tomando como centro al nodo (*ocupará*) VERB, se observa que no tiene un ancestro de tipo VERB. En el paso siguiente se obtienen los nodos descendientes del nodo central, en este caso (*que*) PRON (pronombre) y el nodo (*plazo*) NOUN. Del nodo NOUN como se describe en la Figura 4.26 se pueden identificar más nodos: (*por*) ADP, (*un*) DET y (*máximo*) ADJ. El procedimiento continua analizando E2, en este ejemplo posee un descendiente el nodo (*de*) ADP. Una vez identificados los nodos se procede a construir la relación extrayendo el texto de cada uno de los nodos previamente identificados. La relación entre entidades es *"que ocupará por un plazo máximo de"*. La tripleta (<*gobernador_interino_ante_el_Congreso_del_estado_TIT, que ocupará por un plazo máximo de, cinco_meses_TIM*>) formada se almacena.

La Figura 4.28 ilustra el árbol de dependencias parcial de la oración del segundo ejemplo: *"El Gobierno de México decidió este viernes no firmar la última declaración del Grupo_de_Lima_ORG , en el que los países miembros acordaron no reconocer la legitimidad de un nuevo Gobierno_ORG de Nicolás Maduro y lo instaron a no efectuar la asunción del mando el próximo jueves ."* El proceso consiste en identificar el primer nodo POS de tipo

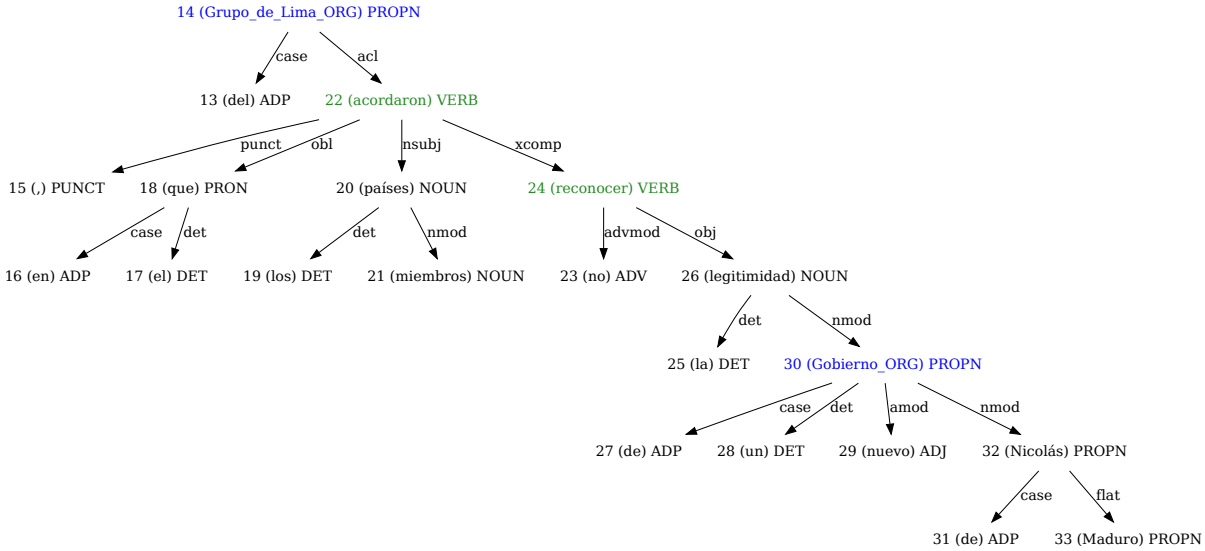

Fig. 4.28 Ejemplo 2 del árbol de dependencias parcial de relaciones con un verbo entre ellas.

VERB iniciando de E2 hacia E1. Del ejemplo se identifica el nodo (*reconocer*) VERB como el nodo principal y del cual se identificaran los nodos que lo rodean. El primer paso es verificar el ancestro del nodo principal, se observa que es el nodo (*acordaron*) VERB con una relación de dependencia *xcomp* justo como se establece en el método. El nodo (*acordaron*) VERB se identifica y se procede a analizar sus descendientes y de cumplir lo establecido en la Figura 4.26 del nodo VERB*. Los descendientes que cumplen los requisitos son (*que*) PRON y (*países*) NOUN. A su vez se verifican los descendientes del nodo PRON y NOUN identificando los nodos (*en*) ADP, (*el*) DET y (*los*) DET respectivamente.

Lo siguiente es analizar los descendientes del nodo central VERB. Se identifica el nodo (*legitimidad*) NOUN y su descendiente el nodo (*no*) ADV. El paso final es analizar si E2 contiene descendientes. En este caso se identifican los nodos (*de*) ADP, (*un*) y (*nuevo*) ADJ. Se extrae el texto de los nodos identificados formando la relación entre entidades nombradas *"en el que los países acordaron no reconocer la legitimidad de un nuevo"*. La tripleta formada por ambas entidades y relación <*Grupo_de_Lima_ORG, en el que los países acordaron no reconocer la legitimidad de un nuevo, Gobierno_ORG*> se almacena y se continua con la siguiente oración.

Relaciones que tienen como ancestro un verbo y recaen en la Entidad 1

El método para identificar extraer relaciones de dos entidades se centra en el procedimiento de identificar un verbo como ancestro de ambas entidades nombradas. En la Figura 4.29 se

describe la estructura para poder identificar y extraer relaciones entre entidades. El primer paso es realizar una búsqueda simple del nodo E1 y el nodo E2 dentro del grafo hacia el nodo raíz (ROOT). Después se verifica que exista un nodo POS de tipo VERB que intersecte el camino de ambas rutas simples de E1 y E2, es decir que compartan el mismo nodo ancestro y este sea de tipo VERB. A continuación se valida que exista una relación de dependencia *nsubj* hacia E1, y una relación *obj* o *obl* hacia E2.

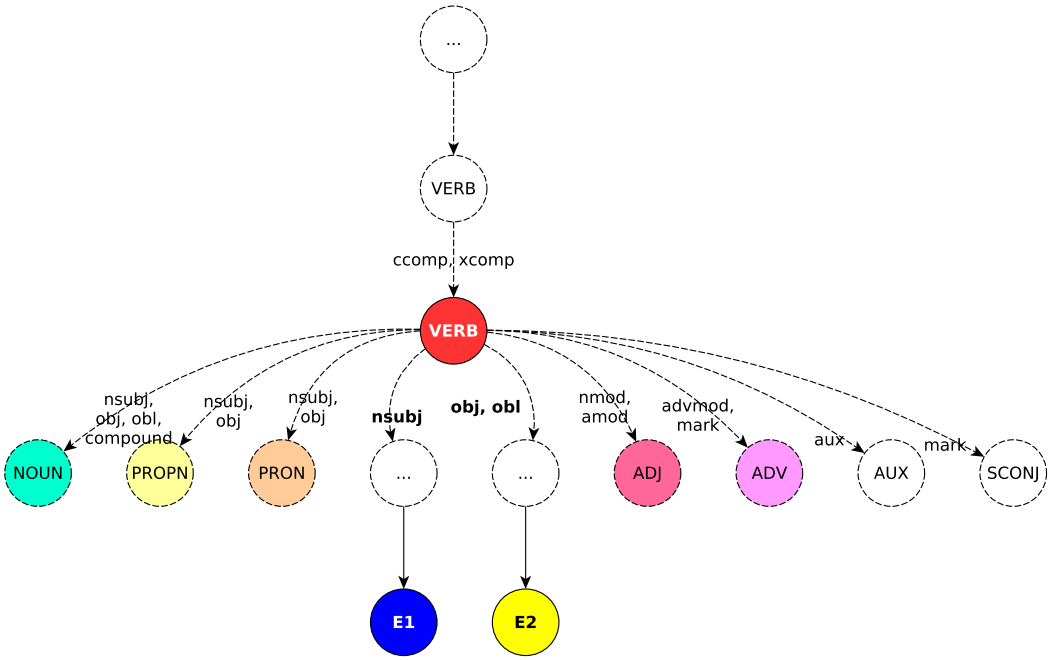

Fig. 4.29 Estructura para identificar relaciones con un verbo como ancestro de E1.

Si se cumple lo anterior el nodo VERB se identifica como el nodo principal. Partiendo del nodo VERB se analiza si su ancestro es un nodo POS del tipo VERB con alguna de las relaciones de dependencia permitidas (*ccomp* o *xcomp*). De ser así se identifica el nodo ancestro VERB.

Lo siguiente es analizar los descendientes del nodo principal VERB. Cuando la relación de dependencia ligada del nodo principal hacia sus descendientes se cumple. Se procede a analizar el nodo en cuestión siguiendo los lineamientos de la Figura 4.30. Las estructuras mostradas en la Figura 4.30 complementan las condiciones establecidas para poder identificar más nodos. Se puede seguir la estructura correspondiente del nodo siguiendo su color de la Figura 4.29 a la Figura 4.30.

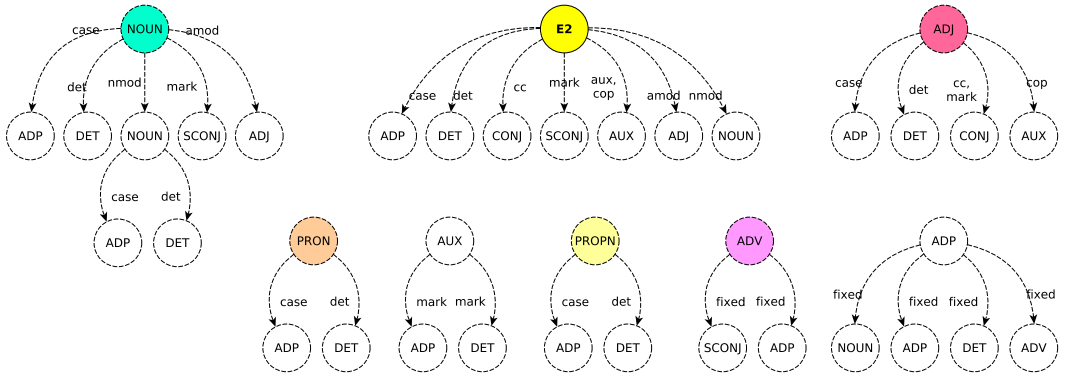

Fig. 4.30 Complemento de la estructura para identificar relaciones con un verbo como ancestro de E1.

Un ejemplo se presenta para ilustrar el procedimiento de este método. De la oración *"También precisó , que sí habrá filtros carreteros , ya que este martes , la Secretaría_de_Salud_ORG se reunirán con el comandante de la Guardia_Nacional_ORG para definir cómo se trabajará con ellos para establecer estás acciones , y con ello , estar realizando los trabajos ."* se obtiene su árbol de dependencias parcial y se visualiza en la Figura 4.31. Como se observa en la Figura 4.31 el nodo (*reunirán*) VERB es el nodo sobre el que se centrara el análisis. La relación de dependencia del nodo (*reunirán*) VERB hacia E1 es *nsubj* y la relación de dependencia hacia E2 es *obl*. Se cumple la base para analizar la estructura en busca la relación entre las dos entidades nombradas.

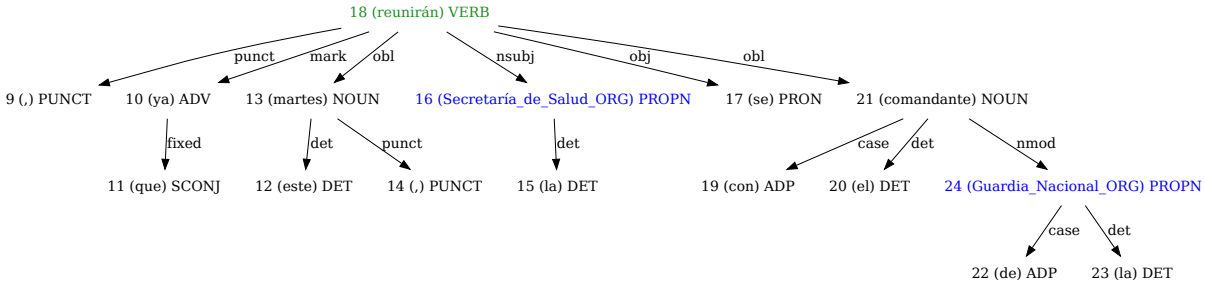

Fig. 4.31 Ejemplo 1 del árbol de dependencias parcial para relaciones con un verbo como ancestro de E1.

Partiendo el análisis desde el nodo VERB se observan los nodos descendientes (*ya*) ADV, (*martes*) NOUN, (*se*) PRON y (*comandante*) NOUN. Todos estos nodos son identificados pues cumplen con lo establecido previamente. Se revisan los nodos descendientes de los nodos identificados, además de los nodos descendientes de E2 siguiendo la estructura de la Figura 4.30.

La relación resultante es *"ya que este martes se reunirán con el comandante de la"*. La tripleta entonces es <*Secretaría_de_Salud_ORG, ya que este martes se reunirán con el comandante de la, Guardia_Nacional_ORG*>.

El segundo ejemplo contiene la oración *"De cantante y actor a diputado , no cabe duda que el mundo de la farándula está dando de qué hablar en la Cámara de Diputados y es que ahora Ernesto_D_Alessio_PER ha sido asignado a la comisión del Deporte por el Partido_Encuentro_Social_PEX ."* y su árbol de dependencias parcial se observa en la Figura 4.32. Como paso inicial se busca un nodo POS de tipo VERB que compartan ambas entidades en su ruta de nodos hacia la raíz. Después se verifica que VERB tenga una relación de dependencia *nsubj* hacia E1 y una relación de dependencia *obj* u *obl* hacia E2. Tras cumplirse estas premisas se procede a analizar los nodos descendientes del nodo (*asignado*) VERB y son: (*que*) SCONJ (conjunción subordinada), (*ahora*) ADV, (*ha*) AUX, (*sido*) AUX y (*comision*) NOUN. El nodo con descendientes a identificar es (*comision*) NOUN así como los descendientes de E2.

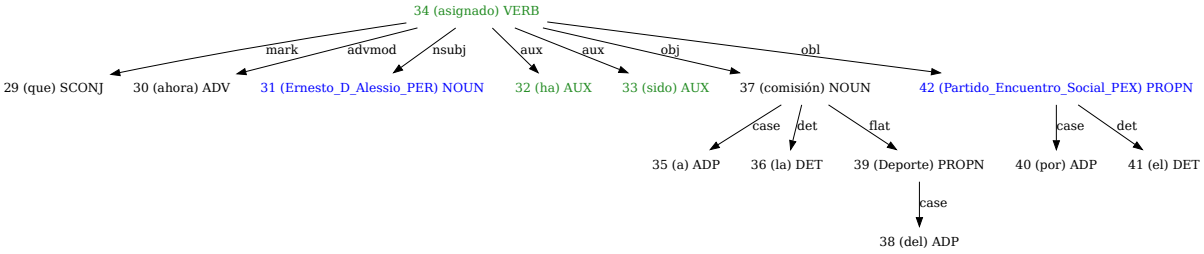

Fig. 4.32 Ejemplo 2 del árbol de dependencias parcial de relaciones con un verbo como ancestro de E1.

Una vez identificados los nodos se extrae el respectivo texto de cada uno y se construye la relación siguiendo el orden (*id*). La relación es *"que ahora ha sido asignado a la comisión por el"* y la tripleta (<*Ernesto_D_Alessio_PER, que ahora ha sido asignado a la comisión por el, Partido_Encuentro_Social_PEX*>) formada se almacena.

Relaciones que tienen como ancestro un verbo y recaen en la Entidad 2

Este es el ultimo método propuesto para identificar y extraer relaciones de dos entidades. El procedimiento es muy similar al método previo con algunas excepciones. La base del método igualmente consiste en encontrar un nodo POS del tipo VERB que compartan ambas entidades en su ruta de nodos hacia la raíz, siempre iniciando de E1 hacia la raíz y E2 hacia la raíz.

La diferencia yace en las relaciones de dependencia. En este método el nodo VERB debe tener una relación de dependencia *obj* o *obl* hacia E1 y la relación de dependencia *nsubj* hacia E2, como se ilustra en la Figura 4.33.

En este método se busca identificar la relación que existe entre dos entidades, asumiendo que E2 será la primer entidad en la tripleta resultante y E1 será la segunda, por ello se ha intercambiado el order de las relaciones de dependencia del nodo principal hacia las entidades. Bajo estas condiciones se estableció otro cambio. No se buscara identificar nodos descendientes en E2, ahora esta tarea se asigna a E1 como se observa en la Figura 4.34.

El primer ejemplo describe el árbol de dependencias parcial en la Figura 4.35 perteneciente a la oración *"Desde la Tribuna_ORG , Batres_Guadarrama_PER reiteró su convicción de que Villamil Rodríguez conoce la problemática de los medios públicos y que se ha conducido con ética , honradez intelectual y profesionalismo en su actividad pública ."*

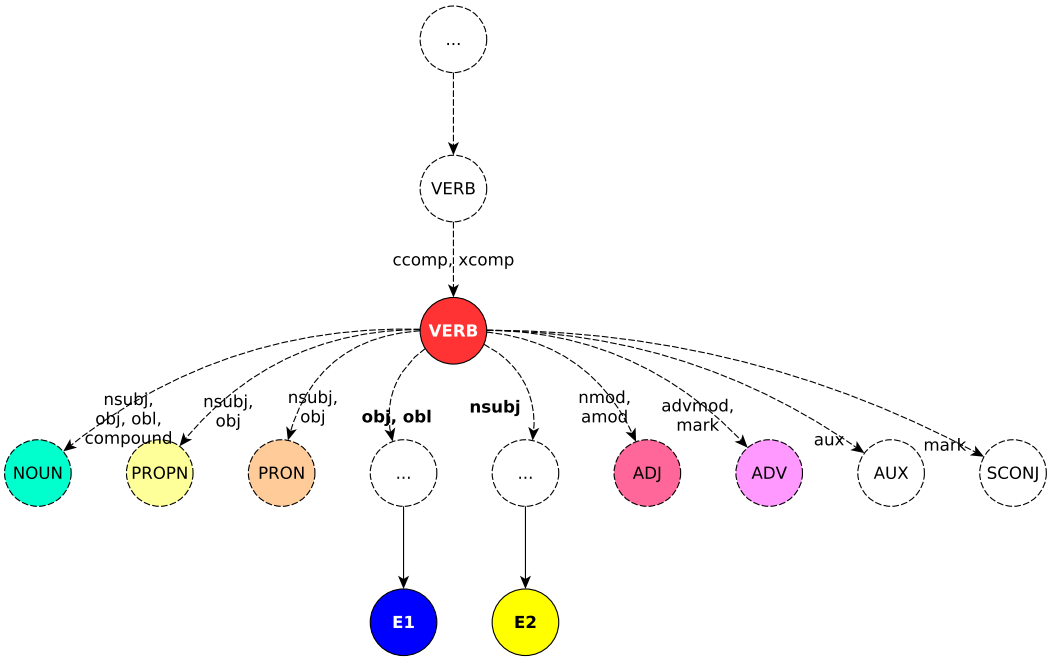

Fig. 4.33 Estructura para identificar relaciones con un verbo como ancestro de E2.

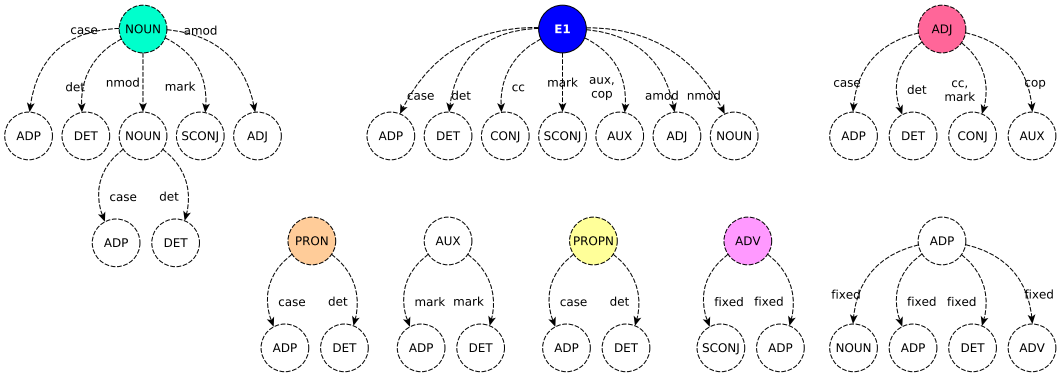

Fig. 4.34 Complemento de la estructura para identificar relaciones con un verbo como ancestro de E2.

Como se observa en la Figura 4.35 el nodo principal del que se parte el análisis es (*reiteró*) VERB. El nodo (*convicción*) NOUN cumple las condiciones dadas al mismo tiempo que su descendiente el nodo (*su*) DET, estos nodos se identifican. Después se procede a revisar los nodos descendientes de E1 que son los nodos (*Desde*) ADP y (*la*) DET, se identifican los nodos.

Para extraer la relación de dos entidades nombradas en primera instancia, se extrae el texto de los nodos que son descendientes directos (hijos) del nodo VERB así como los descendientes de los descendientes (nietos), la cadena se forma en orden ascendente en base a su *id* (orden de aparición en el texto de la oración). Como segundo paso se extrae el texto de los descendientes de E1, formando una cadena ordenada en forma ascendente basándose en el *id* de los nodos. Finalmente la relación se forma concatenando la cadena resultante del primer paso con la cadena del segundo paso. Así para este ejemplo, se obtiene la relación entre dos entidades: relación *"reiteró su convicción Desde la"*. Como se ha explicado previamente la tripleta se forma <*entidad2, relación, entidad1*> por lo que de este ejemplo se tiene <*Batres_Guadarrama_PER, reiteró su convicción Desde la, Tribuna_ORG*>.

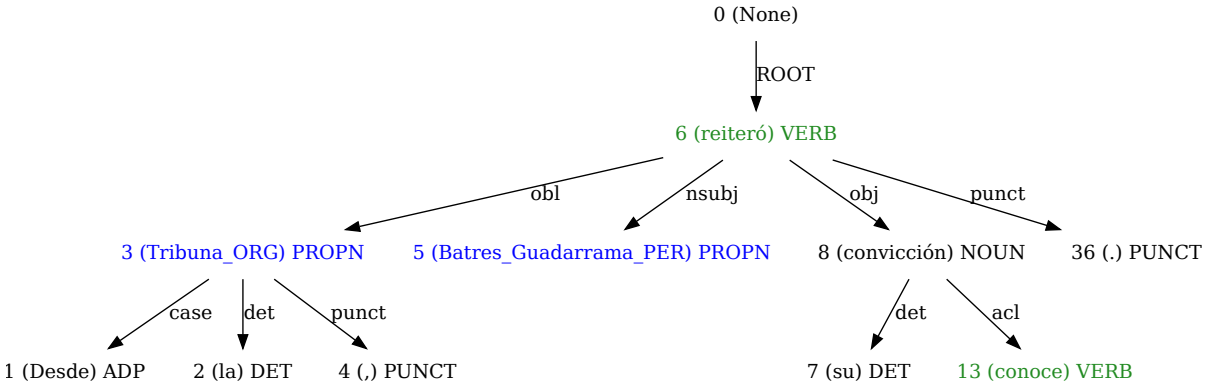

Fig. 4.35 Ejemplo 1 del árbol de dependencias parcial de relaciones con un verbo como ancestro de E2.

Para el segundo ejemplo se presenta la oración *"Con especial énfasis , desde el 1_de_diciembre_de_2018_DAT , cuando asumió el poder , López_Obrador_PER ha desestimado datos de corte económico que sugieren la desaceleración de la economía mexicana , principalmente provenientes de instituciones como el Inegi y el Banco de México o de corporaciones de renombre como Bank of America , Moody s o el mismo Fondo Monetario Internacional ."* y su árbol de dependencias parcial se observa en la Figura 4.36. En el ejemplo el nodo principal del cual parte el análisis es el nodo (*desestimado*) VERB. Los nodos que cumplen para ser identificados son (*énfasis*) NOUN, (*ha*) AUX y (*datos*) NOUN así como sus nodos descendientes. Posteriormente se identifican los descendientes de E1. Todo los identificados cumplen con la estructura presentada en las Figuras 4.33 y 4.34.

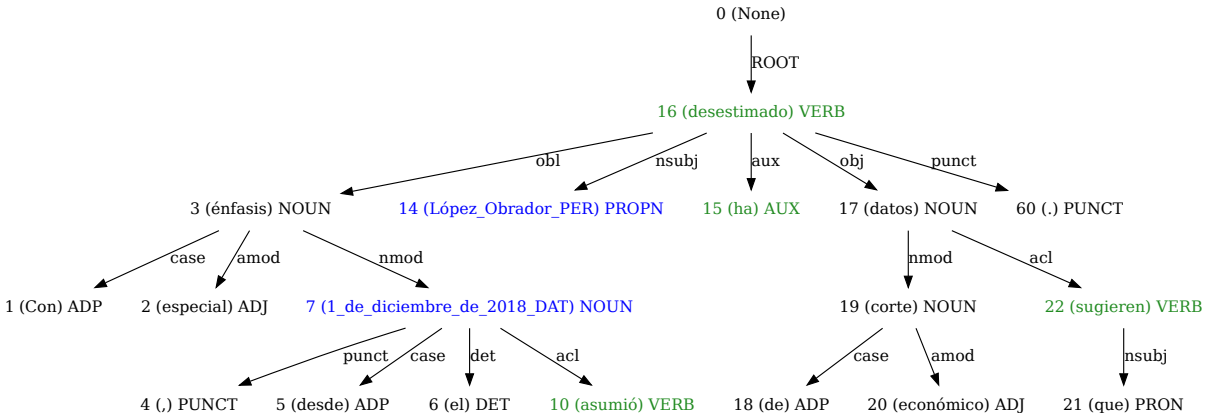

Fig. 4.36 Ejemplo 2 del árbol de dependencias parcial de relaciones con un verbo como ancestro de E2.

La extracción del texto para formar la relación se hace en dos faces. Primero se extrae el texto de todos los nodos identificados como descendientes y/o ancestro del nodo principal VERB, con excepción de los nodos descendientes de E1. El texto forma una cadena ordenada ascendentemente en base al *id*. La segunda fase consiste de extraer el texto de los nodos descendientes identificados de E1. Se crea una cadena ordenando las palabras en base a su *id* de forma ascendente. Finalmente se concatenan ambas cadenas, la cadena obtenida de la primera fase concatenada con la cadena de la segunda fase. La relación resultante *"Con especial énfasis ha desestimado datos desde el"*. Entonces la tripleta final es *<López_Obrador_PER, Con especial énfasis ha desestimado datos desde el, 1_de_diciembre_de_2018_DAT>*.

4.3 Base de hechos y Reglas lógicas

En esta sección se realiza la evaluación manual de las tripletas que previamente se extrajeron y se almacenaron en una base de datos. Después de la evaluación se "traducen" las tripletas a hechos siguiendo la sintaxis de Prolog. Además se definen reglas de forma genérica para los hechos obtenidos previamente. Una vez terminada la base de hechos se evalúan las reglas definidas. De los resultados (hechos) obtenidos al ejecutar las reglas se realiza un análisis sobre las oraciones. Lo anterior descrito se observa en la Figura 4.38 que muestra la metodología general.

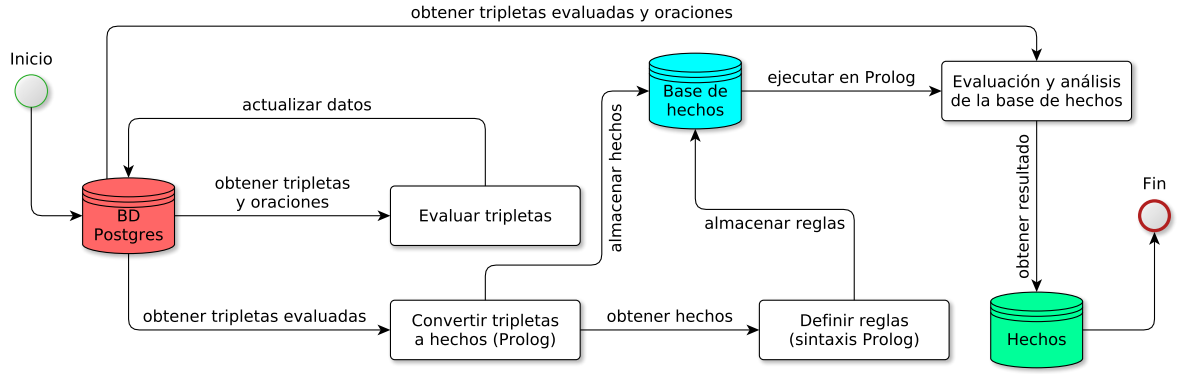

Fig. 4.37 Metodología para Creación de la base de hechos y reglas.

4.3.1 Evaluación de tripletas

La tripletas se encuentran almacenadas en la base de datos, existe una tabla que contiene todas las oraciones usadas para la extracción de relaciones, las oraciones cuentan con un *id* único así como el *id* del documento del que provienen. Otra tabla que contiene registros sobre cada tripleta llamada relaciones, donde cada tripleta contiene un *id* y no existen tripletas duplicadas. La tabla relaciones se encuentra asociada a la tabla oraciones. Una tripleta puede estar relacionada con una o más oraciones.

Fig. 4.38 Interfaz para la evaluación de tripletas.

La evaluación consiste en obtener la tripleta a evaluar y la oración u oraciones relacionadas a esta tripleta. Los elementos que componen la tripleta se evalúa por separado, la *primer entidad*, la *relación de las entidades* y la *segunda entidad*. Además se muestra las oración relacionada a la tripleta, en el caso de haber más de una oración se muestran como máximo 10 oraciones. Las oraciones son necesarias para evaluar si la relación fue correctamente identificada y extraída a criterio del evaluador. La Figura 4.38 describe de forma general la interfaz usada para la evaluación.

4.3.2 Convertir tripletas a hechos

El conjunto de tripletas extraídas previamente y almacenadas en la base de datos son empleadas para construir una base de hechos de forma automática. De las tripletas (<*Entidad, relación, Entidad*>) se procede a transformar cada uno de los elementos en hechos según la sintaxis de Prolog. El proceso se muestra con las siguientes tripletas de ejemplo:

1. <presidente_electo_TIT, *es el título de*, Andrés_Manuel_López_Obrador_PER>
2. <Instituto_Nacional_Electoral_ORG, *tiene el acrónimo de*, INE_ORG>
3. <Martha_Erika_Alonso_PER, *de la coalición*, Por_Puebla_al_Frente_PEX>
4. <Presidente_electo_TIT, *es el título de*, Andrés_Manuel_López_Obrador_PER>

Previo a convertir las tripletas a hechos se realiza un pre-procesamiento. Consiste en realizar una limpieza de las tripletas removiendo acentos, símbolos e intercambiando la letra *ñ* a *n*. De acuerdo a la sintaxis de Prolog los hechos deben estar en minúsculas por lo que se aplica esta transformación a las tripletas. Después del proceso pueden existir tripletas con duplicados, estos se omiten. Este caso ocurre con la tripleta número 1 y 4 son exactamente las misma con excepción de la primer letra de *presidente* y *Presidente* respectivamente. El resultado del pre-procesamiento se describe en el siguiente listado de tripletas.

1. <presidente electo tit, *es el titulo de*, andres manuel lopez obrador per>
2. <instituto nacional electoral org, *tiene el acronimo de*, ine org>
3. <martha erika alonso per, *de la coalicion*, por puebla al frente pex>

El paso final es convertir la tripleta a hechos según los lineamientos de Prolog. La *clase* de la entidad es usada para definir el *término* de cada entidad. Para finalizar se definen *términos compuestos* que contienen a su vez tres términos: *primera entidad*, *segunda entidad* y *relación*. La definición sigue la sintaxis: *triple(claseEnt1("entidad1"),claseEnt2("entidad2"),relacion("relacion")).* A continuación se enumeran el resultado de este proceso.

1. **triple**(**titulo**("presidente electo"),**persona**("andres manuel lopez obrador"),**relacion**("es el titulo de")).
2. **triple**(**organizacion**("instituto nacional electoral"),**organizacion**("ine"),**relacion**("tiene el acronimo de")).
3. **triple**(**persona**(mrtha erika alonso),**partidopolitico**("por puebla al frente"),**relacion**("de la coalicion")).

4.3.3 Definición de reglas

En esta sección se definen un conjunto de reglas de forma genérica, es decir, son reglas que se diseñan para obtener un resultado no específico. Por ejemplo, si la base de hechos consiste de información sobre los integrantes de una familia, una regla específica podría ser aquella para identificar a la *madre* de cualquier miembro. En cambio una regla genérica podría enfocarse en "descubrir" información sobre dos miembros que a simple vista no presenten un vínculo.

El conjunto de reglas se define siguiendo la sintaxis de Prolog. Las reglas se definen en el mismo archivo a continuación de la base de hechos. El archivo se carga en Prolog, que se encarga de verificar la sintaxis.

4.3.4 Evaluación y análisis de la base de hechos

Las reglas definidas previamente son ejecutadas en SWI Prolog. Los hechos obtenidos se almacenan en forma de tripleta para su posterior análisis. Este análisis se realiza con los elementos que componen a los hechos obtenidos, las oraciones donde están presentes (*entidades nombradas* y *relaciones*).

4.4 Grafo de Conocimiento

Para la construcción del grafo de conocimiento se utiliza el conjunto de hechos obtenidos tras la ejecución de la reglas en SWI Prolog. Las tripletas son empleadas siguiendo los lineamientos establecidos por la W3C para la construcción de un grafo de conocimiento, donde los *nodos* son las entidades nombradas y las *aristas* las relaciones. Además se toman algunos esquemas de bases de conocimiento existentes, y algunos esquemas se definen. Finalmente se visualiza el grafo de conocimiento como se ilustra en la Figura 4.39.

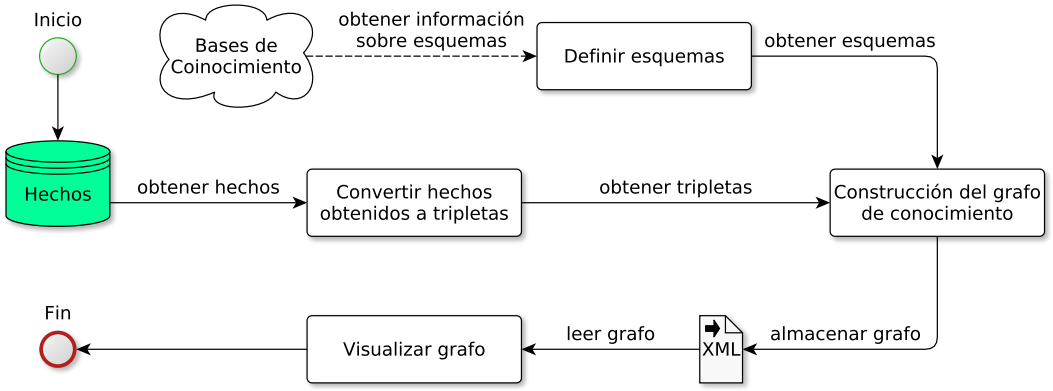

Fig. 4.39 Metodología para la construcción del grafo de conocimiento.

4.4.1 Convertir hechos a tripletas

De cada hecho (tripleta) obtenido se obtienen sus elementos, *entidad 1* y su *clase*, *relación* y *entidad 2* y su *clase*. Posteriormente cada elemento formara una tripleta siguiendo las normas de la W3C usando RDF (Resource Description Framework)[7].

4.4.2 Definir de esquemas

Un esquema es un lenguaje para expresar restricciones sobre un documento XML. Hay varios lenguajes de esquema diferentes de uso generalizado. Un esquema puede ser usado para:

- Proporcionar una lista de elemntos y atributos en un vocabulario
- Asociar tipos como *integer*, *string*, etc., o más específicamente como *hat_size*, *sock_colour*, etc., con valores encontrados en documentos
- Restringir dónde pueden aparecer los elementos y atributos, y qué puede aparecer dentro de esos elementos, como decir que el título de un capítulo ocurre dentro de un capítulo, y que un capítulo debe constar de un título de capítulo seguido de uno o más párrafos de texto
- Proporcionar documentación que sea legible por humanos y procesable por máquinas
- Dar una descripción formal de uno o más documentos

[7]https://www.w3.org/TR/rdf11-concepts/

La Tabla 4.7 Describe los esquemas utilizados para la construcción del grafo de conocimiento. En la primer columna se listan los prefijos correspondientes, en la segunda columna se encuentran los *namespaces* IRI (Internationalized Resource Identifier). Por último la tercera columna describe los esquemas.

Tabla 4.7 Prefijos, IRI y descripción de los *namespaces* empleados.

Prefijo	**IRI**	**Descripción**
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Vocabulario incorporado en RDF
foaf	http://xmlns.com/foaf/0.1/	Especificación FOAF del Vocabulario de clases y propiedades nombradas
schema	https://schema.org/	Promueve esquemas para datos estructurados en Internet, en páginas web, en mensajes de correo electrónico y más.
ns1	http://mx-kg.com/	Define una entidad nombrada o relación del conjunto de datos de noticias

4.4.3 Construcción del grafo de conocimiento

La construcción es realizada empleando la biblioteca de Pyhton rdflib[8] (versión 5.0.0) definiendo cada una de las tripletas con base a los lineamientos de la W3C. La estructura del archivo RDF es definida como sigue. Dentro de cada entidad se establece la clase a la que pertenece y su nombre es definido como *literal* (atributo). Para vincular una entidad hacia otra a través de la relación, se define la relación en el nodo origen (primer entidad) con dirección al nodo destino (segunda entidad). Los nodos se definen de la siguiente forma:

[8]https://rdflib.readthedocs.io/en/stable/

```
1:  <?xml version="1.0"?>
2:  <rdf:RDF
3:    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:    xmlns:foaf="http://xmlns.com/foaf/0.1/"
5:    xmlns:lke="http://lke.buap.mx/0.1/"
6:  >
7:    <!-- definición de un nodo -->
8:    <rdf:Description rdf:about="http://lke.buap.mx/0.1/clase_entidad/nombre_entidad">
9:      <rdf:type rdf:resource="http://dbpedia.org/ontology/Clase"/>
10:     <foaf:name xml:lang="es">nombre de la entidad</foaf:name>
11:     <!-- definición de la relación -->
12:     <lke:relacion rdf:resource="http://lke.buap.mx/0.1/clase_entidad/nombre_entidad"/>
13:   </rdf:Description>
14: </rdf:RDF>
```

Fig. 4.40 Definición de un nodo en RDF.

De la Figura 4.40 la línea 1 indica que el tipo de archivo (XML) y la versión. La línea 2 establece el inicio de la definición RDF. Dentro de esta definición se encuentran los *namespaces* con su prefijo e IRI correspondiente líneas 3 al 5. En la línea 8 se establece un nodo (Description), se especifica el IRI del nodo, que incluye la clase a la que pertenece la entidad nombrada y su nombre. La línea 9 establece el tipo de recurso (clase a la que pertenece el nodo, por ejemplo *Persona, Organización, etc.*). La línea 12 se establece cuando el nodo en cuestión es el nodo origen, se define el prefijo *lke* dos puntos y a continuación el *nombre de la relación* además del nodo destino (IRI clase de la entidad y nombre). La línea 13 indica el cierre de la definición del nodo, y la línea 14 cierra la especificación RDF.

Para clarificar la creación de una tripleta en RDF, un ejemplo se define a continuación con la tripleta conformada por <*presidente* (título), *es_el_título_de, López_Obrador* (persona)>. La Figura 4.41 describe la definición de está tripleta en RDF. La línea 10 establece el IRI del nodo origen seguido de la clase de la entidad **tit** (*título*) y el nombre de la entidad **presidente**. La línea 11 específica el tipo de recurso, en este caso se emplea la definición de DBpedia *Person Function*. El nombre del nodo se define en la línea 12. En la línea 13 se establece la relación *es_el_título_de* que apunta hacia el nodo *López_Obrador*. El nodo destino describe su IRI seguido de la clase **per** (*persona*) y el nombre de la entidad **López_Obrador**. Para el tipo de recurso se utiliza FOAF (Friend Of A Friend) que describe la especificación *Person* en la línea 18. La línea 19 define el nombre del nodo.

```
1:  <?xml version="1.0"?>
2:  <rdf:RDF
3:    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:    xmlns:foaf="http://xmlns.com/foaf/0.1/"
5:    xmlns:lke="http://lke.buap.mx/0.1/"
6:  >
7:
9:    <!-- definición de nodo origen -->
10:   <rdf:Description rdf:about="http://lke.buap.mx/0.1/tit/presidente">
11:     <rdf:type rdf:resource="http://dbpedia.org/ontology/PersonFunction"/>
12:     <foaf:name xml:lang="es">presidente</foaf:name>
13:     <!-- definición de la relación -->
13:     <lke:es_el_título_de rdf:resource="http://lke.buap.mx/0.1/per/López_Obrador"/>
14:   </rdf:Description>
15:
16:   <!-- definición de nodo destino -->
17:   <rdf:Description rdf:about="http://lke.buap.mx/0.1/per/López_Obrador">
18:     <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
19:     <foaf:name xml:lang="es">López Obrador</foaf:name>
20:   </rdf:Description>
21:
22: </rdf:RDF>
```

Fig. 4.41 Ejemplo de tripleta RDF.

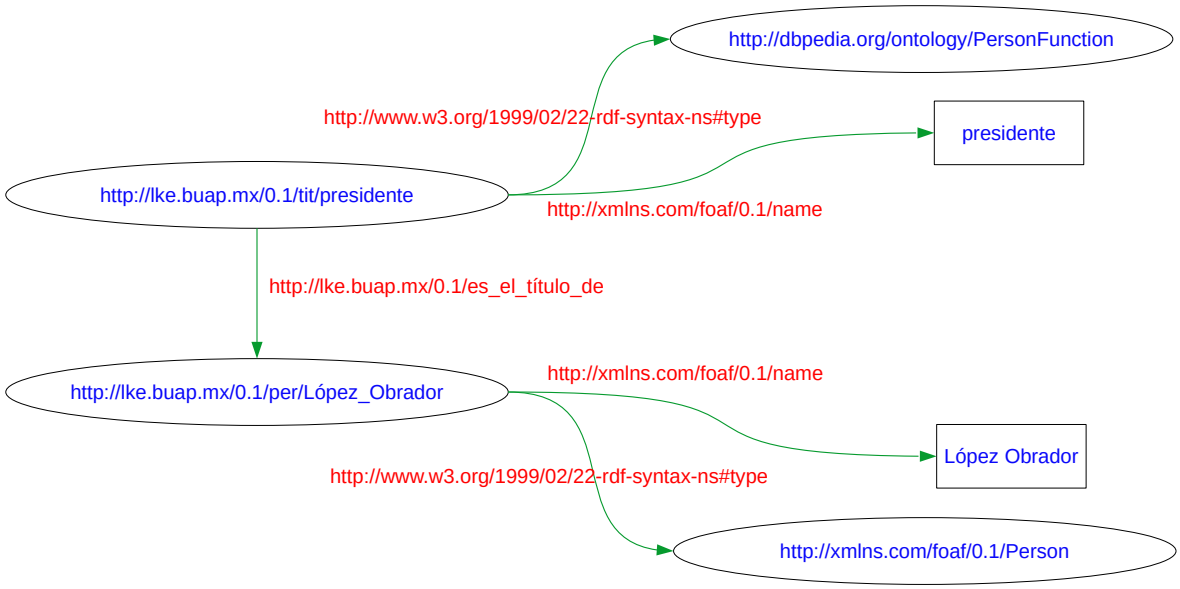

Fig. 4.42 Ejemplo gráfico una tripleta RDF.

La Figura 4.42 ilustra el ejemplo presentado en la Figura 4.41. Las elipses identifican a los nodos, los nodos son aquellos que se han definido para cada una de las entidades nombradas, al mismo tiempo se crean los nodos que pertenece al tipo de recurso de los nodos. Los rectángulos representan las *literales* (atributos) de los nodos, en el ejemplo definen el nombre de de las

entidades nombradas. Por último las líneas son las aristas que apuntan al nodo y/o literal correspondiente para formar cada tripleta en RDF. Cada elemento en la Figura 4.42 es rotulado de forma correspondiente.

El proceso para la creación de tripletas se realiza con todos los hechos obtenidos. Un nodo origen puede tener *n-relaciones* que apuntan a diferentes nodos, o *n-relaciones* dirigidas a un único nodo. Asimismo uno nodo destino puede convertirse en un nodo origen, y de esta manera pueden definirse *n-relaciones* hacia otros nodos dentro de este. De esta forma se crean diferentes conexiones entre nodos a través de sus relaciones para construir un grafo de conocimiento. Todas las tripletas definidas se almacenan en un archivo XML.

4.4.4 Visualización del grafo

Para visualizar el grafo se carga el archivo XML que contiene el conjunto de tripletas definidas bajo las especificaciones de la W3C. La biblioteca de Python *rdflib* es empleada para cargar el archivo. La visualización se realiza mediante la biblioteca de Python *networkx*[9] (version 2.4) que realiza la transformación de tripletas RDF a nodos y aristas.

[9]https://networkx.org/

Capítulo 5

Resultados

En este capítulo se presentan los resultados de los experimentos realizados sobre el Reconocimiento de Entidades Nombradas, utilizando modelos probabilísticos y de redes neuronales recurrentes con ambos conjuntos de datos (CoNLL-2002 y Mx-news). Los resultados obtenidos y que son descritos en las matrices de confusión, corresponden a las evaluaciones sobre las clases (entidades nombradas) y no sobre clases individuales. Los modelos CRF, Bi-LSTM y Bi-LSTM-ELMo fueron usados como se describe en las Secciones 4.1.3, 4.1.3 y 4.1.3 respectivamente. El término *ascendente* será usado para referirse a: los experimentos que inician con la clase más dispersamente etiquetada, adicionando una clase cada vez, hasta llegar a la más densamente etiquetada. Para el caso donde los experimentos inician con la clase más densa, agregando una clase cada vez, hasta terminar con la clase más dispersa se empleará el término *descendente*.

Además se describen los experimentos para la extracción de relaciones entre dos entidades nombradas, utilizando árboles de dependencia para identificar etiquetas POS y dependencias entre los tokens.

5.1 Reconocimiento de Entidades Nombradas

A continuación se describen los experimentos realizados para el reconocimiento de entidades nombradas sobre documentos de noticias políticas.

5.1.1 Resultados sobre el corpus CoNLL-2002

Los resultados de los experimentos sobre el corpus CoNLL-2002 para cada uno de los modelos (CRF, Bi-LSTM y BI-LSTM-ELMo), se describen usando matrices de confusión y gráficas evaluadas sobre la métrica *F1-score*.

La Figura 5.1 muestra las matrices de confusión como resultado de evaluar el *ensemble* del corpus CoNLL-2002, el *ensemble* está formado por todas las particiones del corpus y es el que presenta mejores resultados. Las matriz de la izquierda Figura 5.1 en su diagonal describe el número entidades reconocidas de forma correcta, y los datos que no pertenecen a la diagonal son el número de entidades reconocidas erróneamente. La matriz de la izquierda es usada con la intención de mostrar en términos de porcentaje, el número de entidades reconocidas correctamente y que son encontradas en la diagonal de la matriz.

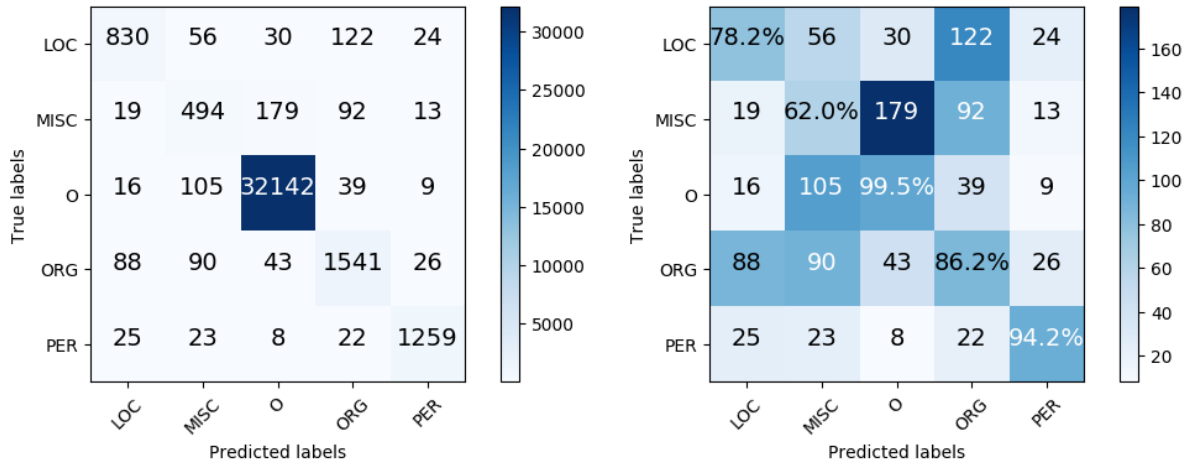

Fig. 5.1 Matrices de confusión obtenidas del modelo CRF sobre CoNLL-2002.

Las matrices de la Figura 5.1 contienen una clase adicional “O”, que es agregada para mostrar el número de clases que se reconocieron erróneamente, es decir no se reconocieron como entidades. La clase con el mayor número de entidades reconocidas es la “ORG”, seguida de las clases “PER”, “LOC” y “MISC”. La evaluación de la métricas $F1-score_i$ y $F1-score_c$ se muestran en la Figura 5.2, donde el primer renglón de los gráficos ilustra el comportamiento del modelo CRF iniciando con la clase más densa a la más dispersa (*ascendente*). De forma inversa (*descendente*) ocurre en el segundo renglón de gráficos.

Las matrices obtenidas con respecto al modelo de redes neuronales recurrentes Bi-LSTM se muestran en la Figura 5.3. La clase con el mayor número de entidades reconocidas es “ORG”, y

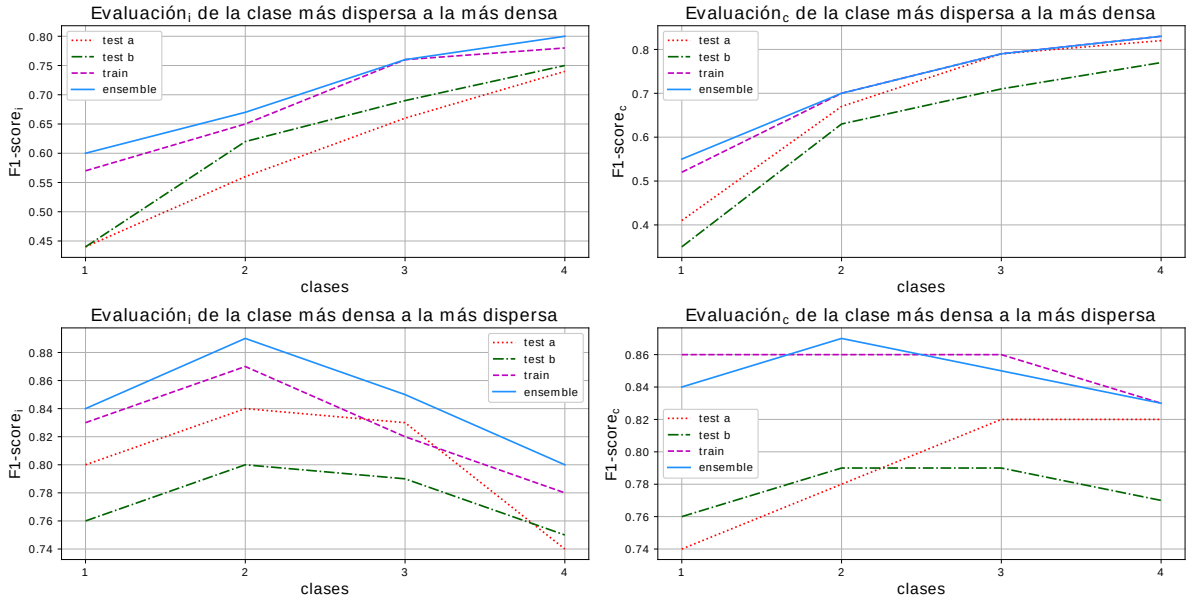

Fig. 5.2 *F1-score* del modelo CRF sobre el corpus CoNLL-2002.

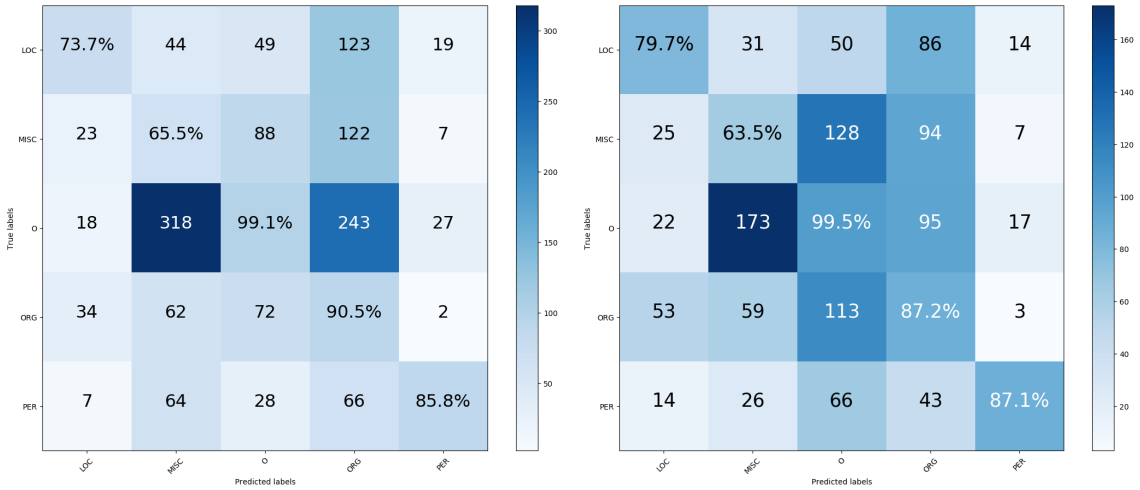

Fig. 5.3 Matrices de confusión obtenidas del modelo Bi-LSTM sobre CoNLL-2002.

la peor entidad reconocida es “MISC” sobre la evaluación *ascendente*. De manera similar ocurre con la matriz de la derecha que representa las evaluaciones *descendentes*. El comportamiento del modelo Bi-LSTM se visualiza en la Figura 5.4, donde se aprecia el comportamiento agregando una clase en cada experimento cada vez (*ascendente*) y viceversa (*descendente*).

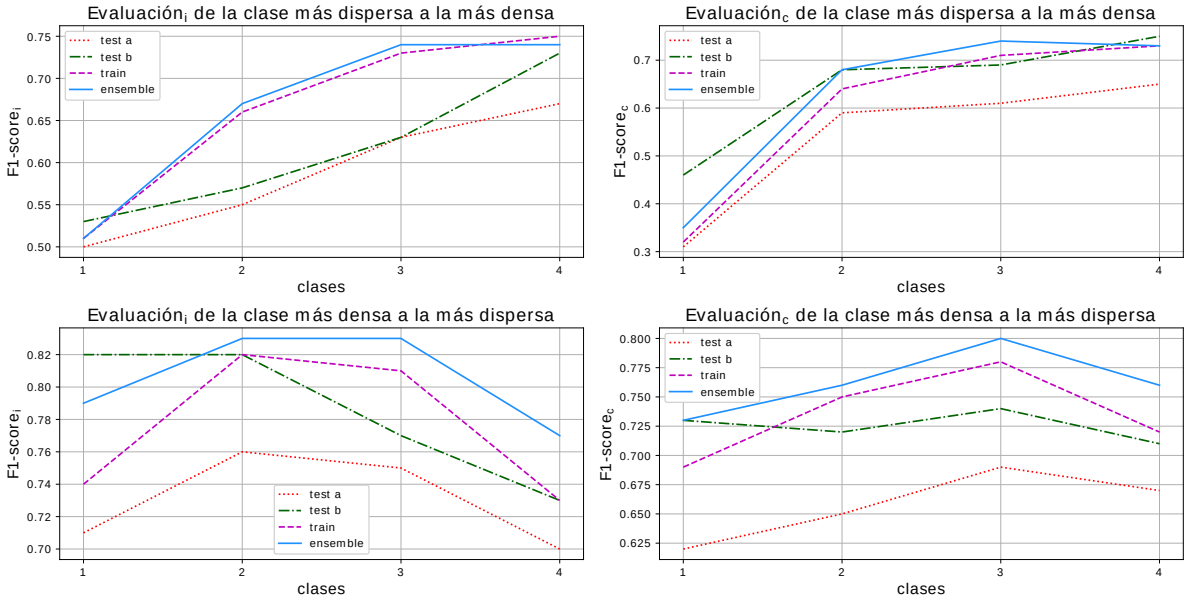

Fig. 5.4 *F1-score* del modelo Bi-LSTM sobre el corpus CoNLL-2002.

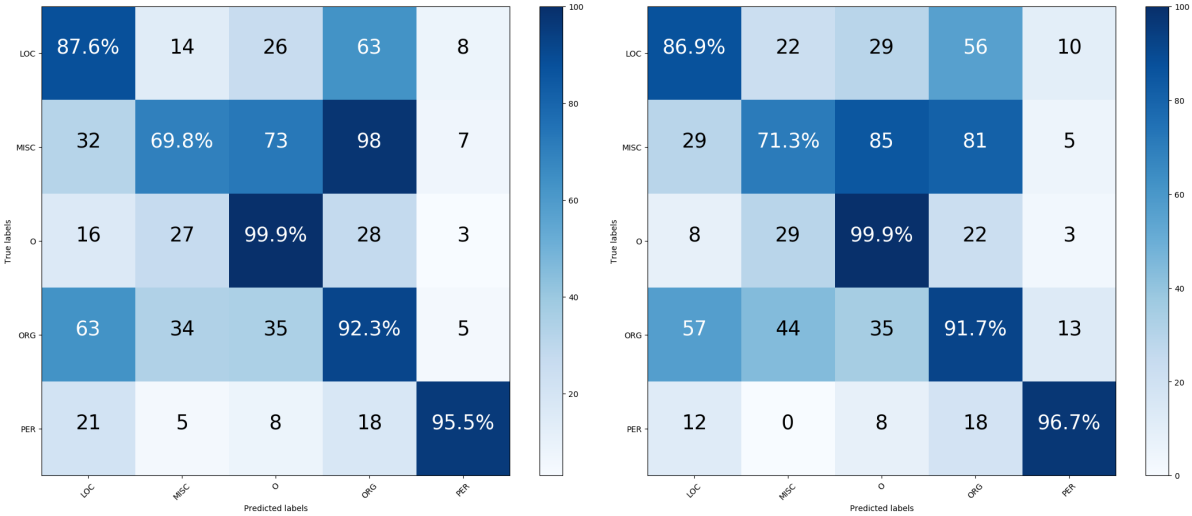

Fig. 5.5 Matrices de confusión obtenidas del modelo Bi-LSTM-ELMo sobre CoNLL-2002.

La Figura 5.5 muestra los experimentos con el modelo Bi-LSTM-ELMo utilizando el *ensemble* del corpus, donde se puede observar que las clases mejor reconocidas con este modelo son “PER” en los experimentos *ascendente* y *descendente*, alcanzando un 95.5% y 96.7% respectivamente. Las clases con los porcentajes más bajos (69.8% y 71.3%) son “MISC” para los experimentos de forma *ascendente* y *descendente*.

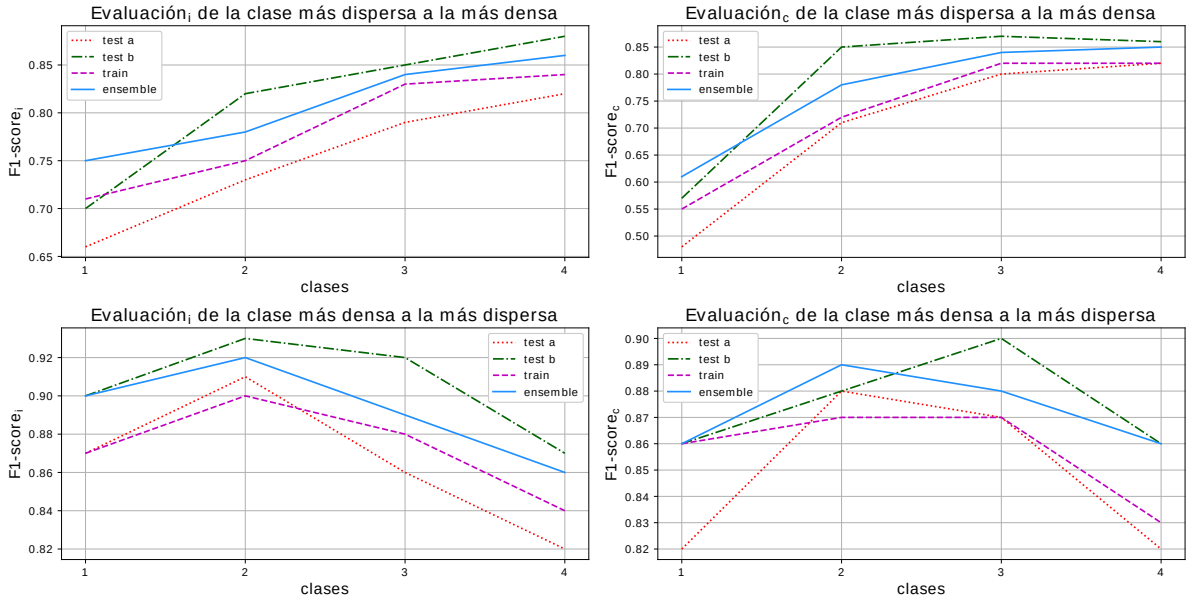

Fig. 5.6 *F1-score* del modelo Bi-LSTM-ELMo sobre el corpus CoNLL-2002.

Los resultados obtenidos sobre las métricas $F1-score_i$ y $F1-score_c$ usando el modelo Bi-LSTM-ELMo, indican que el conjunto de datos correspondiente al *Test b* obtiene de forma general mejores puntajes que *ensemble*, esto con ambas métricas y en ambos experimentos *ascendente* y *descendente*. Sin embargo al analizar la matriz de confusión del *Test b* los porcentajes para las clases "PER" son de 93.9% (*ascendente*) y 95.3% *descendente* siendo estás las más altas, las clases 'MISC' son las más bajas con 81.2% (*ascendente*) y 71.9% (*descendente*). De este modo los mejores puntajes son para *ensemble* por otro lado los resultados del *test b* los puntajes son mejores para todas las clases.

5.1.2 Resultados sobre el corpus Mx-news

Las matrices de confusión de los experimentos realizados sobre el *ensemble* (es el conjunto de todas las particiones del corpus) se ilustran en la Figura 5.7. De igual manera que en la Figura 5.1 la diagonal indica el número de las entidades reconocidas de forma correcta con el modelo CRF. La matriz de la derecha del gráfico de la Figura 5.7 muestra en porcentajes las entidades reconocidas de forma correcta. La clase reconocida por el modelo con el puntaje más alto es "PER" alcanzando un 97.1% y la clase con el porcentaje más bajo es "LOC" con 75.0%. Este resultado coincide con el etiquetado de las clases, siendo la clase "LOC" la que cuenta con menos entidades anotadas y la clase "PER" con el mayor número de entidades anotadas.

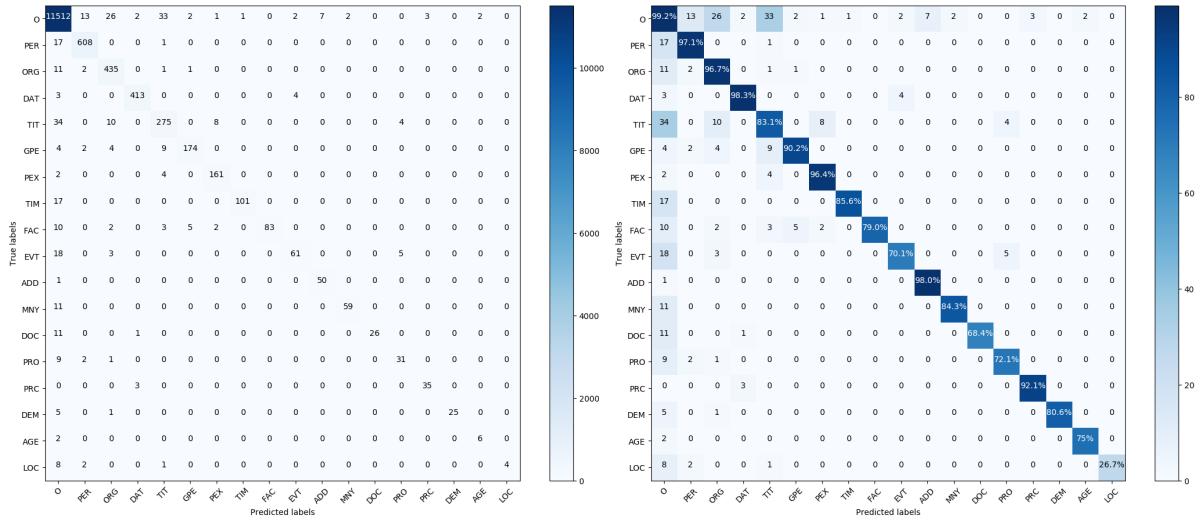

Fig. 5.7 Matrices de confusión obtenidas del modelo CRF sobre Mx-news.

La Figura 5.8 muestra los resultados obtenidos del *ensemble* para las evaluaciones $F1 - score_i$ y $F1 - score_c$, iniciando de la clase más dispersamente anotada a las más densamente anotada (*ascendente*), representado sobre el primer renglón de la Figura 5.8. Para mostrar el comportamiento de la clase más densa a la más dispersa (*descendente*) en los gráficos del segundo renglón.

Los resultados de los experimentos usando el modelo Bi-LSTM con *ensemble* se presentan en la Figura 5.9, donde la diagonal de las matrices está dado en porcentajes. La clase mejor reconocida es "PER" con 97.5% y con 50.0% la clase "LOC" siendo la peor reconocida. Como se puede observar en ambas matrices la integración de clases de forma *ascendente* y *descendente* no presentan grandes diferencias, los pequeños cambios se observan reflejados en las clases erróneamente reconocidas.

El comportamiento en los gráficos de la Figura 5.10 con las métricas $F1 - score_i$ y $F1 - score_c$ de forma *ascendente* y *descendente* muestran que el *ensemble* presenta el mejor rendimiento respecto a las otras particiones del corpus. Los experimentos *ascendentes* muestran un aumento conforme se agregan clases, por otra parte los experimentos *descendentes* inician con puntajes altos y con las siguientes clases no presentan un descenso considerable, manteniéndose por encima de las otras particiones del corpus.

Finalmente los resultados de los experimentos usando el modelo Bi-LSTM-ELMo con *ensemble* son mostrados en la Figura 5.11. La mejor clase coincide con los resultados anteriores,

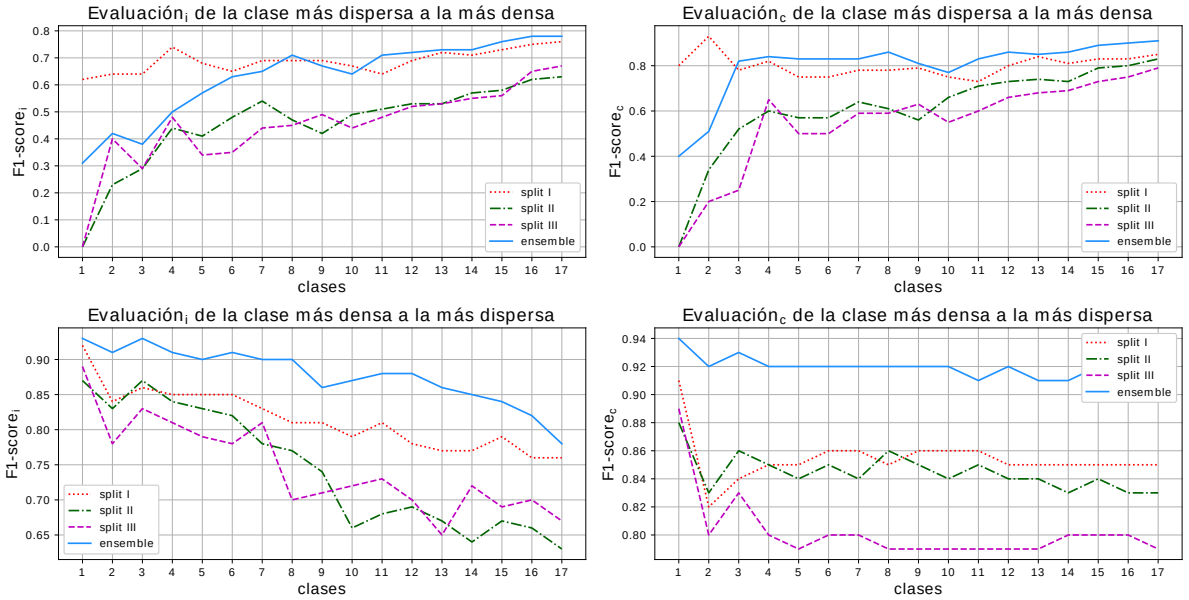

Fig. 5.8 *F1-score* del modelo CRF sobre el corpus Mx-news.

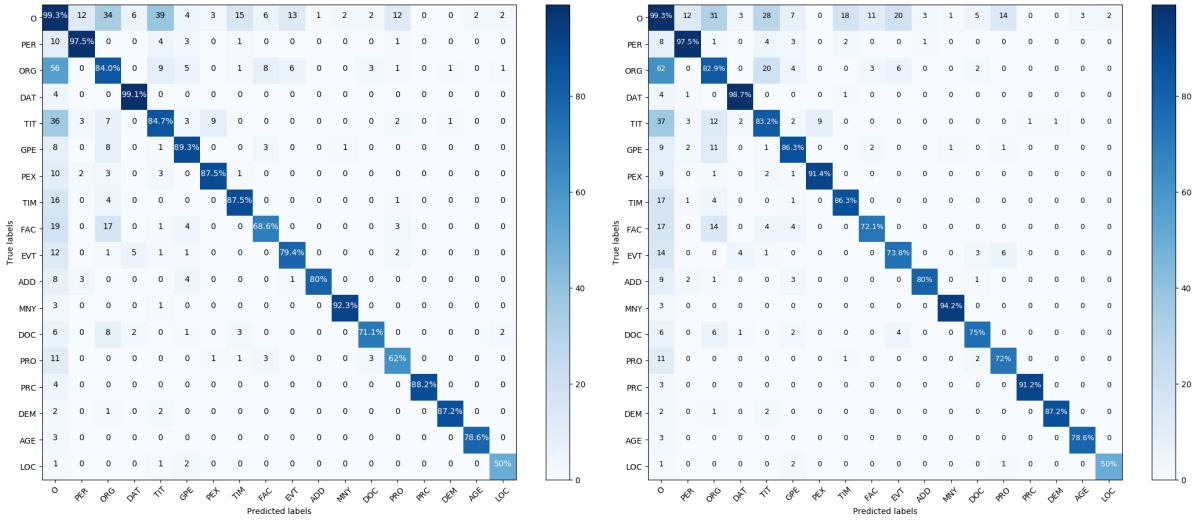

Fig. 5.9 Matrices de confusión obtenidas del modelo Bi-LSTM sobre Mx-news.

"PER" con 98.0% y 98.3% de forma *ascendente* y *descendente* respectivamente, mientras que la clase "LOC" con el peor porcentaje alcanzando en forma *ascendente* y *descendente* 76.5%.

La Figura 5.12 describe el comportamiento del modelo Bi-LSTM-ELMo, donde nuevamente el conjunto de datos *ensemble* obtiene de forma general un mejor rendimiento al resto.

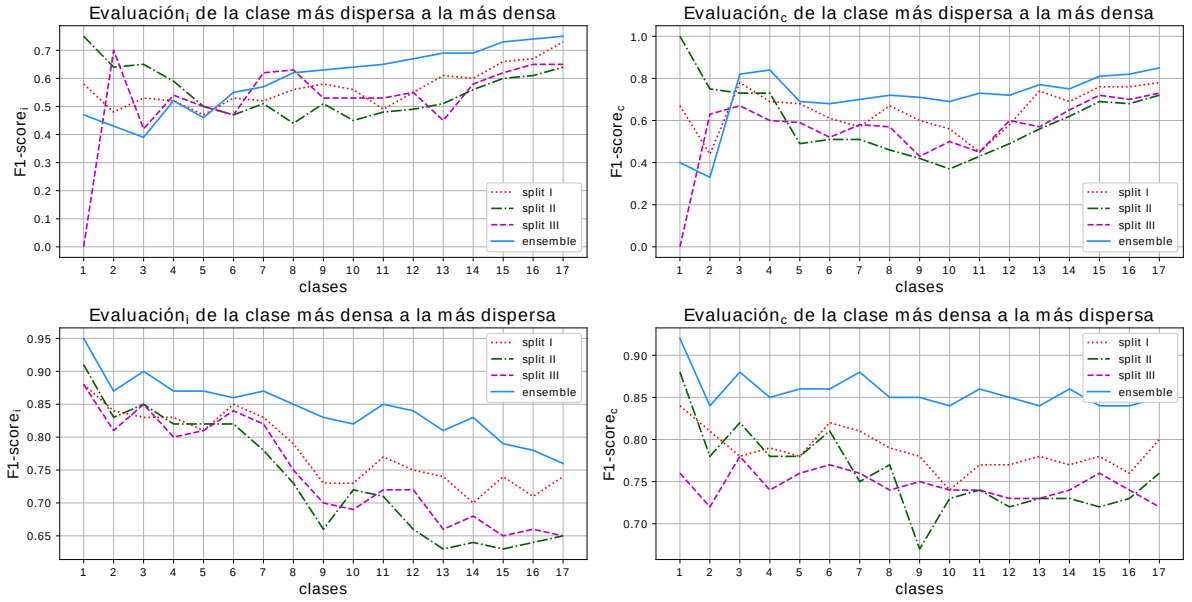

Fig. 5.10 *F1-score* del modelo Bi-LSTM sobre el corpus Mx-news.

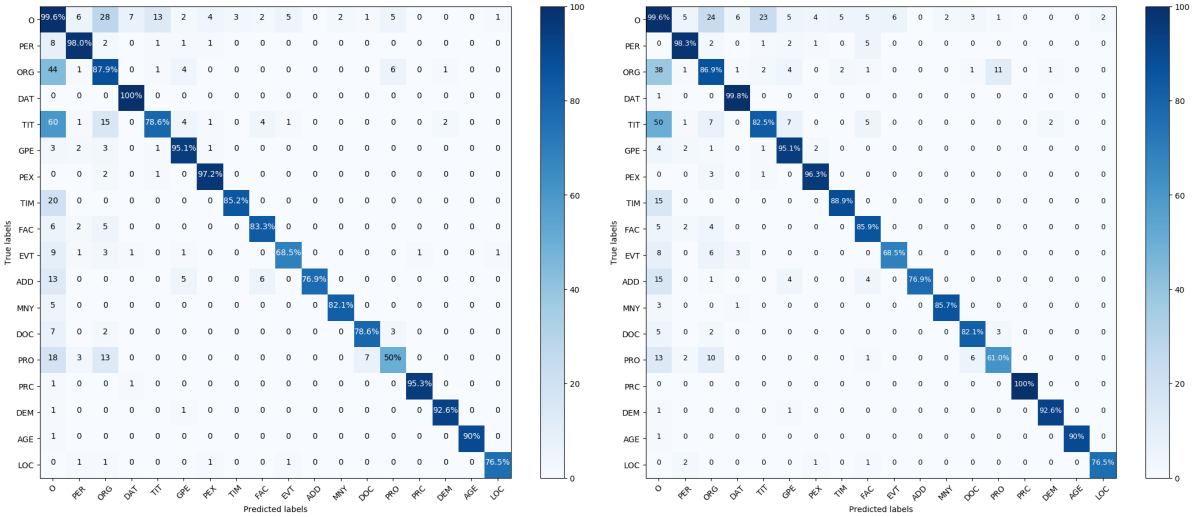

Fig. 5.11 Matrices de confusión obtenidas del modelo Bi-LSTM-ELMo sobre Mx-news.

5.1.3 Análisis sobre los modelos y corpus

Los modelos CRF, Bi-LSTM y BI-LSTM-ELMo se emplearon en los experimentos para analizar el comportamiento de los dos conjuntos de datos, debido a que presentan diferencias entre ellos como son: los esquemas de etiquetado IOB e IOBES, cuatro clases para CoNLL-2002

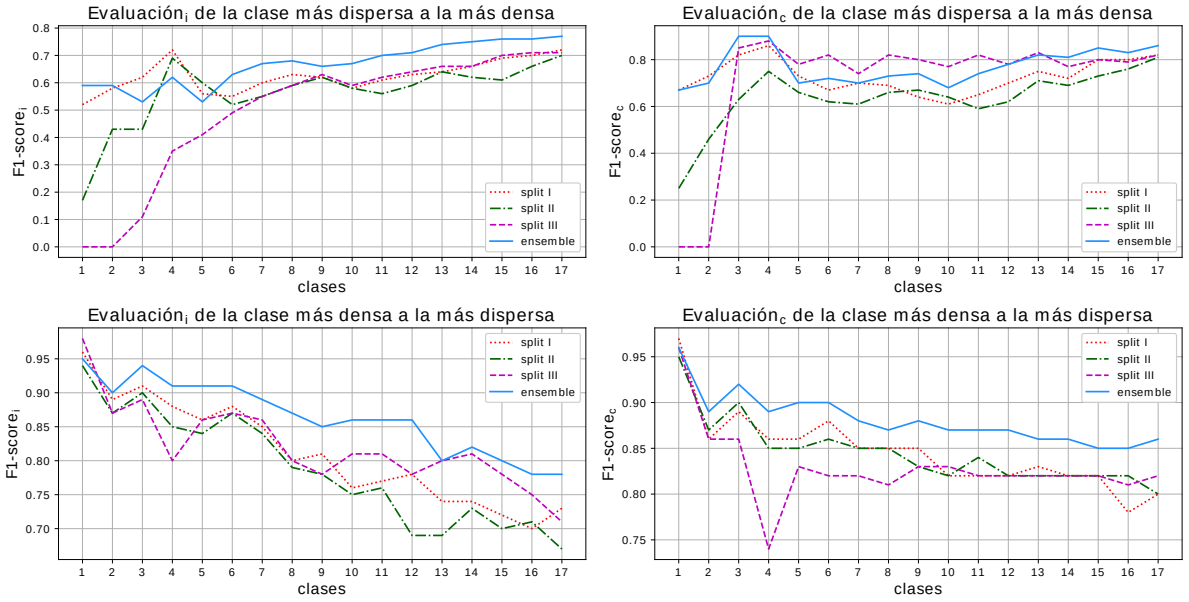

Fig. 5.12 *F1-score* del modelo Bi-LSTM-ELMo sobre el corpus Mx-news.

y diecisiete de Mx-news, y un mayor número de desbalance en las clases de Mx-news en comparación con CoNLL-2002.

Tabla 5.1 Mejores y peores clases reconocidas para CoNLL-2002.

Modelo	**Mejor**	**Asc**	**Desc**	**Peor**	**Asc**	**Desc**
CRF	ORG	86.2%	86.2%	MISC	62.0%	62.0%
Bi-LSTM	ORG	90.5%	87.2%	MISC	65.5%	63.5%
Bi-LSTM-ELMo	**PER**	**95.5%**	**96.7%**	MISC	**69.8%**	**71.3%**

Las Tablas 5.1 y 5.2 presentan las etiquetas mejor y peor reconocidas en base a su porcentaje, estos resultados son tomados de los experimentos con el conjunto de datos *ensemble* de ambos corpus. Las columnas *Asc* y *Desc* corresponden al orden en que fueron agregadas las clases dentro de los experimentos. En ambas Tablas 5.1 y 5.2 los porcentajes más altos son con el modelo Bi-LSTM-ELMo, además la mejor y peor clase coincide con la cantidad de entidades anotadas, por lo que las clases con más datos anotados presenta los mejores puntajes, y las clases con menos datos anotadas son las peores evaluadas.

Los puntajes obtenidos de los experimentos sobre el conjunto de datos *ensemble* del corpus CoNLL-2002 son presentados en la Tabla 5.3. Donde las columnas *P*, *R* y *F1* son la *precision*, *recall* y *F1-score* respectivamente. Los subíndices representan las evaluaciones sobre etiquetas

Tabla 5.2 Mejores y peores clases reconocidas para Mx-news.

Modelo	**Mejor**	**Asc**	**Desc**	**Peor**	**Asc**	**Desc**
CRF	PER	97.1%	97.1%	LOC	26.7%	26.7%
Bi-LSTM	PER	97.5%	97.5%	LOC	50.0%	50.0%
Bi-LSTM-ELMo	PER	**98.0%**	**98.3%**	LOC	**76.5%**	**76.5%**

individuales (*i*) y las evaluaciones de la entidad nombrada completa (*c*). Cada modelo es representado por dos renglones con la forma *Asc* y *Desc* para agregar clases a los experimentos. En la Tabla 5.3 se observa que los mejores puntajes obtenidos son con el modelo Bi-LSTM-ELMo, donde el porcentaje más alto alcanzado es con la métrica de $precision_i$ con un 88%, y el $precision_c$ con 71% siendo el puntaje más bajo del modelo Bi-LSTM.

Tabla 5.3 Puntajes obtenidos de los modelos empleados con el corpus CoNLL-2002.

Modelos	**Orden**	$\mathbf{P}_i$	$\mathbf{R}_i$	$\mathbf{F1}_i$	$\mathbf{P}_c$	$\mathbf{R}_c$	$\mathbf{F1}_c$
CRF	Asc	0.82	0.79	0.80	**0.85**	0.82	0.83
	Desc	0.82	0.79	0.80	**0.85**	0.82	0.83
Bi-LSTM	Asc	0.74	0.76	0.74	0.71	0.77	0.73
	Desc	0.79	0.76	0.77	0.75	0.78	0.76
Bi-LSTM-ELMo	Asc	**0.88**	**0.84**	**0.86**	0.84	0.86	0.85
	Desc	**0.88**	**0.84**	**0.86**	**0.85**	**0.87**	**0.86**

La Tabla 5.4 donde se evaluó al corpus Mx-news presenta al modelo CRF con un mejor rendimiento que los modelos basados en redes neuronales. Donde sus mejores puntajes se obtuvieron el la métrica $precision_c$ con 93% y el peor porcentaje de 74% fue con la métrica $recall_i$ del modelo Bi-LSTM.

Tabla 5.4 Puntajes obtenidos de los modelos empleados con el corpus Mx-news.

Modelos	**Orden**	$\mathbf{P}_i$	$\mathbf{R}_i$	$\mathbf{F1}_i$	$\mathbf{P}_c$	$\mathbf{R}_c$	$\mathbf{F1}_c$
CRF	Asc	**0.84**	0.76	**0.78**	**0.93**	**0.90**	**0.91**
	Desc	**0.84**	0.76	**0.78**	**0.93**	**0.90**	**0.91**
Bi-LSTM	Asc	0.78	0.74	0.75	0.83	0.87	0.85
	Desc	0.79	0.74	0.76	0.84	0.87	0.85
Bi-LSTM-ELMo	Asc	0.80	0.75	0.77	0.83	0.88	0.86
	Desc	0.80	**0.77**	**0.78**	0.84	0.89	0.86

5.2 Extracción automática de relaciones

Experimentos para la extracción automática de relaciones sobre documentos no estructurados de noticias se describen en esta sección.

5.2.1 Conjunto de datos

El conjunto de datos utilizado consta de 32,147 documentos de noticias políticas, que previamente se ha etiquetado con el modelo NER presentado en la sección anterior. Se etiquetaron 17 clases de entidades nombradas como ilustra en la Tabla 5.5 y de estas solo se usaron 15 clases, se omitieron las entidades nombradas de porcentaje (PRC) y dirección (ADD). De los 32,147 documentos se emplearon 175,754 oraciones para la identificación y extracción de relaciones. En la Tabla 5.5 la columna *Frec.* enumera el total de entidades nombradas por clase, mientras que la columna *Única* describe el número de entidades únicas.

Tabla 5.5 Entidades reconocidas en el corpus de noticias.

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

La Tabla 5.6 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 M: Método usado para extraer la relación; 1:

Puestos de Trabajo, 2: Verbo como ancestro de E1. 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 5.6 Top 10 de relaciones identificadas y extraídas.

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

De la Tabla 5.6 se observa que FO es igual a 1 en los renglones del 4,5 y del 7 al 10. Esto se debe a que la oración suele aparecer al final del documento; ocurre en documentos donde la noticia principal se extrajo junto con los “pies de página” donde se detallan pequeñas síntesis de otras noticias o encabezados que ligan a una noticia diferente.

5.2.2 Extracción de relaciones

Para la extracción de relaciones se definieron nueve métodos (véase Sección 4.2.2) para la identificación y extracción de las relaciones. Se seleccionó una cantidad específica de relaciones extraídas para su evaluación de forma manual. Para ello se realizaron dos experimentos. En el primer experimento se usó el corpus completo y en el segundo un corpus de 300 documentos de forma aleatoria.

La evaluación manual consiste en ordenar las tripletas en base a la frecuencia y al método usado para su extracción, esto de forma descendente. La evaluación se realizó usando un sistema web desplegando una tripleta a la vez, así como visualizando a lo mas diez oraciones en las que ocurre la tripleta. Lo que permite al evaluador calificar como correcta o incorrecta cada una de las partes que componen la tripleta (<*entidad1*, *relación*, *entidad2*>). Las oraciones desplegadas sirven como referencia al evaluador para corroborar que la relación fue identificada y definida de forma correcta.

Experimento 1

Se utilizaron los 32,147 documentos para extraer relaciones siguiendo los métodos propuestos. Las relaciones obtenidas (tripletas <*entidad1, relación, entidad2*>) en este experimento son **86,917**. Sin embargo, para la evaluación manual solo se tomaron 200 tripletas de cada método, exceptuando al método *REPR* que contiene 33 tripletas extraídas.

La Figura 5.13 muestra los porcentajes alcanzados al evaluar las tripletas extraídas por cada método. Además por cada método se presentan dos barras, la primer barra representa el porcentaje de evaluación de las relaciones, esto sin tomar en cuenta la evaluación de las dos entidades nombradas que acompañan la relación. La segunda barra de cada método representa la evaluación de la tripleta, donde para ser evaluada correctamente la *entidad1*, la *relación* y la *entidad2* deben ser correctas.

Una descripción detallada de la evaluación se describe en la Tabla 5.8 para cada uno de los métodos usados para extraer relaciones y formar tripletas. La columna *Total* enumera la cantidad neta de relaciones extraídas por método. Como se ha mencionado previamente para la evaluación (columna *Eval* Evaluadas) 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.

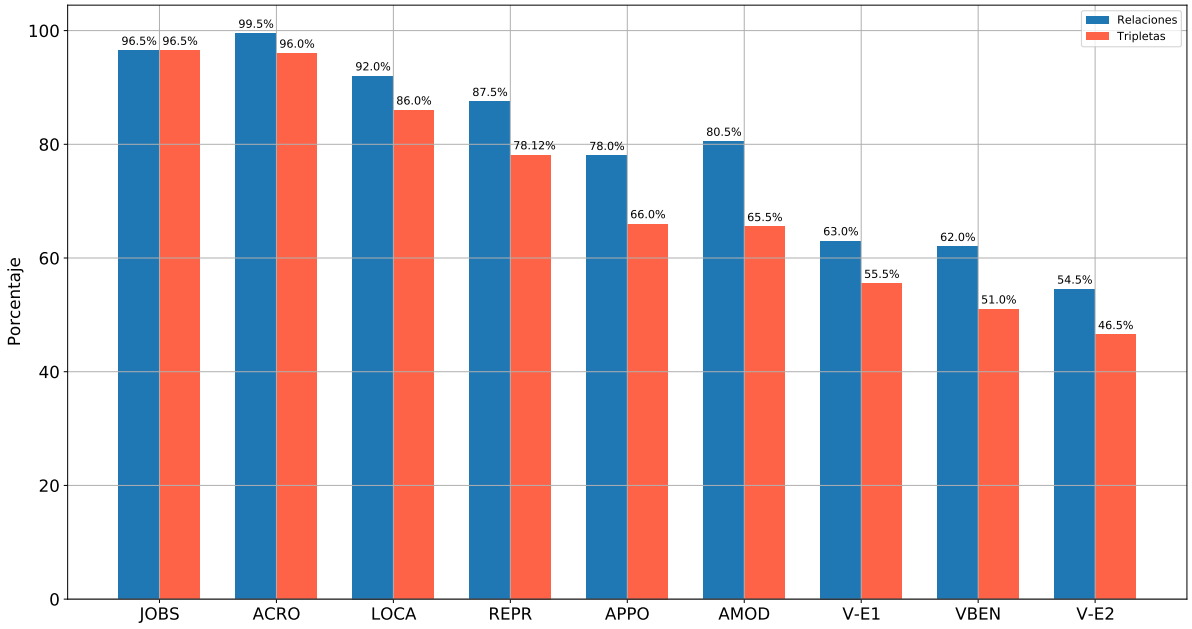

Fig. 5.13 Evaluación de métodos empleados para el experimento 1.

Tabla 5.8 Evaluación de métodos para el experimento 1.

No.	Código	Método	Total	Eval	CR	CT	PR	PT
1	JOBS	Puestos de trabajo	17,073	200	193	193	96.5%	**96.5%**
2	ACRO	Acrónimo	2,367	200	199	192	**99.5%**	96.0%
3	LOCA	Localizado	343	200	184	172	92.0%	86.0%
4	REPR	Representado por	33	32	28	25	87.5%	78.12%
5	APPO	Dependencia appos	1,408	200	156	132	78.0%	66.0%
6	AMOD	Dependencia amod	1,922	200	161	131	80.5%	65.5%
7	V-E1	Verbo ancestro E1	44,479	200	126	111	63.0%	55.5%
8	VBEN	Verbo entre entidades	7,896	200	124	102	62.0%	51.0%
9	V-E2	Verbo ancestro E2	11,396	200	109	93	54.5%	46.5%
			86,917	**1,632**	**1,280**	**1,151**	**78.43%**	**70.52%**

La frecuencia de las entidades nombradas evaluadas correctamente del Experimento 1 se listan en la Tabla 5.9. Las entidades con más frecuencia son ORG (*Organización*), PER (*Persona*), TIT (*Título*), GPE (*Geopolítica*) y PEX (*Partido político*) en ambas evaluaciones (relaciones y tripletas). La entidad LOC (*Lugar*) no aparece involucrada en ninguna de las evaluaciones.

Tabla 5.9 Frecuencia de entidades nombradas evaluadas en el Experimento 1.

No.	**Evaluación de Relaciones**		**Evaluación de Tripletas**	
	Frecuencia	**Entidad**	**Frecuencia**	**Entidad**
1	768	Organización	705	Organización
2	616	Persona	533	Persona
3	384	Título	363	Título
4	335	Geopolítica	288	Geopolítica
5	132	Partido político	127	Partido político
6	83	Fecha	79	Fecha
7	56	Instalación	53	Evento
8	56	Evento	44	Instalación
9	29	Gentilicio	26	Documento
10	28	Documento	24	Gentilicio
11	27	Moneda	23	Moneda
12	25	Producto	21	Producto
13	12	Tiempo	10	Tiempo
14	9	Edad	6	Edad
15	0	Lugar	0	Lugar

Las Tabla 5.10 lista el top 20 de relaciones obtenidas con más frecuencia. Del top 3 la relación *"tiene el acrónimo de"* forma una tripleta empleando en su mayoría dos entidades ORG, la relación *"es el título de"* forma una tripleta con las entidades TIT y PER y la relación *"que pertenece a"* consta de las entidades ORG y GPE para formar una tripleta. Por esta razón se observa que las entidades (Tabla 5.9) y relaciones (Tabla 5.10) con mayor frecuencia poseen una mayor correlación debido a la estructura para formar tripletas. Por otro lado, las relaciones dentro del top 6 se encuentran en los métodos JOBS, ACRO y REPR, en estos métodos se definieron de forma manual las relaciones en base al patrón de las entidades dentro de cada tripleta. El total de relaciones únicas es de 625

Tabla 5.10 Top 20 de frecuencias sobre relaciones obtenidas en el Experimento 1.

#	Evaluación de Relaciones		Evaluación de Tripletas	
	Frc	**Relación obtenida**	**Frc**	**Relación obtenida**
1	199	tiene el acrónimo de	192	tiene el acrónimo de
2	163	que pertenece a	155	es el título de
3	155	es el título de	153	que pertenece a
4	28	es representado por	25	es representado por
5	23	es un puesto de trabajo en	23	es un puesto de trabajo en
6	21	se localiza en	19	se localiza en
7	10	de la coalición	10	de la coalición
8	8	tiene el título de	8	tiene el título de
9	6	es donde desempeña sus funciones	6	es donde desempeña sus funciones
10	5	encabezado por	5	encabezado por
11	4	propuesto como	4	propuesto como
12	4	titular de la	4	titular de la
13	3	conocido como	3	encabezada por el
14	3	encabezada por el	3	por la coalición
15	3	por la coalición	3	futuro
16	3	futuro	3	nominado a la
17	3	para el ejercicio	3	del partido
18	3	nominado a la	2	manifestó En el
19	3	del partido	2	conocido como
20	2	manifestó En el	2	que encabezará

De la evaluaciones sobre relaciones sin tomar en cuenta la evaluación de las entidades se obtuvieron **625** relaciones únicas evaluadas de forma correcta. En cambio se obtuvieron **522** tripletas únicas evaluadas de forma correcta.

Experimento 2

Para este experimento se seleccionaron aleatoriamente 300 documentos del corpus principal. La Figura 5.14 describe los porcentajes de los métodos evaluados donde se ilustran dos columnas por método. Los métodos están ordenados en base al porcentaje alcanzado por la columna de las tripletas esto en forma descendente. En este experimento el método *REPR* no presentó ninguna

relación extraída, por lo que no es presentado en los resultados. Por otro lado, se observa el método *LOCA* (ambas columnas) y el método *AMOD*) (primer columna) presentan el 100%, sin embargo las relaciones y tripletas evaluadas no son significativas, como se presenta en la Tabla 5.11.

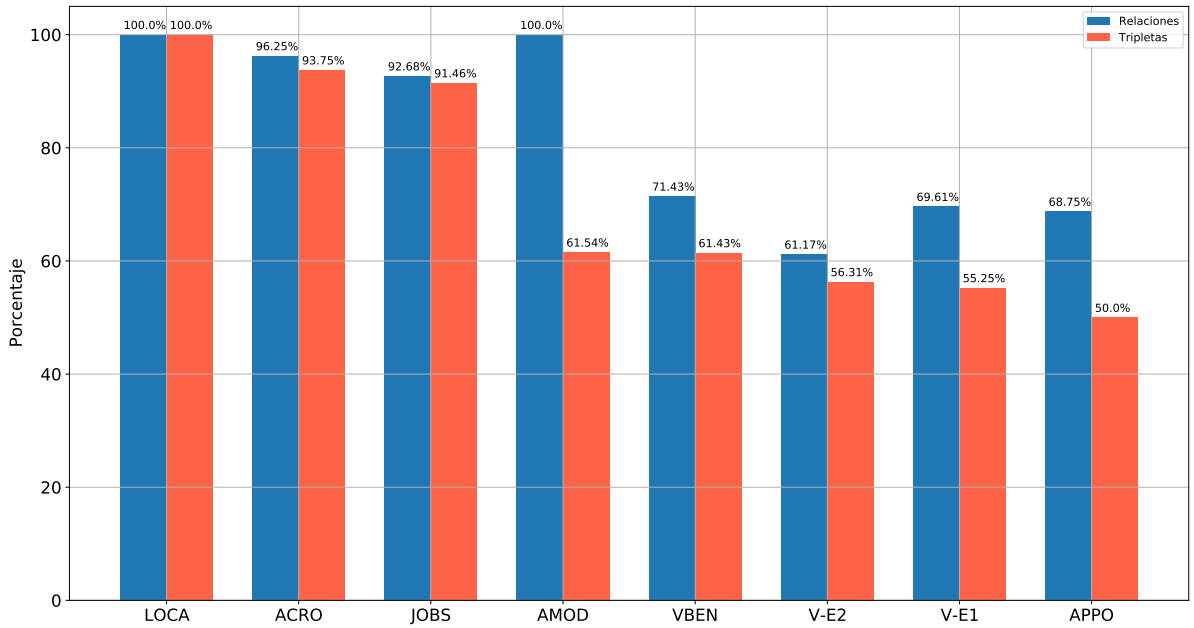

Fig. 5.14 Evaluación de métodos empleados para el Experimento 2.

Tabla 5.11 Evaluación de métodos para el Experimento 2.

No.	Código	Método	Total	Eval	CR	CT	PR	PT
1	LOCA	Localizado	4	4	4	4	**100.0%**	**100.0%**
2	ACRO	Acrónimo	80	80	77	75	96.25%	93.75%
3	JOBS	Puestos de trabajo	334	328	304	300	92.68%	91.46%
4	AMOD	Dependencia amod	16	13	13	8	**100.0%**	61.54%
5	VBEN	Verbo entre entidades	76	70	50	43	71.43%	61.43%
6	V-E2	Verbo ancestro E2	115	103	63	58	61.17%	56.31%
7	V-E1	Verbo ancestro E1	435	362	252	200	69.61%	55.25%
8	APPO	Dependencia appos	18	16	11	8	68.75%	50.0%
			1,078	**976**	**774**	**696**	**79.30%**	**71.31%**

Los métodos con el mayor número de tripletas evaluadas son *V-E1*, *JOBS* y *V-E2* de la Tabla 5.11 con un *55.25%*, *91.46%* y *56.31%* respectivamente. Por otro lado el método *LOCA* con muy pocas tripletas evaluadas presenta un *100%*. Como se ha mencionado en el Experimento 1, las columnas de la Tabla 5.11 *Total* representa el número total de tripletas, *Eval* (Evaluadas) enumera las tripletas que son evaluadas (algunas tripletas se descartaron porque las oraciones donde están involucradas no aportan información necesaria para definir la relación), *CR* (Conteo Relaciones) lista las relaciones evaluadas correctamente sin tomar en cuenta la validación de las entidades nombradas, *CT* (Conteo Tripletas) enumera la cantidad de tripletas evaluadas correctamente, *PR* (Porcentaje Relaciones) y *PT* (Porcentaje Tripletas) representan el porcentaje alcanzado respectivamente. Adicionalmente la última fila muestra los totales de cada columna.

Tabla 5.12 Frecuencia de entidades en las Evaluaciones.

No.	**Evaluación de Relaciones**		**Evaluación de Tripletas**	
	Frecuencia	**Entidad**	**Frecuencia**	**Entidad**
1	494	Persona	436	Persona
2	363	Título	347	Título
3	343	Organización	304	Organización
4	101	Partido político	96	Partido político
5	81	Fecha	73	Fecha
6	79	Geopolítica	63	Geopolítica
7	20	Instalación	15	Instalación
8	16	Evento	15	Documento
9	15	Documento	15	Evento
10	11	Moneda	9	Moneda
11	9	Tiempo	7	Tiempo
12	6	Producto	5	Gentilicio
13	6	Gentilicio	4	Edad
14	4	Edad	3	Producto
15	0	Lugar	0	Lugar

La frecuencia de entidades nombradas evaluadas de forma correcta en las evaluaciones son presentada en la Tabla 5.12. Donde las entidades PER (*Persona*), TIT (*Título*), ORG (*Organización*) y PEX (*Partido político*) son las de mayor frecuencia. A diferencia de la entidad LOC (*Lugar*) que no aparece en las evaluaciones.

Tabla 5.13 Top 20 de frecuencias sobre relaciones obtenidas en el Experimento 2.

#	**Evaluación de Relaciones**		**Evaluación de Tripletas**	
	Frc	**Relación obtenida**	**Frc**	**Relación obtenida**
1	223	es el título de	221	es el título de
2	77	tiene el acrónimo de	75	tiene el acrónimo de
3	29	es un puesto de trabajo en	29	es un puesto de trabajo en
4	25	tiene el título de	25	tiene el título de
5	19	desempeña sus funciones en	18	desempeña sus funciones en
6	7	es donde desempeña sus funciones	6	es donde desempeña sus funciones
7	4	que pertenece a	4	que pertenece a
8	2	explicó En	2	explicó En
9	2	continuará en Sonora su gira tras los resultados este	2	continuará en Sonora su gira tras los resultados este
10	1	procedente de	1	procedente de
11	1	referente a la creación de la	1	referente a la creación de la
12	1	de la sociedad cooperativa	1	dueño de
13	1	dueño de	1	que buscará tranquilizar a los inversionistas encabezada
14	1	en el que también militará su esposo el	1	que el próximo cumplirá un mes
15	1	celebrada en el	1	que sí alcanzará el presupuesto de
16	1	que buscará tranquilizar a los inversionistas encabezada	1	periodo durará
17	1	que el próximo cumplirá un mes	1	Por parte mediante su cuenta escribió de
18	1	que sí alcanzará el presupuesto de	1	refirió que en la conversación este le manifestó el mensaje al
19	1	periodo durará	1	la bancada presentó una iniciativa El pasado
20	1	Por parte mediante su cuenta escribió de	1	En los avances explicó del

Las relaciones en el top 5 de la Tabla 5.13 fueron definidas de forma manual en los métodos JOBS y ACRO. De las relaciones obtenidas, *"tiene el acrónimo de"* conforma una tripleta en su mayoría con dos entidades de la clase ORG, las relaciones *"es el título de"*, *"es un puesto de trabajo en"*, *"tiene el título de"* y *"desempeña sus funciones en"* pertenecen al método JOBS. De igual forma que en el Experimento 1, las entidades (Tabla 5.12) y relaciones (Tabla 5.13) poseen una mayor correlación cuando su frecuencia también es mayor, esto a causa de la forma en que se definió la estructura de las tripletas.

El total de relaciones únicas evaluadas de forma correcta no tomando en cuenta la evaluación de las entidades corresponde a **395**. En comparación **323** tripletas únicas evaluadas correctamente fueron obtenidas.

5.3 Base de hechos y Reglas lógicas

En esta sección se describe la base de hechos obtenida a partir de las tripletas generadas en la extracción de relaciones. Se presentan las base de hechos obtenida del Experimento 1 (Sección 5.2.2). En este experimento se evaluaron las relaciones obtenidas (sin tomar en cuenta la evaluación de las entidades) y las tripletas (se contemplaron las entidades y relación en la evaluación). Para la construcción de la base de hechos se utilizaron las tripletas.

5.3.1 Base de hechos

Del Experimento 1 (Sección 5.2.2) con **1,110** tripletas evaluadas correctamente se realizó el proceso de transformación de acuerdo a los lineamientos de Prolog, que consiste en convertir a minúsculas todas las entidades nombradas y eliminar sus símbolos no permitidos. Las tripletas con la estructura <*entidad1, relación, entidad2*> se transformaron a *triple(clase_entidad1(texto_entidad1), relacion(texto_relacion)), clase_entidad2(texto_entidad2).* como se describe en la Sección 4.3.2. Algunos ejemplos que constituyen la base de hechos generada se describen a continuación:

- triple(*moneda*("5 billones 600 mil pesos"), *relacion*("aprobados para el"), *fecha*("2019")).
- triple(*documento*("ley general"), *relacion*("aprobada en", *fecha*("mayo"))).
- triple(*organizacion*("congreso"), *relacion*("que pertenece a"), *geopolitica*("cdmx")).
- triple(*persona*("amlo"), *relacion*("se reune con"), *persona*("meade")).

5.3.2 Reglas lógicas definidas

Se han definido algunas reglas lógicas para *descubrir conocimiento* que a simple vista no podría observarse en los datos. La Figura 5.15 describe de forma gráfica cuatro de estas reglas.

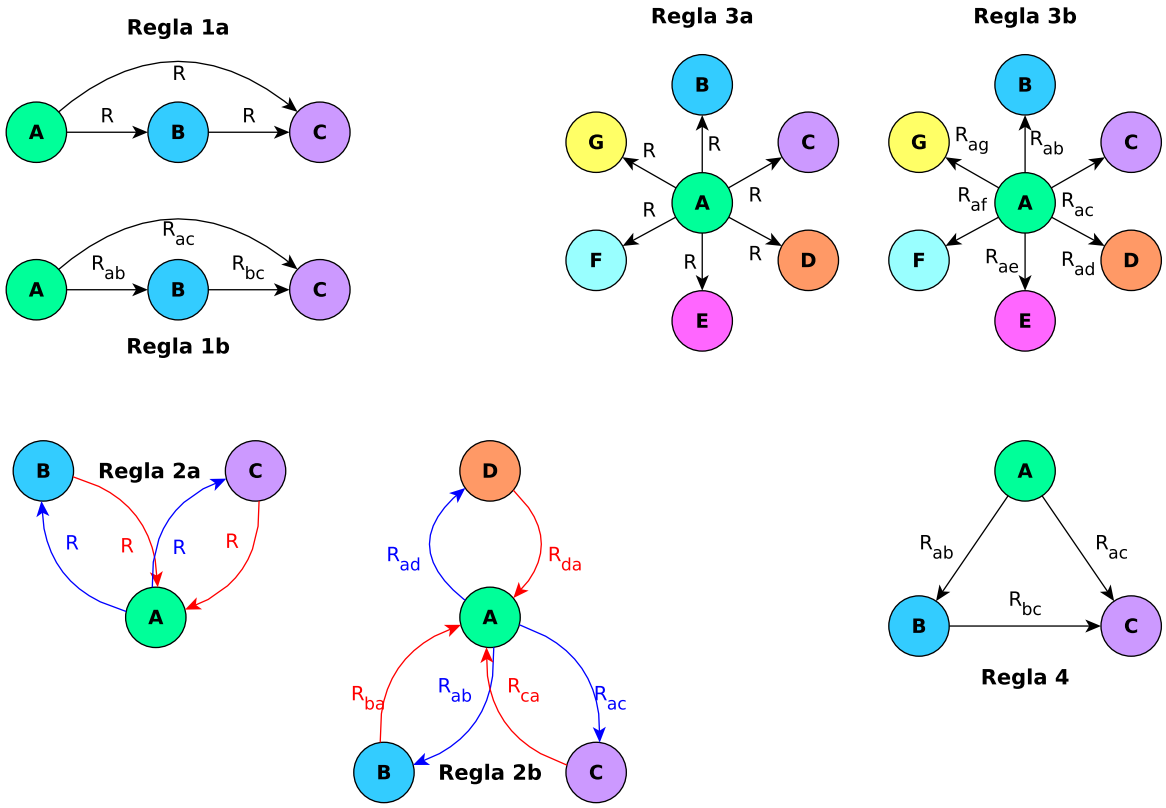

Fig. 5.15 Reglas básicas propuestas.

- **Regla 1**. Al definir la regla se pretende observar en primera instancia la relación que existe de $A \rightarrow B$ y de $B \rightarrow C$. Asimismo se desea observar si existe una relación directa de $A \rightarrow C$ sin pasar por B. La regla 1 consiste de dos definiciones (véase la Figura 5.15) como se describe a continuación. La primera regla nombrada como **regla1a** hace uso de una única relación R para vincular a dos entidades. Las entidades representadas con letras mayúsculas (variables) tienen que ser diferentes entre sí, se define con el símbolo (\=) entre las variables involucradas. Las comas (,) representan la conjunción, la palabra *nl* al final de la regla inserta una *nueva línea* después de cada resultado obtenido, y el punto (.) se usa para indicar el fin de una regla.

En cambio la **regla1b** define una relación diferente para cada una de las entidades, además las entidades y relaciones deben de ser diferentes entre sí (\=) como se observa tanto en la Figura 5.15.

- **Regla 2**. Se define para observar si existe una relación que parta de una entidad nombrada hacia otra, y a su vez contenga una relación de regreso a la entidad inicial. Al igual que la anterior se establecieron dos reglas. La **regla2a** donde se pretende conocer si existe una relación $A \rightarrow B$ y una relación de regreso ($A \leftarrow B$). De igual forma para $A \rightarrow C$ y de $A \leftarrow C$. En la regla se establece que la relación R debe ser la misma para todas las posibles relaciones.

  A su vez la **regla2b** se define de la misma forma que la regla **regla2a** con algunas excepciones. Las relaciones en la **regla2b** deben ser todas diferentes y se adiciona una entidad más (D). La definición de estas reglas tiene el objetivo de observar los hechos obtenidos a partir del cumplimiento por medio de la inferencia de las condiciones dadas.

- **Regla 3**. La intención de la regla es obtener hechos que tengan una relación con una entidad en particular. La **regla3a** tiene como foco una entidad A y debe contener exactamente la misma relación hacia otras entidades (B, C, D, E, F y G).

  Su contra parte la **regla3b** establece que las relaciones de A hacia B, C, D, E, F y G deben ser todas diferentes.

- **Regla 4**. La regla aquí definida deben contener entidades que se encuentran ligadas a la entidad de partida A de la siguiente forma: $A \rightarrow B$ y $A \rightarrow C$. La intención de está regla es observar los hechos resultantes de $B \rightarrow C$, y determinar si se relaciona la información obtenida con la entidad A. Esta regla se debe cumplir para diferentes entidades R entre cada para de entidades.

Un conjunto de reglas adicionales se han definido en la Figura 5.16 que ilustra cuatro reglas más, dentro de las reglas establecidas todas las relaciones como restricción tienen que ser diferentes. La descripción de cada una de las reglas se lista a continuación.

- **Regla 5**. Siendo muy similar a la *Regla 4* se define un nivel más de relaciones y entidades, además las relaciones R tienen que ser diferentes en cada hecho (tripleta). El objetivo es observar los hechos obtenidos entre $D \rightarrow E$ y observar si se relaciona este nuevo nivel con las entidades ancestro.

- **Regla 6**. La regla definida tiene un punto de partida (entidad *A*) con relaciones *R* diferentes para cada uno de los hechos. Nuevamente, se desea observar el conocimiento que puede aportar los hechos en los niveles más bajos (color rojo), así como conocer el comportamiento de los hechos restantes hacia el punto de partida.

- **Regla 7**. La definición de la regla contiene a su vez dos reglas *transitivas* (por su definición es equivalente a la *Regla 1b*). En la Figura 5.16 las entidades *E* y *F* así como las relaciones que las involucran, no se visualizaran en los resultados, sin embargo, deben cumplirse para llegar a un resultado.

- **Regla 8**. La regla se encuentra constituida de tres reglas *transitivas* que comparten la entidad *A* además de ser el origen de la regla. En esta regla Prolog desplegara toda la información.

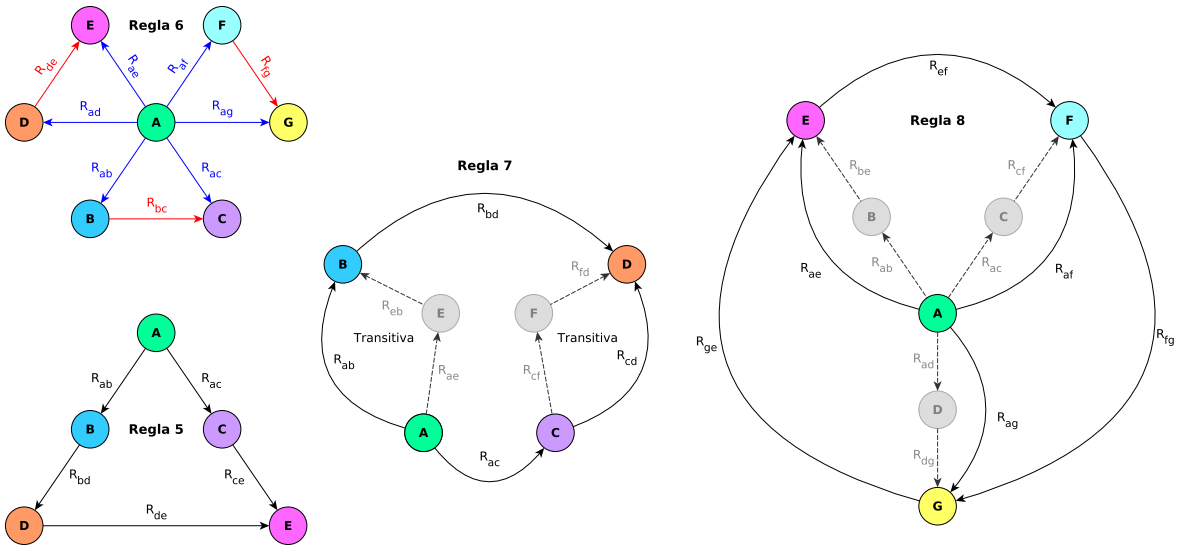

Fig. 5.16 Conjunto de reglas definidas adicionalmente.

5.3.3 Resultados de los experimentos

En los experimentos se emplearon las bases de hechos del Experimento 1 (Base1: 1,151 hechos) y el Experimento 2 (Base2: 696 hechos) ambos en la Sección 5.2.2, así como la base de hechos completa (Base3: 86,871 hechos). Algunas de las reglas propuestas no se satisfacen por la poca cantidad de hechos contenidos en sus respectivas bases de hechos. La Tabla 5.15 describe cada

una de las reglas establecidas y en cuales conjuntos de datos se obtuvieron hechos. Como se observa en la Tabla 5.15 la *Regla 1a* y la *Regla 2a* solo se satisfacen con los hechos establecidos en la Base3. Así el símbolo (✓) indica que la regla fue satisfecha, en caso contrario se usa el símbolo (-) para denotar esto.

Tabla 5.15 Estado de las reglas establecidas para obtener hechos.

#	Reglas	Base1	Base2	Base3
1	1a	-	-	✓
2	1b	✓	✓	✓
3	2a	-	-	✓
4	2b	✓	-	✓
5	3a	✓	✓	✓
6	3b	✓	✓	✓
7	4	✓	✓	✓
8	5	✓	✓	✓
9	6	✓	-	✓
10	7	✓	✓	✓
11	8	✓	-	✓

Regla 1a

Los resultados obtenidos de la *Regla 1a* se ilustran de forma gráfica en la Figura 5.17, donde se observa un resultado obtenido. La ejecución de la regla fue realizada sobre la Base3 y se describe a continuación:

```
?- regla1a(A,B,C,R).
A = organizacion("FGR"),
B = organizacion("Fiscalía General de la República"),
C = organizacion("PGR"),
R = relacion("tiene el acrónimo de")
```

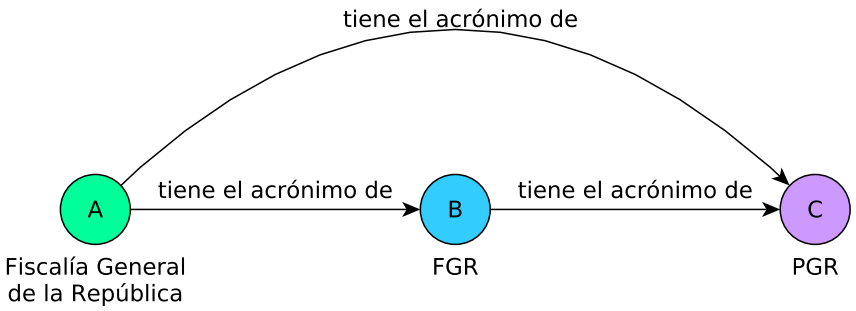

Fig. 5.17 Resultado de aplicar la regla 1a.

La Tabla 5.16 detalla la procedencia de cada una de las tripletas involucradas en la obtención de hechos de la *Regla 1a*. Los resultados de acuerdo a la especificación de la *Regla 1a* deben tener la misma relación, en este caso es $R =$ *"tiene el acrónimo de"*.

Tabla 5.16 Origen de hechos obtenidos de la Regla 1a.

#	IdD	IdO	Oración
1	1112805	77345	Dos de los más fuertes prospectos a encabezar la *Fiscalía General de la República ORG* (*FGR ORG*) , Bernardo Bátiz y Alejandro Gertz Manero , comparecerán este martes ante la Comisión de Justicia del
2	1495831	111869	La investigación inició desde hace casi tres años , la *FGR ORG* (antes *PGR ORG*) solicitará nuevas órdenes de aprehensión a jueces federales en las próximas semanas
3	1380612	101505	Asimismo , Ángel Ávila , de la dirigencia nacional del PRD , también pidió a la *Fiscalía General de la República ORG* (antes *PGR ORG*) ampliar la investigación sobre la relación que existe entre el senador de Morena y el principal operador financiero del ex Gobernador Javier Duarte .

De la Tabla 5.16 se tiene la columna **idD** (id del documento) e **idO** (id de la oración), así como el texto de la oración empleada. Como se observa se identificó la relación *"tiene el acrónimo de"* conforme fue establecida. Se analizaron las oraciones involucradas y los documentos de donde proceden, las consultas se realizaron en la base de datos que contiene

dicha información. En el análisis la oración 1 se encuentra en 237 documentos, la oración 2 en 1 documento y la oración 3 pertenece a 2 documentos. Además, se observó que la oración 1 comparte un mismo documento con la oración 2, mientras que la oración 3 se encuentra en documentos independientes. Este análisis tiene el objetivo de verificar si los hechos obtenidos proceden de la misma fuente (documento) o no. En caso de que los hechos procedan de la misma fuente se puede asumir que comparten el mismo contexto. En caso contrario se puede asumir que se "ha descubierto" nuevo conocimiento.

Regla 1b

Al ejecutar la regla se obtuvieron resultados en las tres bases de hechos. El resultado presentado a continuación pertenece a la Base1:

```
?- regla1b(A,B,C,Rab,Rbc,Rac).
A = puestotrabajo("diputada"),
B = partidopolitico("Movimiento de Regeneración Nacional"),
C = partidopolitico("Morena"),
Rab = relacion("del partido"),
Rbc = relacion("tiene el acrónimo de"),
Rac = relacion("es un puesto de trabajo en")
```

Una vez obtenidos los hechos resultantes se procede al análisis de ellos. Se buscaron las oraciones en las que aparece cada una de las tripletas. De los resultados se emplean las relaciones *Rab*, *Rbc* y *Rac* para rehacer las tripletas. Por ejemplo, de la relación *Rab* se obtiene la tripleta <*"diputada", "del partido", "Movimiento de Regeneración Nacional"*>, de este modo se analiza cada una de las tripletas y la información obtenida se describe en la Tabla 5.18. Este proceso se realiza para cada una de las relaciones en los resultados obtenidos.

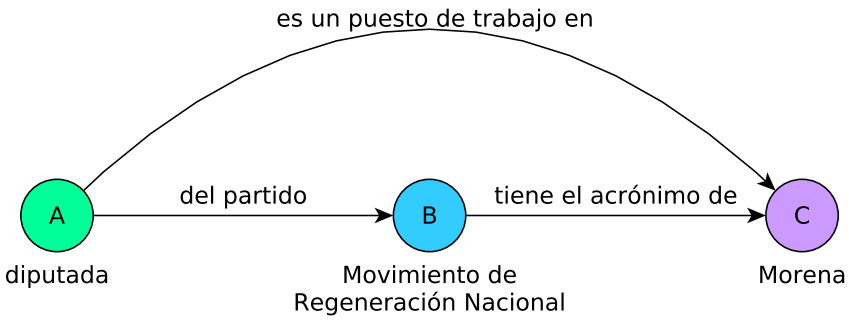

Fig. 5.18 Resultados de aplicar la regla 1b en la Base1.

Los resultados obtenidos se ilustran de forma gráfica en la Figura 5.18. Donde de acuerdo a la *Regla 1b* entre cada par de entidades debe contener una relación diferente. En la Tabla 5.18 la última columna contiene el texto de la oración, las entidades nombradas están identificadas por guiones bajos para separar las palabras y la clase, las relaciones están marcadas en negrita con excepción de las relaciones establecidas de forma manual. En la Tabla 5.18 la relación de $A \rightarrow B$ está indicada en negrita en el primer renglón. Las relaciones siguientes fueron definidas de forma manual, por lo que no aparecen directamente en el texto.

Tabla 5.18 Origen de hechos obtenidos de la Regla 1b.

#	IdD	IdO	Oración
1	257526	12312	En el pleno , solo se tuvo un voto en contra , el de la *diputada TIT* **del partido** *Movimiento de Regeneración Nacional PEX* , instituto político que será el que ocupe la mayoría en la 62 legislatura local
2	29171	1716	El secretario del Partido de la Revolución Democrática (PRD) en Quintana Roo , Carlos Montalbán , afirmó que la regidora Alejandra Cárdenas , quien recientemente se sumó al *Movimiento de Regeneración Nacional PEX* (*Morena PEX*) , realmente no tiene un capital político en Solidaridad ya que quienes la han acompañado son militantes de Cancún ; y aseguró que sólo buscan el interés económico , ya que cuando se le limitó el gasto del partido decidieron emigrar al partido que fundó Andrés Manuel López Obrador .
3	29179	1711	El gobernador , en un acto de apertura política y de reconocimiento a la voluntad del pueblo expresado en la elección del pasado primero de julio , manifestó que la *diputada TIT* federal electa de *Morena PEX* será un factor importante para ayudarle a Guerrero en esta segunda etapa de su gobierno .

En la Tabla 5.18 la oración 1 se encontró en 1 documento, la oración 2 se encuentra en 349 documentos y la oración 3 se identifica en 160 documentos. Sin embargo, la oración 2 y 3 comparten su procedencia de un documento. Así se comprueba que la entidad *"diputada"* se encuentra estrechamente relacionada con las entidades de tipo *partido político*: *"Movimiento de Regeneración Nacional"* y *"Morena"*.

Regla 2a

En la regla se utilizó la Base3, ya que las bases de hechos 1 y 2 no satisfacen la regla. La Figura 5.19 muestra de forma gráfica uno de los resultados obtenidos de la *Regla 2a*. Los hechos obtenidos al ejecutar la regla se describen a continuación:

```
?- regla2a(A,B,C,R).
A = documento("Tratado de Libre Comercio de América del Norte"),
```

```
B = documento("TLC"),
C = documento("TLCAN"),
R = relacion("tiene el acrónimo de")
```

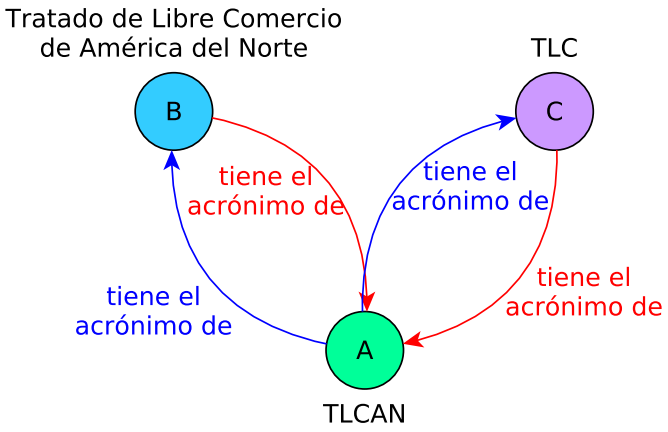

Fig. 5.19 Resultados de aplicar la regla 2a en la Base3.

El análisis de los hechos obtenidos en forma de tripleta se analizaron y se detallan en la Tabla 5.20. Una vez realizado el análisis de cada uno de los hechos se concluyo que ninguno de los hechos (tripletas) comparten el mismo documento. La relación que presentan es la definida de forma manual *"tiene el acrónimo de"*. Como se observa en la Tabla 5.20 conforme a la definición de la relación *"tiene el acrónimo de"* se cumple que *"TLC"*, *"TLCAN"* y *"Tratado de Libre Comercio de América del Norte"* sean tomados como *acrónimos* entre sí. Es claro que la relación de estos hechos es estrecha por la temática de todas las entidades. En análisis de las oraciones se observó que ninguna de las oraciones comparten un mismo documento, todas las oraciones provienen de documentos distintos.

Tabla 5.20 Origen de hechos obtenidos de la Regla 2a sobre la Base3.

#	IdD.	IdO	Oración
1	9506	475	Los principales retos que enfrentará Hacienda en los siguientes meses son la incertidumbre de la renegociación del *Tratado de Libre Comercio de América del Norte DOC* (*TLC DOC*) , los efectos de la reforma fiscal de Estados Unidos y la transición de Gobierno , señaló el funcionario .
2	1389006	102226	(Kushner) le pidió dejarle México a él porque tendría amarrada (la renegociación de) el *TLC DOC* (el *Tratado de Libre Comercio de América del Norte DOC*)
3	10446	536	Tras esto , López Obrador manifestó que tenderá la mano al Gobierno del país vecino , con el que México atraviesa unos tensos momentos por la renegociación del *Tratado de Libre Comercio de América del Norte DOC* (*TLCAN DOC*) y por las diferencias en los asuntos migratorios .
4	69102	3644	Estamos hablando con México sobre el *TLCAN DOC* (*Tratado de Libre Comercio de América del Norte DOC*) , y creo que vamos a poder resolverlo , afirmó Trump , sin dar más detalles .

Regla 2b

Los resultados obtenidos pertenecen a la Base3. Se debe cumplir que de A hacia B,C,D contengan una relación distinta, el mismo caso debe ocurrir con la relación de retorno de B,C,D hacia A. Al realizar los experimentos con la *Regla 2b* empleando la base de hechos 3, se obtienen una gran cantidad de resultados. Para analizar el comportamiento de esta reglas se fijo la entidad A como una entidad nombrada de clase *"persona"* y con un nombre específico ($A =$ *persona(Martha Erika Alonso)*). De los resultados obtenidos se tomó uno para ser analizado. Por esta razón en los resultados presentados no aparece la variable A, ya que fue definida directamente en la regla previo a su ejecución. Los hechos obtenidos se muestran en la Figura 5.20 de forma gráfica, y el resultado de la ejecución de lista a continuación:

```
?- regla2b(persona("Martha Erika Alonso"),B,C,D,Rab,Rba,Rac,Rca,Rad,Rda).
B = partidopolitico("PAN"),
C = organizacion("Tribunal Electoral del Poder Judicial de la
Federación"),
D = partidopolitico("Por Puebla al Frente"),
Rab = relacion("a agradeció el respaldo del"),
Rba = relacion("con una ventaja respalda triunfo a"),
Rac = relacion("celebró la decisión del"),
Rca = relacion("que ratificara el triunfo en la elección de"),
Rad = relacion("de la coalición"),
Rda = relacion("miembros respaldaron la candidatura de")
```

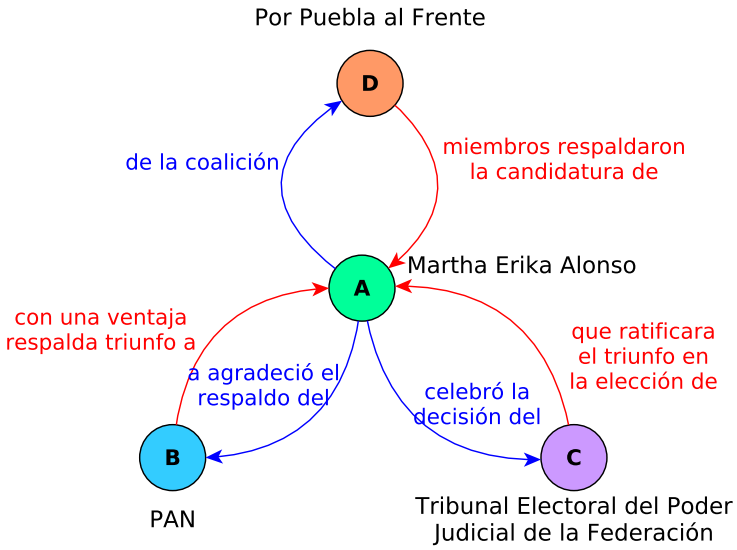

Fig. 5.20 Resultados de aplicar la regla 2b.

En la Figura 5.20 se observan de color azul a las relaciones de $A \rightarrow B$, $A \rightarrow C$ y $A \rightarrow D$. En color rojo a las relaciones $A \leftarrow B$, $A \leftarrow C$ y $A \leftarrow D$. El análisis de cada uno de los hechos (tripletas) obtenidos se describen en la Tabla 5.22.

Tabla 5.22 Origen de hechos obtenidos de la Regla 2b.

#	IdD.	IdO	Oración
1	935229	59934	**A** su vez , *Martha Erika Alonso PER* **agradeció el respaldo** de la dirigencia nacional y las consejeras **del** *PAN PEX* .
2	**533658**	28230	**Con una ventaja** de 122 mil votos , *PAN PEX* **respalda triunfo a** *Martha Erika Alonso PER*
3	489597	25111	*Martha Erika Alonso PER* **celebró la decisión del** *Tribunal Electoral del Poder Judicial de la Federación ORG* al ordenar un recuento de voto por voto en la elección para la gubernatura de Puebla en la que ella como candidata de la alianza por Puebla al Frente obtuvo la mayoría de voto y recibió la constancia de mayoría .
4	938181	60156	Buena noticia para la democracia **que** el *Tribunal Electoral del Poder Judicial de la Federación ORG* **ratificara el triunfo de** *Martha Erika Alonso PER* **en la elección** para gobernadora de Puebla .
5	524706	27221	Con casi hora y media de retraso , las 9 : 25 horas comenzó el recuento de los 3.7 millones de votos de la elección de gobernador de Puebla , el cual fue ordenado por la Sala Superior del Tribunal Electoral del Poder Judicial de la Federación para darle certeza al resultado de los comicios que tiene como contendientes a *Martha Erika Alonso PER* , **de la coalición** *Por Puebla al Frente PEX* , y Miguel Barbosa , de Juntos Haremos Historia .
6	**533658**	28231	**Miembros** del Partido Acción Nacional (PAN) e integrantes de la coalición *Por Puebla al Frente PEX* **respaldaron la candidatura de** *Martha Erika Alonso PER* , quien de acuerdo con el último conteo del PREP , la esposa de Rafael Moreno Valle cuenta con una ventaja de cuatro puntos por arriba de Luis Miguel Barbosa , candidato por Morena a la gubernatura del estado .

Para el análisis de los hechos se buscaron todas las oraciones donde aparecen además de los documentos a los que pertenecen dichas oraciones. Cada una de las oraciones de la Tabla 5.22 pertenecen a documentos distintos, con excepción de la segunda y sexta oración, ambas oraciones fueron encontradas en el mismo documento. Al observar los hechos obtenidos y la

información contenida en cada una de las oraciones, indican conocimiento en común, por lo que se puede decir que se han encontrado hechos que están ligados por la entidad *A* y pertenecen a un contexto similar. Además, estos hechos no se podrían conocer (con excepción de las oraciones 2 y 6) sin la inferencia realizada por Prolog, ya que las oraciones (1, 3, 4 y 5) provienen de documentos distintos y extraer su conocimiento cuando se tiene un gran volumen de datos no es una tarea trivial.

5.3.4 Regla 3a

Los resultados de la regla se ilustran de forma gráfica en la Figura 5.21. La regla aporta resultados en las tres bases de hecho, el resultado presentado pertenece a la Base3. Los hechos obtenidos contienen la relación *R "tiene el título de"* (relación definida de forma manual) y como foco a la entidad *A* (*"Arturo Zaldívar"*).

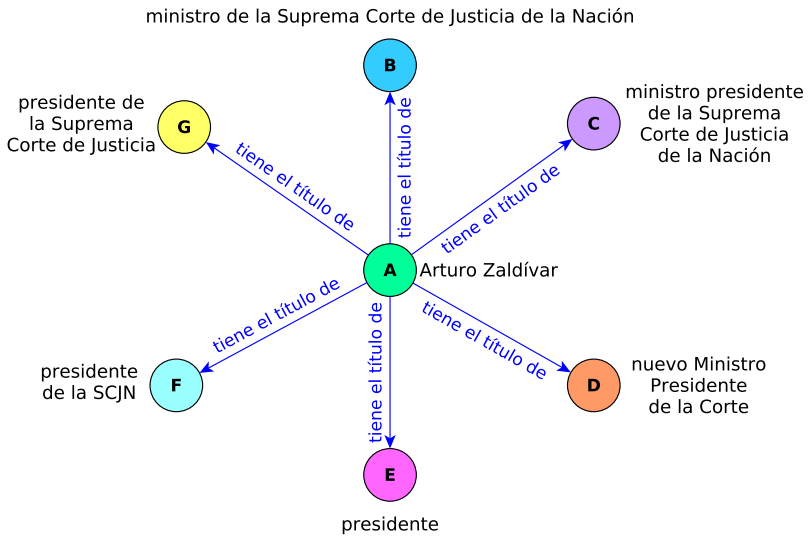

Fig. 5.21 Resultados de aplicar la regla 3a.

En el análisis realizado a las oraciones que contienen a las tripletas involucradas, se corroboró que las oraciones se encuentran en diferentes documentos. El objetivo de establecer dicha regla es encontrar las diferentes formas en las que se hace referencia a una misma entidad, como es el caso del *puesto de trabajo* de una *persona*. De igual manera, la regla permite conocer el nombre de la *organización* y/o el *partido político* donde labora una *persona*. También permite conocer el origen *geopolítico* de una *organización*. La Tabla 5.24 detalla las oraciones de donde

los hechos (tripletas) proceden, así como el identificador del documento al que pertenecen las oraciones.

Tabla 5.24 Origen de hechos obtenidos de la Regla 3a.

#	IdD.	IdO	Oración
1	2141264	174129	*Arturo Zaldívar PER* , *ministro de la Suprema Corte de Justicia de la Nación TIT* , informó que se cancelaron las sesiones por el brote de la polémica enfermedad en la nación mexicana
2	2089392	170042	*Arturo Zaldívar PER* , *ministro presidente de la Suprema Corte de Justicia de la Nación TIT* , consideró que el símbolo patrio representa a la nación , cuyos designios dependen del trabajo conjunto por el bien de todas las personas que la habitan .
3	1055558	72598	Él es *Arturo Zaldívar PER* , *nuevo Ministro Presidente de la Corte TIT*
4	1048299	71767	*Arturo Zaldívar PER* , nuevo *presidente TIT* de la Suprema Corte de Justicia de la Nación
5	1049352	71853	*Arturo Zaldívar PER* , nuevo *presidente de la SCJN TIT* .
6	2034880	164941	*Arturo Zaldívar PER* , *presidente de la Suprema Corte de Justicia TIT* , participará en las reuniones que el Presidente Andrés Manuel López Obrador sostiene con los padres de los normalistas de Ayotzinapa .

5.3.5 Regla 3b

Los hechos resultantes de la regla se ilustran en la Figura 5.22, donde la entidad $A =$ *"Andrés Manuel López Obrador"* es el punto de partida hacia las otras entidades. Como se observa, se obtuvieron diferentes relaciones a diferentes tipos de entidades: *evento, geopolítica (lugares habitados), fecha, tiempo* e *instalación*.

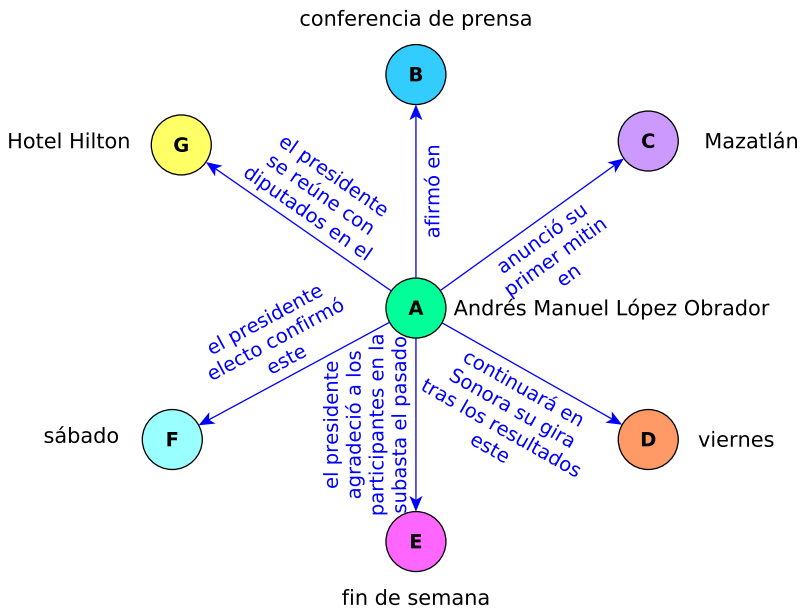

Fig. 5.22 Resultados de aplicar la regla 3b.

En la Tabla 5.26 se listan las oraciones de donde provienen los hechos obtenidos al ejecutar la regla. En el análisis de los hechos se verifico el origen de cada una de las oraciones, y cada una de ellas pertenece a un documento distinto. Las tripletas se resaltan en la Tabla 5.26 con letra cursiva para las entidades nombradas y en negrita para la relación entre las entidades. En la primera oración el hecho fue identificado y extraído intercambiando el orden de las entidades nombradas, por lo que la tripleta se lee $<$*"Andrés Manuel López Obrador"*, *"afirmó en"*, *"conferencia de prensa"*$>$. El conocimiento adquirido al ejecutar la regla centrada en la entidad $A =$ *"Andrés Manuel López Obrador"* indica como se relaciona con otras entidades, y como las oraciones pertenecen a distintos documentos se puede asumir que se ha encontrado *"conocimiento nuevo"*, y este solo se podría conocer analizando cada uno de los documentos involucrados por separado.

Tabla 5.26 Origen de hechos obtenidos de la Regla 3b.

#	IdD.	IdO	Oración
1	1328624	96552	**En** *conferencia de prensa EVT* , *Andrés Manuel López Obrador PER* **afirmó** que en el tema de Venezuela No quiere meterse en opiniones de ninguna índole , pero llamó a las partes involucradas a que se sienten y dialoguen .
2	712947	40204	*Andrés Manuel López Obrador PER* , **anunció su primer mitin** de agradecimiento **en** *Mazatlán GPE* que cuando asuma el poder transformará al Banco del Ahorro Nacional y Servicios Financieros , así como le cambiará el nombre por el de Banco del Bienestar .
3	501509	25909	*Andrés Manuel López Obrador PER* **continuará este** *viernes DAT* **en Sonora su gira** de agradecimiento , **tras los resultados** de los comicios del pasado 1 de julio .
4	1323287	96102	**El presidente** *Andrés Manuel López Obrador PER* , **agradeció a los participantes en la subasta** de autos de lujo que se realizó **el pasado** *fin de semana TIM* en la base militar de Santa Lucía , la cual calificó de exitosa durante La Mañanera de este lunes , realizada en Palacio Nacional .
5	860892	51075	**El presidente electo** de México , *Andrés Manuel López Obrador PER* , **confirmó este** *sábado DAT* que los mexicanos decidirán el 21 de marzo de 2019 si se abren expedientes para procesar a expresidentes de México .
6	752650	42595	**El Presidente** electo , *Andrés Manuel López Obrador PER* , **se reúne con diputados** y senadores **en el** *Hotel Hilton FAC* .

5.3.6 Regla 4

El objetivo de la regla es observar la relación que presentan las entidades B y C. Como se observa en la Figura 5.23 la entidad $A =$ *"PAN"* se vincula directamente hacia $B =$ *"Nicolás Maduro"* y $C =$ *"AMLO"*. El hecho obtenido entre las entidades B y C es la tripleta $<$*"Nicolás Maduro", "viene a posesión de", "AMLO"*$>$, la relación resultante aporta conocimiento aunque

este no se encuentre directamente relacionado con la entidad *A*, siendo el punto de partida de la regla.

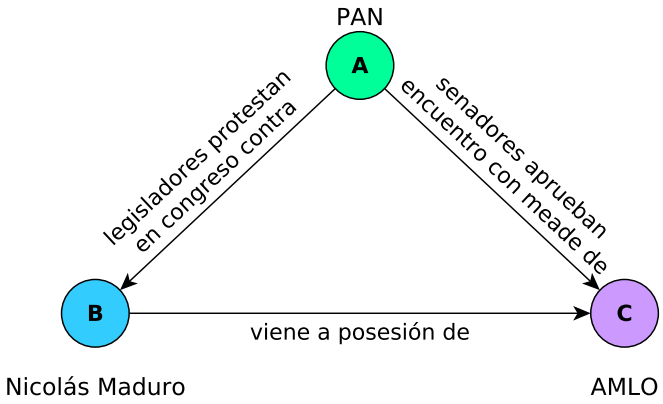

Fig. 5.23 Resultados de aplicar la regla 4.

Para cada una de las oraciones de la Tabla 5.28 se verificaron los documentos de los que proceden. Ninguna de las oraciones comparte el mismo documento. Las relaciones se encuentran resaltadas en negrita. Con la idea de que las oraciones se encuentran en distintos documentos, se puede decir la regla aporta *"nuevo conocimiento"* que ha sido *"descubierto"* al aplicar la regla sobre la Base1.

Tabla 5.28 Origen de hechos obtenidos de la Regla 4.

#	IdD.	IdO	Oración
1	896861	54963	**Legisladores** del *PAN PEX* **protestan contra** *Nicolás Maduro PER* **en Congreso** de la Unión
2	158497	7531	**Senadores** del PRI , *PAN PEX* y pt morena **aprueban encuentro** de *AMLO PER* **con** Meade
3	710043	40020	**Viene** *Nicolás Maduro PER* **a** toma de **posesión de** *AMLO PER* , confirma Ebrard

5.3.7 Regla 5

Es una ampliación del la *Regla 4* y conlleva al mismo objetivo. Observar la relación que existe entre dos entidades con un nivel mas extendido. Los resultados de ejecutar la regla se ilustran en la Figura 5.24. La regla inicia con la entidad $A =$ *"108 mil pesos"* que es una entidad de clase *moneda*. La información que proporciona los hechos resultantes indican la *fecha* en la que *"AMLO"* inicie ganando la cantidad especificada por *A*, además representa la fecha en la que el *"presidente"* tome posesión. A su vez, la solicitud de intervención por parte de *"AMLO"* a *"Peña Nieto"*. Por último, se indica que *"presidente es el título de Peña Nieto"*.

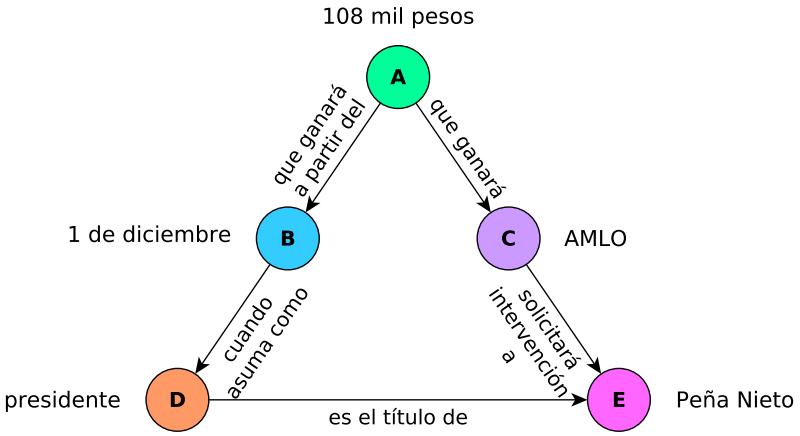

Fig. 5.24 Resultados de aplicar la regla 5.

Para cada uno de los hechos obtenidos (tripletas) se buscaron las oraciones de las que provienen así como los documentos de donde se seleccionaron dichas oraciones, como se detalla en la Tabla 5.30. En el análisis de los hechos se observo que no existe un documento en común para las cinco oraciones, es decir; provienen de diferentes documentos. La *Regla 5* nos indica que la información que se obtiene de los hechos resultantes aporta *"conocimiento nuevo"*, debido a que provienen de documentos distintos.

Tabla 5.30 Origen de hechos obtenidos de la Regla 5.

#	IdD.	IdO	Oración
1	23782	1448	Los *108 mil pesos MNY* **que ganará a partir del** *1 de diciembre DAT* , advirtió , serán el techo salarial universal para toda la administración pública y los poderes .
2	41009	2258	En conclusión , si todos estos funcionarios aceptaran reducir su salario a los *108 mil pesos MNY* **que ganará** *AMLO PER* ,
3	21715	1355	El tabasqueño señaló que las reuniones preparatorias que está teniendo su futuro gabinete son para iniciar la transformación del país a partir del *1 de diciembre DAT* , **cuando asuma como** *presidente TIT*
4	140307	6679	*AMLO PER* **solicitará intervención a** *Peña Nieto PER* .
5	10461	645	Entonces tenemos mucho trabajo importante que continuar durante los próximos meses con la administración del *presidente TIT Peña Nieto PER* .

5.3.8 Regla 6

En la regla deben de cumplirse en primera instancia que las relaciones entre entidades deben ser diferentes. De los resultados obtenidos al ejecutarla, se seleccionó un resultado y sus hechos son presentados en forma gráfica en la Figura 5.25. El punto de partida de la regla es la entidad $A =$ *"Ley de Remuneraciones"* siendo una entidad de tipo *documento*, y se relaciona con las entidades $B =$ *"Congreso de la Unión"* de la clase *organización*, $C =$ *"Presidente"* de tipo *puesto de trabajo*, $D =$ *"108 mil pesos"* del tipo *moneda*, $E =$ *"AMLO"* perteneciente a la clase *persona*, $F =$ *"presidente"* del tipo *puesto de trabajo* y $G =$ *"ayer"* que es una entidad de tipo *tiempo*. Al observar los resultados en los hechos de las tripletas $B \rightarrow C$, $D \rightarrow E$ y $F \rightarrow G$ sus respectivas relaciones (color rojo) no se encuentran en el mismo contexto de las relaciones que parten directamente de la entidad A, solo aquellas que si lo hacen (color azul).

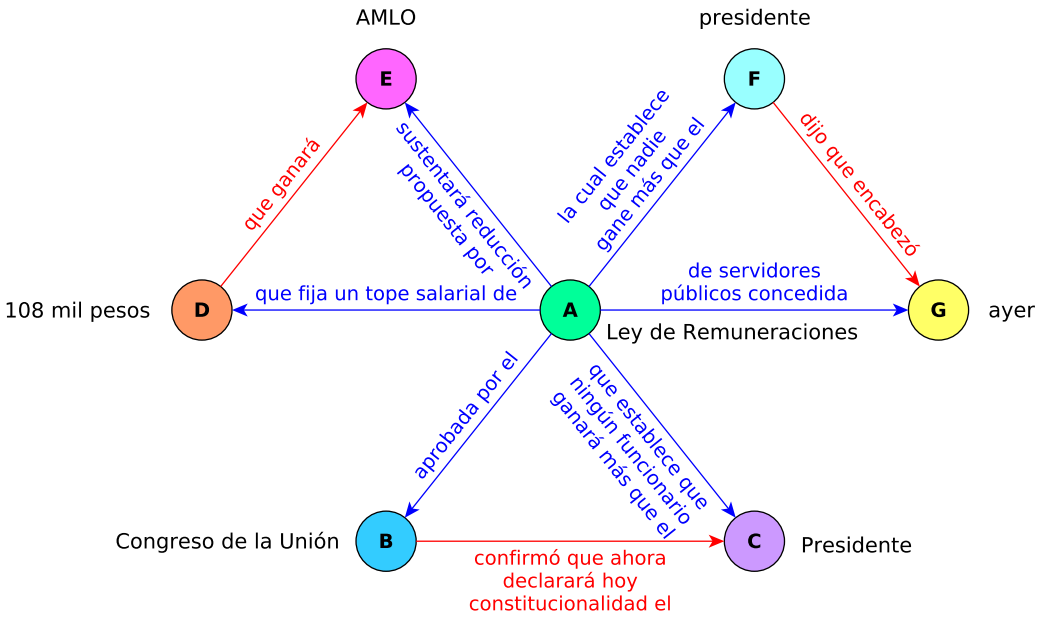

Fig. 5.25 Resultados de aplicar la regla 6.

Al analizar las oraciones en la Tabla 5.32 se observó que las oraciones 1, 3, 4, 6, 7, 8 y 9 se encuentran en diferentes documentos, a diferencia de las oraciones 2 y 5 que provienen del mismo documento, como se puede observar en el Id del documento. Las relaciones están resaltadas en negrita dentro de las oraciones. De los hechos se puede obtener conocimiento relevante sobre cada una de las relaciones que parten de la entidad A. En cambio, para los hechos unidos por relaciones en color rojo poseen conocimiento únicamente entre ellos, con excepción de $D \rightarrow E$. Si se conjunta la información que aportan estos hechos con $A \rightarrow D$ y $A \rightarrow E$ se obtiene que la *"Ley de Remuneraciones fija un tope salarial de 108 mil pesos, esa cantidad es la que ganará AMLO, y además esa ley sustenta una reducción propuesta por AMLO"*. Este se puede decir que es *"nuevo conocimiento"* que se ha obtenido con la ejecución de la regla.

Tabla 5.32 Origen de hechos obtenidos de la Regla 6.

#	IdD.	IdO	Oración
1	1149851	80543	Estas modificaciones se propondrán en la sesión del Consejo General programada para este miércoles , donde también quedarán definidos los cambios a los salarios de toda la plantilla del INE ya acorde a los criterios de la nueva *Ley de Remuneraciones DOC* **aprobada** a finales del año pasado **por el** *Congreso de la Unión ORG* .
2	**464426**	23895	Con 433 votos a favor , 9 en contra y una abstención diputados avalaron en lo general la *Ley de Remuneraciones DOC* **que establece que ningún funcionario ganará más que el** *Presidente TIT* y la reducción de salario a burócratas .
3	1363561	100132	El presidente de la Junta de Coordinación Política en el Palacio de San Lázaro , Mario Delgado , **confirmó** a su bancada **que** , **ahora** sí , el *Congreso de la Unión ORG* **declarará hoy constitucionalidad** de las reformas para crear la nueva institución de seguridad , a la que **el** *Presidente TIT* apuesta como el ariete para abatir el crimen y la violencia .
4	927415	59251	Luego de que senadores del PAN presentaron una acción de inconstitucionalidad contra la *Ley de Remuneraciones DOC* , **que fija un tope salarial de** *108 mil pesos MNY* para los funcionarios , el Presidente Andrés Manuel López Obrador comentó que los legisladores están en su derecho , pero ironizó con el tema .
5	**464426**	23894	Diputados aprobaron *Ley de Remuneraciones DOC* , que regula salarios de funcionarios y **sustentará reducción propuesta por** *AMLO PER* ; va al Ejecutivo .
6	41009	2258	En conclusión , si todos estos funcionarios aceptaran reducir su salario a los *108 mil pesos MNY* **que ganará** *AMLO PER* ,
7	951616	61760	Este lunes , en un acto histórico jueces y magistrados de todo el país se manifestaron en contra de la *Ley de Remuneraciones DOC* **la cual establece que nadie gane más que el** *presidente TIT* , medida que no fue bien recibida por el poder judicial y el Instituto Nacional Electoral .

Continuación de la Tabla 5.32

#	IdD.	IdO	Oración
8	934431	59788	El Presidente Andrés Manuel López Obrador criticó la suspensión a la *Ley de Remuneraciones DOC* **de servidores públicos concedida** *ayer TIM* por la Suprema Corte .
9	1810917	143322	No tiene que ver ni cómo se encuentra el presupuesto ni presionar a los contribuyentes , pero lo que sí queremos asegurarnos es que todos estamos contribuyendo en la forma en que la ley lo dispone , **dijo** durante la conferencia de prensa **que encabezó** *ayer TIM* el *presidente TIT* Andrés Manuel López Obrador , en Palacio Nacional

5.3.9 Regla 7

La regla establece la relación que hay entre dos reglas transitivas creando una relación entre sus puntos inicial y final. En primera instancia se deben cumplir las reglas *transitivas*, para que la regla sea válida para $A \rightarrow B$ internamente se tiene que cumplir internamente que $A \rightarrow E \rightarrow B$. Lo mismo ocurre en para las entidades de $C \rightarrow D$, se tiene que cumplir $C \rightarrow F \rightarrow D$. Finalmente la regla define que exista una relación de $A \rightarrow C$ (punto inicial) y una relación de $B \rightarrow D$. Según a la definición de la regla no devuelve las entidades E y F así como las relaciones de ellas.

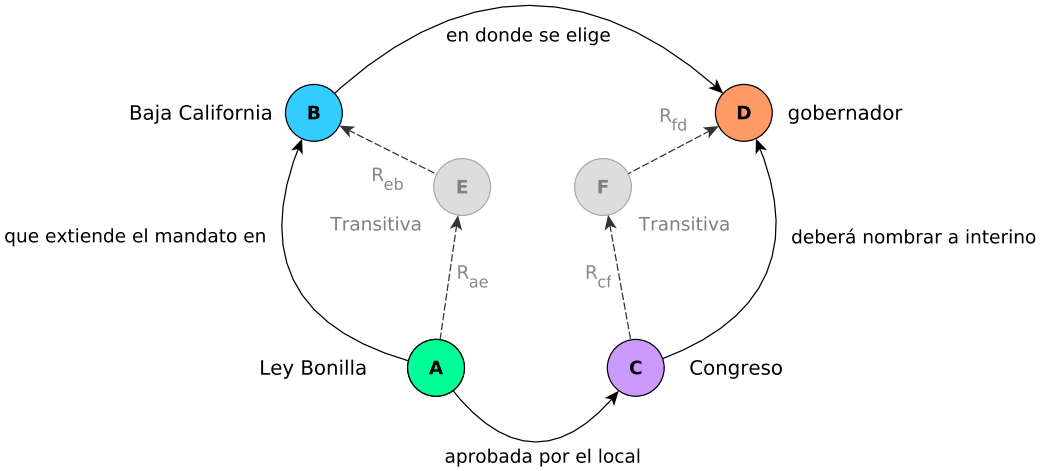

Fig. 5.26 Resultados de aplicar la regla 7.

Los hechos obtenidos al ejecutar la regla se ilustran en la Figura 5.26. De los resultados obtenidos se observan que la entidad $A =$ *"Ley Bonilla"* es de la clase *documento* y la entidad $C =$ *"Congreso"* es del tipo *organización*, siendo estas los puntos iniciales de las reglas transitivas internas. De los hechos obtenidos en este resultado se tiene que la *"Ley Bonilla extiende el mandato en Baja California y en este lugar se elige gobernador"* además se sabe que la *"Ley Bonilla es aprobada por el local Congreso"*. Sin embargo, los resultados de $C \rightarrow D$ no se encuentran en el mismo contexto al de los otros hechos obtenidos, como se puede corroborar en la Tabla 5.33.

Tabla 5.33 Origen de hechos obtenidos de la Regla 7.

#	IdD.	IdO	Oración
1	1689937	131491	Por otro lado , el partido también ha seguido tomando medidas contra la reforma denominada como la *Ley Bonilla DOC* , **que extiende el mandato en** *Baja California GPE* .
2	1015932	68786	*Congreso ORG* **deberá nombrar a** *gobernador TIT* **interino** de Puebla
3	1789396	141399	Incluye Baja California , se le cuestinó en relación a la *Ley Bonilla DOC* **aprobada por el** *Congreso ORG* **local** para alargar el periodo del Gobernador electo
4	570833	30707	El tricolor tiene ante sí un negro panorama ; el próximo año deberá enfrentar elecciones en cinco entidades federativas (*Baja California GPE* , el único estado **en donde se elige** *gobernador TIT* , Durango , Tamaulipas , Aguascalientes y Quintana Roo) y sin la organización que antes tuvo , sin recursos económicos , su escenario parece ser la pérdida del registro estatal .

La Tabla 5.33 lista las oraciones que contiene los hechos (tripletas) ilustrados en la Figura 5.26. En la Tabla 5.33 las entidades nombradas se describen en cursiva y la relación entre ellas en negrita. El análisis de cada una de las oraciones indicó que las oraciones provienen de documentos independientes, lo que indica que los hechos obtenidos se relacionan entre sí debido a la especificación de la regla, y por ello se asume que se han *"descubierto nuevos hechos"* o *"nuevo conocimiento"*. Esto con excepción a los hechos de la segunda oración, debido a que su contexto es distinto al de las otras oraciones.

5.3.10 Regla 8

Uno de los resultados obtenidos en la ejecución de la regla se ilustra en la Figura 5.27. La regla tiene como centro y punto de origen la entidad A, y está formada por tres reglas transitivas además de estar ligadas en los extremos. Las variable $A =$ *"Ley de Austeridad Republicana"* es de la clase *documento*, y está vinculada con la entidad $E =$ *"Cámara de Diputados"* del tipo *organización*, con la entidad $F =$ *"Morena"* de tipo *partido político* y con la entidad $G =$ *"Senado"* que pertenece a la clase *organización*.

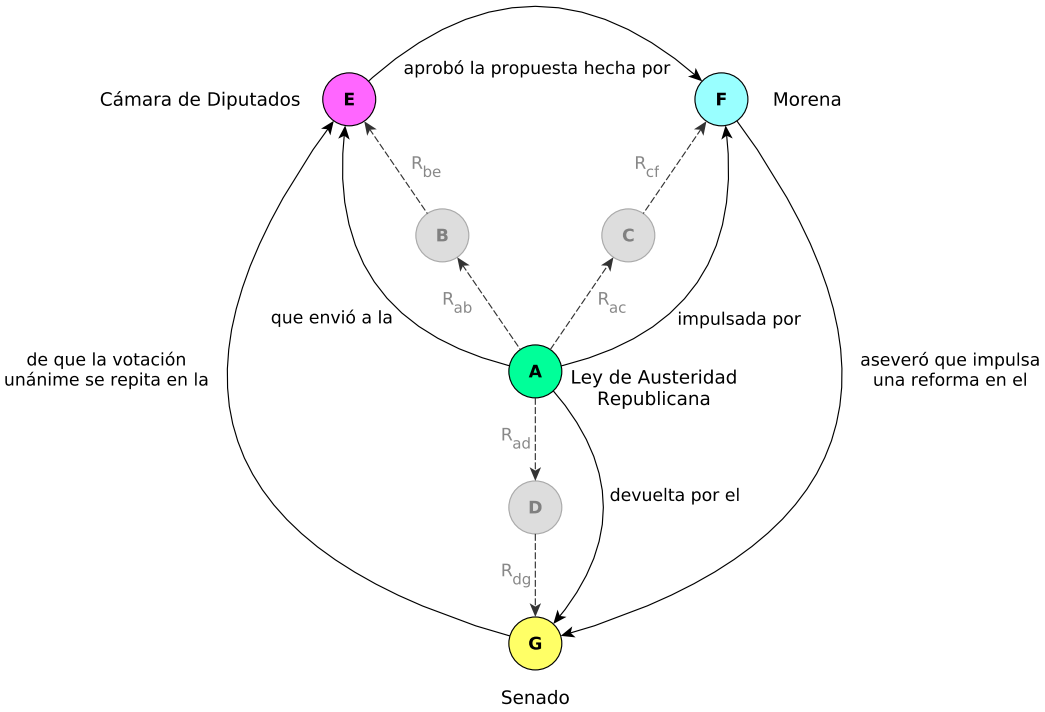

Fig. 5.27 Resultados de aplicar la regla 8.

De los hechos obtenidos de la regla se sabe que *"Ley de Austeridad Republicana fue enviada a la Cámara de Diputados, además la ley fue impulsada por Morena y fue devuelta por el Senado"*. La información que aporta en los hechos externos ligados de las reglas transitivas internas dice que, la *"Cámara de Diputados aprobó la propuesta hecha por Morena, a su vez Morena aseveró que impulsa una reforma en el Senado y este menciona de que la votación unánime se repita en la Cámara de Diputados"*. La Tabla 5.35 describe en detalle el contexto de cada uno de los hechos obtenidos. En cuanto al análisis de las oraciones presentadas en la Tabla 5.35 que corresponden a cada una de las tripletas obtenidas (hechos), se corroboró la

procedencia de cada una de ellas, y ninguna proviene del mismo documento. Las relaciones se destacan en negrita y las entidades en cursiva. Como se observa en la Tabla 5.35, las oraciones 1, 2 y 3 presentan una relación estrecha en coincidencia con la definición de la *Regla 8*. Sin embargo, las oraciones 4, 5 y 6 aunque se encuentran ligadas a las entidades de las primeras tres oraciones, estas últimas describen hechos en una temática distinta en cada una de ellas.

Tabla 5.35 Origen de hechos obtenidos de la Regla 8.

#	IdD.	IdO	Oración
1	1257672	89989	Explicó que esto está planteado en la *Ley de Austeridad Republicana DOC* , **que envió a la** *Cámara de Diputados ORG* , por lo que hizo un llamado a los legisladores a aprobarla .
2	928289	59241	En la Cámara de Diputados tienen listo el dictamen de la *Ley de Austeridad Republicana DOC* , **impulsada por** *Morena PEX* , que entre otras cosas establecerá , de aprobarse , la cancelación de bonos y compensaciones extraordinarias para todos los funcionarios públicos , legisladores , así como para los integrantes del Poder Judicial y los trabajadores de los órganos autónomos .
3	1907612	152632	Las Comisiones Unidas de Hacienda y Presupuesto de la Cámara de Diputados aprobaron la *Ley de Austeridad Republicana DOC* , **devuelta por el** *Senado ORG* para establecer una restricción de 10 años a ex funcionarios para emplearse en la iniciativa privada .
4	1793583	141816	La *Cámara de Diputados ORG* **aprobó la propuesta hecha por** *Morena PEX* , en la cual sugiere modificar los artículos 17 y 51 de la ley Orgánica para poder rotar el liderazgo de la Mesa Directiva
5	856422	50783	**Aseveró que** *Morena PEX* , **en el** *Senado ORG* , **impulsa una reforma** para tipificar la corrupción como delito grave .
6	1327642	96526	Existen posibilidades **de que la votación unánime** para la aprobación de la Guardia Nacional en el *Senado ORG* **se repita en la** *Cámara de Diputados ORG* , aseguró el presidente de la Junta de Coordinación Política (Jucopo) en San Lázaro , Mario Delgado .

5.4 Grafo de conocimiento

En esta sección se presentan los grafos de conocimiento que se construyen a partir del conjunto de hechos (tripletas) obtenidos en la inferencia de Prolog, siguiendo los lineamientos establecidos por la W3C empelando RDF[1]. Para la construcción se hace uso de la biblioteca de Python RDFlib que cuenta con los esquemas básicos definidos como son "*RDF*", "*RDFS*", "*FOAF*", entre otros, lo que facilita la definición del *modelo de datos RDF*[2] (<*subject*> <*predicade*> <*object*>). Las entidades nombradas son definidas como los *nodos* y las relaciones (predicados) se definen como *aristas*, las aristas tienen dirección, parten del <*sujeto*> y tienen como objetivo al <*objeto*>, por lo que el grafo de conocimiento es dirigido.

Tabla 5.37 Lista de clases empleadas en la construcción del grafo de conocimiento.

No	Código	Nombre	Esquema
1	PER	Persona	http://xmlns.com/foaf/0.1/Person
2	TIT	Cargo o Puesto	http://dbpedia.org/ontology/PersonFunction
3	ORG	Organización	http://dbpedia.org/ontology/Organization
4	GPE	Geopolítica	http://dbpedia.org/ontology/PopulatedPlace
5	FAC	Instalación	http://dbpedia.org/ontology/Building
6	DAT	Fecha	http://lke.buap.mx/0.1/ontology/DateExpression
7	MNY	Moneda	http://dbpedia.org/ontology/Currency
8	DOC	Documento	http://xmlns.com/foaf/0.1/Document
9	PRO	Producto	http://schema.org/Product
10	PEX	Partido Político	http://dbpedia.org/ontology/PoliticalParty
11	EVT	Evento	http://dbpedia.org/ontology/Event
12	TIM	Tiempo	http://lke.buap.mx/0.1/ontology/TimeExpression
13	AGE	Edad	http://lke.buap.mx/0.1/ontology/Age
14	DEM	Gentilicio	http://lke.buap.mx/0.1/ontology/Demonym
15	LOC	Lugar	http://dbpedia.org/ontology/Place

El grafo de conocimiento se construye en base a los hechos obtenidos. La Tabla 5.37 describe los esquemas para definir la clase a la que pertenece cada una de las entidades nombradas. En la construcción se descartan entidades duplicadas, lo que ocurre a menudo en la obtención

[1]https://www.w3.org/TR/rdf11-concepts/
[2]https://www.w3.org/TR/rdf11-primer/#section-data-model

de hechos. Los esquemas representan el "*tipo*" o la *"clase"* al que pertenecen los datos, en estos grafos se utilizó el esquema **FOAF** para entidades de tipo *persona* y *documento*. El esquema **schema** para la entidad *producto* y el esquema **DBPedia** se uso para ocho entidades. El esquema **lke.buap.mx/0.1/** se ha definido para las entidades de *fecha* y *tiempo* debido a que casi en su totalidad las entidades no presentan un formato válido. Además se ha usado este último esquema para definir el tipo de entidades *edad* y *gentilicio*.

Los hechos obtenidos de ejecutar la *Regla 1a* suman un total de 4 de los 86,871 hechos de los que se compone la Base3. La regla solo pudo ser satisfecha por la Base3. Una vez obtenidos los hechos se transforman para construir un grafo de conocimiento en formato XML, como se ilustra en la Figura 5.29.

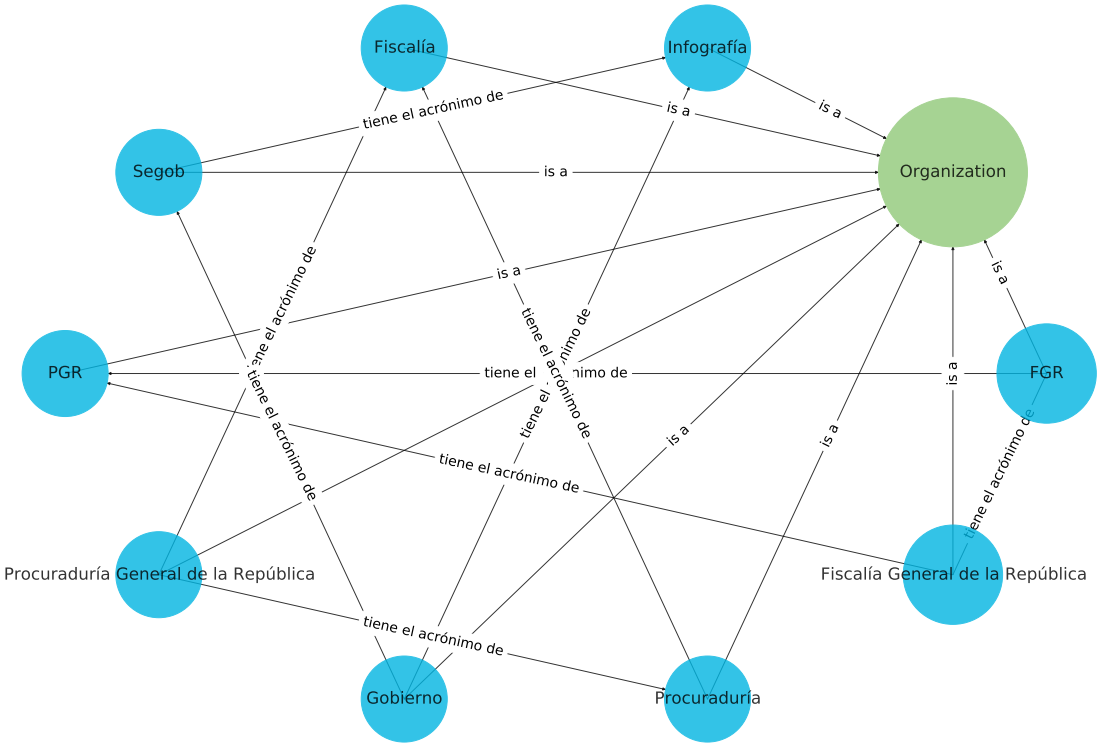

Fig. 5.28 Grafo de conocimiento de los hechos de la regla 1a.

La Figura 5.28 muestra de forma gráfica los hechos obtenidos al ejecutar la *Regla 1a*. En al figura solo se dibujan las relaciones, las entidades y la clase a la que pertenece cada entidad, se omiten las URL que suele acompañar a cada una de las etiquetas, esto con el objetivo de tener una vista más clara de los hechos obtenidos.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:lke="http://lke.buap.mx/0.1/relation/"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/FGR">
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
    <lke:tiene_el_acrónimo_de rdf:resource="http://lke.buap.mx/0.1/organization/PGR"/>
    <lke:tiene_el_acrónimo_de
    rdf:resource="http://lke.buap.mx/0.1/organization/Fiscalía_General_de_la_República"/>
    <foaf:name xml:lang="es">FGR</foaf:name>
  </rdf:Description>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/Segob">
    <lke:tiene_el_acrónimo_de
    rdf:resource="http://lke.buap.mx/0.1/organization/Infografía"/>
    <foaf:name xml:lang="es">Segob</foaf:name>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
  </rdf:Description>
  <rdf:Description
  rdf:about="http://lke.buap.mx/0.1/organization/Fiscalía_General_de_la_República">
    <foaf:name xml:lang="es">Fiscalía General de la República</foaf:name>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
    <lke:tiene_el_acrónimo_de rdf:resource="http://lke.buap.mx/0.1/organization/FGR"/>
    <lke:tiene_el_acrónimo_de rdf:resource="http://lke.buap.mx/0.1/organization/PGR"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/Procuraduría">
    <lke:tiene_el_acrónimo_de rdf:resource="http://lke.buap.mx/0.1/organization/Fiscalía"/>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
    <foaf:name xml:lang="es">Procuraduría</foaf:name>
  </rdf:Description>
  <rdf:Description
  rdf:about="http://lke.buap.mx/0.1/organization/Procuraduría_General_de_la_República">
    <lke:tiene_el_acrónimo_de
    rdf:resource="http://lke.buap.mx/0.1/organization/Procuraduría"/>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
    <foaf:name xml:lang="es">Procuraduría General de la República</foaf:name>
    <lke:tiene_el_acrónimo_de rdf:resource="http://lke.buap.mx/0.1/organization/Fiscalía"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/Gobierno">
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
    <lke:tiene_el_acrónimo_de
    rdf:resource="http://lke.buap.mx/0.1/organization/Infografía"/>
    <foaf:name xml:lang="es">Gobierno</foaf:name>
    <lke:tiene_el_acrónimo_de rdf:resource="http://lke.buap.mx/0.1/organization/Segob"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/Fiscalía">
    <foaf:name xml:lang="es">Fiscalía</foaf:name>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/PGR">
    <foaf:name xml:lang="es">PGR</foaf:name>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://lke.buap.mx/0.1/organization/Infografía">
    <foaf:name xml:lang="es">Infografía</foaf:name>
    <rdf:type rdf:resource="http://dbpedia.org/ontology/Organization"/>
  </rdf:Description>
</rdf:RDF>
```

Fig. 5.29 Grafo de conocimiento en formato XML de los hechos de la regla 1a.

La Figura 5.30 muestra los hechos obtenidos de ejecutar la *Regla 1b* usando la Base1. El grafo está compuesto por 46 nodos y 134 aristas. De los 46 nodos, 9 son nodos de tipo *clase* (Event, Demonym, PersonFunction, PopulatedPlace, Person, Organization, PoliticalParty, DateExpression, y Building).

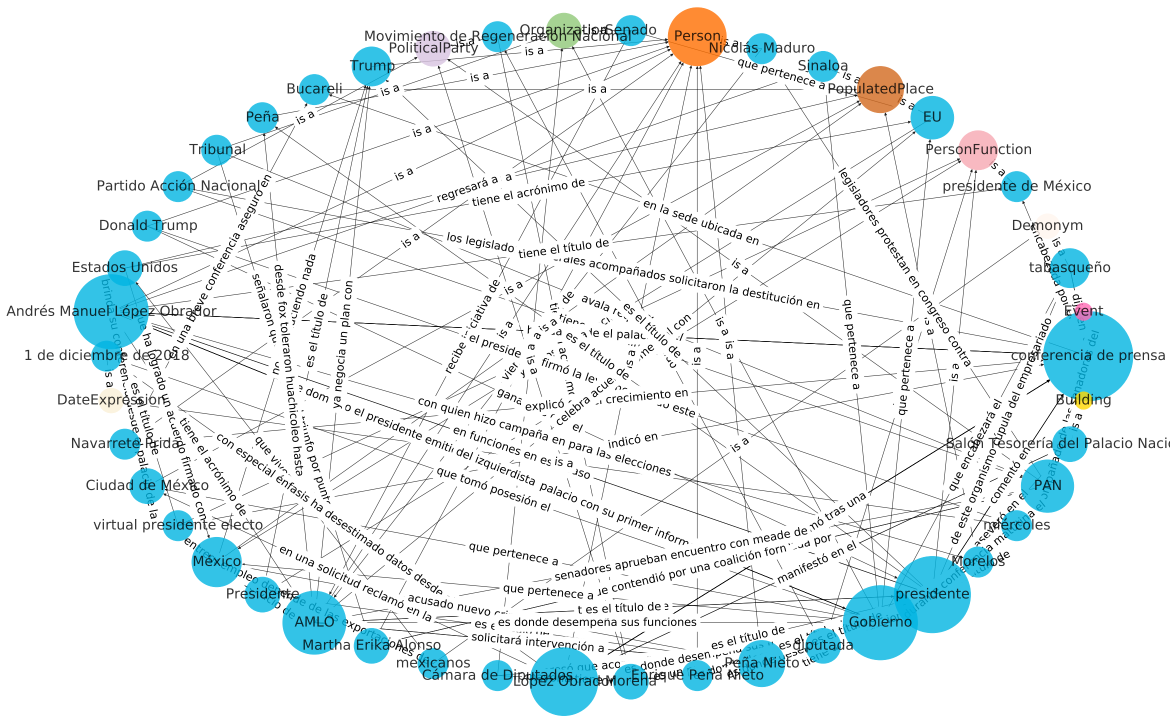

Fig. 5.30 Grafo de conocimiento de los hechos de la regla 1b.

El tamaño de los nodos de la Figura 5.30 es diferente debido a que se dibujaron en forma proporcional a su grado, es decir, la cantidad de aristas que inciden en el. Así, los nodos *"conferencia de prensa", "Andrés Manuel López Obrador", "presidente", "Gobierno", "López Obrador" y "AMLO"* son los que tienen un mayor número de aristas (entrantes y/o salientes) incidiendo en ellos.

La Tabla 5.38 describe en detalle la regla que se usó y sobre que base de hechos para obtener resultados, así como la cantidad de hechos que devolvió el interprete de Prolog al ejecutar cada regla. También se listan los nodos clase (**NC**) y el total de los nodos (**TN**) incluyendo los nodos clase. Finalmente en la última columna se listan las aristas por cada uno de los grafos generados.

Tabla 5.38 Recuento de nodos y aristas sobre los hechos obtenidos de las reglas.

No	**Regla**	**Bases**	**Hechos**	**NC**	**TN**	**Aristas**
1	1a	Base3	4	1	10	19
2	1b	Base1	138	9	46	134
3	2a	Base3	10	4	17	31
4	2b	Base3	101,790	6	20	71
5	3a	Base3	720	2	9	13
6	3b	Base2	10,000	9	25	33
7	4	Base1	138	9	46	134
8	5	Base1	867	12	71	213
9	6	Base3	10,000	7	16	95
10	7	Base3	10,000	8	33	134
11	8	Base3	10,000	3	7	92

La columna de **Hechos** en la Tabla 5.38 hace referencia al número de resultados para cada iteración que satisface a la relación en cuestión. Sin embargo, muchos hechos (tripletas) dentro de cada resultado individual se repite en múltiples ocasiones con los hechos de uno o muchos otros resultados. En la construcción de un grafo de conocimiento, se toman las partes de la tripleta para formar las relaciones RDF, por ende en este proceso se eliminan los hechos redundantes.

Capítulo 6

Conclusiones

Este capítulo describe las conclusiones sobre la adquisición de hechos de forma automática, que involucra el reconocimiento de entidades nombradas, la extracción automática de relaciones entre entidades nombradas, la construcción de la base de hechos y reglas lógicas, así como la construcción del grafo de conocimiento.

6.1 Conjunto de datos

Los datos empleados en este trabajo provienen de documentos no estructurados. Para ello, se implementó un crawler con el objetivo de extraer documentos de la Web, estos documentos provienen de sitios sobre noticias reportadas en cada uno de los estados de la República Mexicana. Del conjunto de datos obtenido se realizó un pre-procesamiento, extrayendo el texto de las páginas HTML, eliminado símbolos y caracteres no deseados.

Del conjunto de datos se filtraron los documentos en el dominio político. Como paso siguiente se realizó un etiquetado manual, definiendo 17 tipos (clases) de entidades nombradas con ayuda de un sistema web. En el etiquetado se usó el esquema IOBES. Este nuevo conjunto de datos contiene 250 documentos etiquetados y lleva por nombre Mx-news. Asimismo, se empleó el corpus CoNLL-2002 en idioma español para los experimentos en la siguiente sección.

6.2 Reconocimiento de Entidades Nombradas

En esta sección se desarrollo un modelo para reconocer y extraer entidades nombradas en documentos no estructurados, específicamente en el dominio de noticias políticas en idioma

español. Para ello se usaron dos enfoques, el primero basado en modelos probabilísticos y el segundo basado en aprendizaje profundo. Los experimentos se realizaron con un modelo CRF y con un modelo de redes neuronales recurrentes, se empleó un conjunto de características base como son: etiquetas POS, banderas para indicar la forma de la palabra e identificar el inicio y fin de las oraciones, para el modelo CRF. Los experimentos consistieron en analizar el comportamiento de una clase (junto con la clase "O"), para después agregar una segunda clase, repitiendo este proceso hasta haber realizado experimentos con todas las clases. Este proceso se hizo a partir de la clase más dispersamente etiquetada hasta la clase más densamente etiquetada y viceversa, estos procedimientos se aplicaron a los conjuntos de datos CoNLL-2002 y Mx-news. Además se emplearon dos formas de evaluación, la primera consiste en evaluar las etiquetas individuales de la etiqueta tomando en cuenta el etiquetado IOB e IOBES. La segunda evaluación no toma en cuenta el correcto orden del etiquetado IOB e IOBES dentro de una entidad predecida y verdadera, la entidad nombrada predecida a evaluar será tomada como válida, siempre y cuando las etiquetas que indican la clase sean las mismas del conjunto de prueba (verdaderas), sin importar que haya sido mál reconocida en términos del esquema IOB o IOBES.

Las ventajas del modelo CRF, recaen en el buen rendimiento para reconocer entidades nombradas cuando se posee un conjunto de entrenamiento pequeño. Puede reconocer una entidad con múltiples etiquetas (IOBES), a diferencia de otros algoritmos de este tipo. Con un conjunto pequeño de características es capaz de obtener buenos resultados. El tiempo computacional es breve y consume pocos recursos con conjuntos de datos pequeños. Los modelos de redes neuronales recurrentes (Bi-LSTM y BI-LSTM-ELMo) presentan una mayor robustez para reconocer entidades en conjuntos de entrenamiento grandes, pues tienen un mecanismo que funciona tanto hacia adelante como hacia atrás, lo que permite incrementar el reconocimiento de entidades. Una clara ventaja de este tipo de modelos, es la de poder proporcionar un conjunto de embeddigs de palabras previamente entrenadas como características para el entrenamiento, proporcionar un conjunto de características propio o asignar de forma aleatoria una capa con pesos para las características previo al entrenamiento.

Las desventaja más evidente del modelo CRF es la complejidad computacional para procesar y analizar grandes conjuntos de datos. Este modelo tiene problemas para reconocer palabras que no fueron vistas en el conjunto de entrenamiento. Para los modelos basados en redes neuronales recurrentes, las desventajas recaen en el alto costo computacional, esto se debe en gran medida a la cantidad de células (el doble, al ser bidireccional) definidas en la red neuronal. Otra desventaja es la cantidad de hiper-parámetros que tienen que ser ajustados correctamente,

para evitar el sobre ajuste (overfitting) o desajuste (underfitting) de la red. Otra desventaja esta marcada por el tiempo computacional (numero de épocas) requerido para alcanzar un buenos resultados.

6.3 Extracción de relaciones

Para llevar a cabo esta tarea se empleo el modelo CRF para reconocer entidades nombradas en un corpus de 32,147 documentos. En cada uno de los documentos se identificaron y extrajeron las oraciones. Posteriormente, se seleccionaron las oraciones con la menos dos entidades nombradas, en caso contrario se descartaron las oraciones.

La evaluación se realizó de forma manual sobre los nueve casos analizados. Se desarrollaron dos evaluaciones por experimento, en la primera evaluación únicamente se contempla que la *relación* sea correcta, y en la segunda se contempla a la *tripleta completa*, es decir, la <*entidad1*, *relación*, *entidad2*> deben de ser correctas. En el experimento1, la mejor evaluación sobre las relaciones fue para el método que extrae relaciones de tipo *"acrónimo"* con 99.5% y con un 96.5% para el método de *"puestos de trabajo"* en las evaluaciones sobre la tripleta completa. En general la evaluación sobre únicamente las relaciones es de 78.43% y de 70.52% para las evaluaciones sobre la tripleta completa. En el Experimento 2, el porcentaje promedio alcanzado al evaluar solo a la relación es de 79.3% y se obtuvo un resultado promedio de 71.31% al evaluar la tripleta completa.

La ventaja de emplear este método para identificar y extraer relaciones de dos entidades nombradas, consiste en los árboles (grafos) de dependencia en sí, representando la oración en estos y permitiendo emplear los algoritmos 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.

Las 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 extraer relaciones con entidades nombradas más frecuentes.

6.4 Base de hechos y reglas

Para construir una base de conocimiento, previamente se realiza un pre-procesamiento de los datos, que consiste en transformar una relación (tripleta) a hecho. Las clases de las entidades encierran a la entidad tal y como se extrajo, el hecho se define en forma de tripleta. Para construir la base de hechos se emplearon los conjuntos de datos de los experimentos usados en la extracción de relaciones. La primer base de conocimiento (Base1) cuenta con 1,151 hechos, la segunda base de hechos (Base2) contiene 696. Además, para los experimentos se empleó la totalidad del corpus de relaciones (sin evaluar), generando una tercera base (Base3) con 86,871 hechos.

Las ventajas de definir reglas lógicas de forma genérica, proporcionan como resultado un conjunto de hechos que no se pretendían obtener de forma específica, esto permite observar a los hechos (tripletas) obtenidos e identificar si se trata de *"conocimiento nuevo"*. La representación de los hechos es similar a su estructura original (tripleta). La definición de la reglas se realizó con pocas líneas de código. Analizar un conjunto de hechos obtenido en el sistema web toma poco tiempo, empleando el lenguaje SQL para obtener la información de la base de datos y desplegar oraciones para su análisis.

Una clara desventaja se encuentra en los resultados obtenidos, presentan hechos *"repetidos"*, debido al backtracking. Se tiene que realizar un filtrado de hechos para observar resultados *"diferentes"*. Otra desventaja es el tiempo que tarda el interprete en proporcionar los resultados, sobre todo cuando la regla involucra una gran cantidad de hechos a verificar, en algunos casos se tiene que detener el proceso de forma manual.

6.5 Grafo de conocimiento

La construcción del grafo consistió en aplicar los lineamientos establecidos por la W3C. El grafo de conocimiento se desarrollo en su forma básica: contempla el sujeto (entidad1), predicado (relación entre las entidades) y el objeto (entidad2). Se usaron esquemas definidos por DBPedia, FOAF y Squema. El grafo se construyó usando RDF en formato XML.

Una ventaja en la construcción del grafo de conocimiento, fue haber conservado la estructura de tripleta en las dos etapas previas. Con el apoyo de bibliotecas de Python la construcción del grafo fue más eficiente. El grafo de conocimiento estructura la información, para poder ser verificada y consultada posteriormente. Se puede validar la sintaxis de los grafos de conocimiento, con *"validadores"* externos.

Existen algunas desventajas en el grafo de conocimiento, se tienen que definir esquemas para datos propios, sobre todo cuando presentan inconsistencias como son las entidades de tiempo. La *"verificación"* de los hechos es sensible a mayúsculas y minúsculas, y las bases de conocimiento existentes no contemplan la mayoría de hechos obtenidos.

6.6 Trabajo a futuro

A continuación se presentan las secciones en las que pretende en un futuro mejorar y/o corregir sobre lo presentado en este trabajo.

6.6.1 Reconocimiento de Entidades Nombradas

Revisar el estado del arte para obtener las pautas establecidas para el etiquetado de entidades nombradas, en idioma español y de ser posible sobre el dominio de noticias, o bien tomar las pautas de otros dominios para ajustarlas al dominio en cuestión. Realizar el etiquetado de forma manual con más personas calificadas, y evaluar la confiabilidad de concordancia del etiquetado con una medida estadística. Además, emplear modelos de aprendizaje profundo como los *transformers* para el reconocimiento de entidades.

6.6.2 Extracción de Relaciones

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.

6.6.3 Base de hechos y Grafo de conocimiento

Para mejorar la calidad de hechos (tripletas) extraídos, se pretende aplicar un conjunto de reglas en la inferencia para filtrar hechos relevantes. Las reglas tendrán el propósito de preservar el significado y evitar la incoherencia en los hechos extraídos.

En este trabajo se construyeron grafos de conocimiento en base a los resultados obtenidos en la inferencia lógica. Por otro lado, lo que se pretende a futuro es crear el grafo de conocimiento en primer instancia, y emplear *razonadores* sobre el árbol completo para descubrir conocimiento nuevo, usando el mismo conjunto de reglas definidas y/o definiendo nuevas, para finalmente realizar una comparación sobre ambos enfoques.

Referencias

Alicante, A., Corazza, A., Isgrò, F., and Silvestri, S. (2016). Unsupervised entity and relation extraction from clinical records in italian. *Computers in biology and medicine*, 72:263–275.

Asahara, M. and Matsumoto, Y. (2003). Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1 of *NAACL '03*, pages 8–15.

Barzegar, S., Davis, B., Handschuh, S., and Freitas, A. (2018). Classification of composite semantic relations by a distributional-relational model. *Data & Knowledge Engineering*, 117:319 – 335.

Bast, H. and Haussmann, E. (2014). More informative open information extraction via simple inference. In *European Conference on Information Retrieval*, pages 585–590. Springer.

Ben Abacha, A. and Zweigenbaum, P. (2011). Automatic extraction of semantic relations between medical entities: a rule based approach. *Journal of Biomedical Semantics*, 2(5):S4.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: A high-performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, pages 194–201, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bitter, C., Elizondo, D. A., and Yang, Y. (2010). Natural language processing: a prolog perspective. *Artificial Intelligence Review*, 33(1-2):151–173.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008a). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008b). Freebase: a collaboratively created graph database for structuring human knowledge. In *In SIGMOD Conference*, pages 1247–1250.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

Borzì, V., Faro, S., and Pavone, A. (2014). Automatic extraction of semantic relations by using web statistical information. In Hernandez, N., Jäschke, R., and Croitoru, M., editors, *Graph-Based Representation and Reasoning*, pages 174–187, Cham. Springer International Publishing.

Bărbulescu, M., Grigoriu, R.-O., Halcu, I., Neculoiu, G., Săndulescu, V. C., Marinescu, M., and Marinescu, V. (2013). Integrating of structured, semi-structured and unstructured data in natural and build environmental engineering. In *2013 11th RoEduNet International Conference*, pages 1–4.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E., and Mitchell, T. (2010). Toward an architecture for never-ending language learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1).

Carreras, X., Màrquez, L., and Padró, L. (2002). Named entity extraction using adaboost. In *Proceedings of the 6th Conference on Natural Language Learning*, volume 20 of *COLING-02*, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.

Che, W., Liu, Y., Wang, Y., Zheng, B., and Liu, T. (2018). Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Chen, D. Y. and Wang, D. Z. (2013). Web-scale knowledge inference using markov logic networks. In *ICML workshop on Structured Learning: Inferring Graphs from Structured and Unstructured Inputs*, pages 106–110. Association for Computational Linguistics.

Chesney, S., Jacquet, G., Steinberger, R., and Piskorski, J. (2017). Multi-word entity classification in a highly multilingual environment. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 11–20.

Colhon, M. and Cristea, Danand Gîfu, D. (2016). Discovering semantic relations within nominals. In TrandabăŢ, D. and Gîfu, D., editors, *Linguistic Linked Open Data*, pages 85–100, Cham. Springer International Publishing.

Cooper, P. (2017). Data, information, knowledge and wisdom. *Anaesthesia & Intensive Care Medicine*, 18(1):55–56.

Dahl, V. and García, A. J. (2018). *Programación Lógica*, volume 2. Triangle.

de Marneffe, M.-C. and Manning, C. D. (2008). The stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545.

Feldman, R. and Rosenfeld, B. (2006). Boosting unsupervised relation extraction by using ner. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 473–481.

Galicia-Haro, S. N. and Gelbukh, A. (2014). Extraction of semantic relations from opinion reviews in spanish. In Gelbukh, A., Espinoza, F. C., and Galicia-Haro, S. N., editors, *Human-Inspired Computing and Its Applications*, pages 175–190, Cham. Springer International Publishing.

Gamallo, P., Garcia, M., and Fernández-Lanza, S. (2012). Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*, pages 10–18.

Gao, J., Li, X., Xu, Y. E., Sisman, B., Dong, X. L., and Yang, J. (2019). Efficient knowledge graph accuracy evaluation. *Proceedings of the VLDB Endowment*, 12(11):1679–1691.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., and Yuret, D. (2007). Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 13–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

Graves, A., Jaitly, N., and Mohamed, A. (2013). Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602 – 610. IJCNN 2005.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hasegawa, T., Sekine, S., and Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 415. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991.

Jiang, S., Lowd, D., and Dou, D. (2012). Learning to refine an automatically extracted knowledge base using markov logic. In *2012 IEEE 12th International Conference on Data Mining*, pages 912–917.

Jiang, X., Wang, Q., Li, P., and Wang, B. (2016). Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480, Osaka, Japan. The COLING 2016 Organizing Committee.

Kendal, S. and Creen, M. (2007). *An Introduction to Knowledge Engineering*. Springer London, London.

Kudo, T. and Matsumoto, Y. (2001). Chunking with support vector machines. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, NAACL '01, pages 1–8.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. (2015). Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Liu, K., Hu, Q., Liu, J., and Xing, C. (2017). Named entity recognition in chinese electronic medical records based on crf. In *2017 14th Web Information Systems and Applications Conference (WISA)*, pages 105–110. IEEE.

Liu, Y. A. (2018). *Logic Programming Applications: What Are the Abstractions and Implementations?*, page 519–548. Association for Computing Machinery and Morgan & Claypool.

Ma, X. and Hovy, E. H. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354.

McCallum, A., Freitag, D., and Pereira, F. C. N. (2000). Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, volume 4 of *CONLL '03*, pages 188–191, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *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*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Mirrezaei, S. I., Martins, B., and Cruz, I. F. (2015). The triplex approach for recognizing semantic relations from noun phrases, appositions, and adjectives. In Gandon, F., Guéret, C., Villata, S., Breslin, J., Faron-Zucker, C., and Zimmermann, A., editors, *The Semantic Web: ESWC 2015 Satellite Events*, pages 230–243, Cham. Springer International Publishing.

Mozharova, V. A. and Loukachevitch, N. V. (2017). Combining knowledge and crf-based approach to named entity recognition in russian. In Ignatov, D. I., Khachay, M. Y., Labunets, V. G., Loukachevitch, N., Nikolenko, S. I., Alexander, P., Savchenko, A. V., and Vorontsov, K., editors, *Analysis of Images, Social Networks and Texts*, pages 185–195. Springer International Publishing.

Özgür, A., Özgür, L., and Güngör, T. (2005). Text categorization with class-based and corpus-based keyword selection. In *Computer and Information Sciences - ISCIS 2005*, pages 606–615, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pan, J. Z., Matentzoglu, N., Jay, C., Vigo, M., and Zhao, Y. (2017). *Understanding Author Intentions: Test Driven Knowledge Graph Construction*, pages 1–26. Springer International Publishing, Cham.

Paulheim, H. (2017a). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.

Paulheim, H. (2017b). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.

Pereira, F. C. and Shieber, S. M. (2002). *Prolog and natural-language analysis*. Microtome Publishing.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL-HLT*.

Pujara, J., Miao, H., Getoor, L., and Cohen, W. (2013). Knowledge graph identification. In Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J. X., Aroyo, L., Noy, N., Welty, C., and Janowicz, K., editors, *The Semantic Web – ISWC 2013*, pages 542–557, Berlin, Heidelberg. Springer Berlin Heidelberg.

Punuru, J. and Chen, J. (2012). Learning non-taxonomical semantic relations from domain texts. *Journal of Intelligent Information Systems*, 38(1):191–207.

Qin, P., XU, W., and Wang, W. Y. (2018). Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2137–2147, Melbourne, Australia. Association for Computational Linguistics.

Ramshaw, L. A. and Marcus, M. P. (1995). *Text Chunking Using Transformation-Based Learning*, pages 157–176. Springer Netherlands, Dordrecht.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In Balcázar, J. L., Bonchi, F., Gionis, A., and Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., Soroa, A., Ploeger, T., and Bogaard, T. (2016). Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37:132–151.

Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., and Chute, C. G. (2010). Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Schoenmackers, S., Davis, J., Etzioni, O., and Weld, D. (2010). Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing*, pages 1088–1098.

Schoenmackers, S., Etzioni, O., and Weld, D. S. (2008). Scaling textual inference to the web. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 79–88.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sena, C. F. L., Glauber, R., and Claro, D. B. (2017). Inference approach to enhance a portuguese open information extraction. In *International Conference on Enterprise Information Systems*, volume 2, pages 442–451. ScitePress.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 134–141.

Shen, H. and Sarkar, A. (2005). Voting between multiple data representations for text chunking. In Kégl, B. and Lapalme, G., editors, *Advances in Artificial Intelligence*, pages 389–400, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. In *Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania*, pages 239–243. Linköping University Electronic Press.

Singh, S. and Karwayun, R. (2010). A comparative study of inference engines. In *2010 Seventh International Conference on Information Technology: New Generations*, pages 53–57. IEEE.

Singhal, A. (2012a). Introducing the knowledge graph: things, not strings. https://googleblog.blogspot.mx/2012/05/introducing-knowledge-graph-things-not.html. Last checked on May 03, 2018.

Singhal, A. (2012b). Introducing the knowledge graph: things, not strings. https://googleblog.blogspot.mx/2012/05/introducing-knowledge-graph-things-not.html. Last checked on May 03, 2018.

Sutton, C. and McCallum, A. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Ta, C. D. and Thi, T. P. (2016). Automatic extraction of semantic relations from text documents. In Dang, T. K., Wagner, R., Küng, J., Thoai, N., Takizawa, M., and Neuhold, E., editors, *Future Data and Security Engineering*, pages 344–351, Cham. Springer International Publishing.

Todorović, B. T., Rančić, S. R., and Mulalić, E. H. (2011). *Context Hidden Markov Model for Named Entity Recognition*, pages 447–460. Springer New York, New York, NY.

Tulkens, S., Šuster, S., and Daelemans, W. (2019). Unsupervised concept extraction from clinical text through semantic composition. *Journal of biomedical informatics*, 91:103120.

Uchimoto, K., Ma, Q., Murata, M., Ozaku, H., and Isahara, H. (2000). Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 326–335.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wang, G., Zhang, W., Wang, R., Zhou, Y., Chen, X., Zhang, W., Zhu, H., and Chen, H. (2018). Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2246–2255, Brussels, Belgium. Association for Computational Linguistics.

Wu, S. and He, Y. (2019). Enriching pre-trained language model with entity information for relation classification. *arXiv preprint arXiv:1905.08284*.

Yan, J., Wang, C., Cheng, W., Gao, M., and Zhou, A. (2018a). A retrospective of knowledge graphs. *Frontiers of Computer Science*, 12(1):55–74.

Yan, J., Wang, C., Cheng, W., Gao, M., and Zhou, A. (2018b). A retrospective of knowledge graphs. *Frontiers of Computer Science*, 12(1):55–74.

Yan, J., Wang, C., Cheng, W., Gao, M., and Zhou, A. (2018c). A retrospective of knowledge graphs. *Frontiers of Computer Science*, 12(1):55–74.

Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., and Ishizuka, M. (2009). Unsupervised relation extraction by mining wikipedia texts using information from the web. In *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: Volume 2-Volume 2*, pages 1021–1029. Association for Computational Linguistics.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.