Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Extracción de marcos en el dominio de los eventos legales

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial

AUTOR: Aida Sánchez Romero
TUTOR/ES: Víctor Rodríguez Doncel y Óscar Corcho

2019

AGRADECIMIENTOS

Quería agradecer, ante todo, el esfuerzo y apoyo constante de mis padres, mi hermana y mi pequeño diablillo. Sin ellos no podría haber finalizado este trabajo y, por lo tanto, esta etapa de mi vida. Sois lo más importante que tengo. Gracias.

A mis amigos, por sus tardes y noches en las que el tiempo se detenía para darnos una oportunidad de desahogo en las que no existían el trabajo ni los estudios, solo nosotros. Gracias.

A mis jefes y compañeros, por su total confianza y libertad para acabar este proyecto. Gracias.

A Víctor y María, por su ayuda, tiempo y dedicación. Sin vosotros no podría haber presentado este trabajo a tiempo. Gracias.

Por todo y mucho más.

GRACIAS.

RESUMEN

SUMMARY

Índice

Índice de figuras

Índice de cuadros

1. INTRODUCCIÓN

Dentro de la **Inteligencia Artificial** se denomina **Extracción de Información** a un campo de trabajo perteneciente al ámbito del **Procesamiento del Lenguaje Natural**, la **Lingüística Computacional** y la **Minería de textos**, que busca localizar cierta información en texto libre en un dominio determinado, ignorando otra información irrelevante. A saber: *quién hizo qué a quién, dónde, cómo y cuándo.* La extracción de información se centra en derivar información estructurada (produciendo repositorios tales como una tabla relacional o un archivo XML) a partir de un texto no estructurado (o repositorio de documentos dado). Esto, aun cuando el dominio está perfectamente acotado, no es una tarea simple debido a la complejidad y ambigüedad del lenguaje natural.

Es importante recalcar que la extracción de información no busca comprender en su totalidad un texto, desentrañando todas las posibles interpretaciones y relaciones gramaticales. Algo que, por otro lado, es una tarea imposible a día de hoy desde un punto de vista tecnológico.

A menudo esta tarea es confundida con la **Recuperación de Información**, que consiste en encontrar material (generalmente documentos) de naturaleza no estructurada (generalmente texto) que satisfaga una necesidad de información dentro de grandes colecciones (generalmente almacenados en computadoras). Dicho de otra forma, consiste en encontrar un subconjunto de documentos, a partir de una colección más grande, que contengan información relevante dada una consulta específica basándose en una búsqueda por palabra clave (o *keyword*, en inglés) que se podría ampliar mediante la utilización de **tesauros** [1]. La lista ordenada de documentos no proporciona ninguna información detallada sobre el contenido de los mismos ya que no se utiliza ningún conocimiento semántico. Por el contrario, el objetivo de la extracción de información no es clasificar o seleccionar documentos, sino extraer de los mismos hechos sobre tipos predeterminados de eventos, entidades o relaciones, con el fin de construir representaciones más significativas, que se pueden utilizar para poblar bases de datos que proporcionen información estructurada, a fin de buscar patrones más complejos (resúmenes, tendencias, etc.) en **corpus** [2].

[1] Un **tesauro** (o *thesaurus*, proveniente del latín) es una lista de palabras o términos que se interrelacionan entre sí a través de (1) relaciones jerárquicas (estructuras Todo/Parte), (2) relaciones de equivalencia (sinonimia, homonimia, antonimia y polisemia) y (3) relaciones asociativas (reducción polijerarquía). De una forma más específica, podemos decir que un tesauro es un intermediario entre el *lenguaje natural* (el usado en los documentos) y el *lenguaje controlado* (el empleado por los especialistas de un determinado campo del saber).

[2] Un **corpus** es un conjunto amplio y estructurado de ejemplos reales de uso de la lengua. Los más comunes son los textos, aunque también puede tratarse de muestras orales generalmente transcritas.

La información que se pretende conseguir mediante las técnicas de extracción de información está previamente especificada en estructuras definidas por los usuarios, denominadas plantillas[3] (o *templates*, en inglés) u objetos, cada una de ellas con un número de espacios (o atributos), que son los que deben instanciarse o completarse por el sistema, conforme procese el texto.

1.1. Motivación

Dentro de la extracción de información existen cuatro tareas principales que serán descritas en profundidad en el Estado del Arte, a saber, *reconocimiento de entidades nombradas, resolución de correferencia, extracción de relaciones* y *extracción de eventos*. Uno de los mayores problemas reside en esta última tarea debido a la creciente cantidad de datos y el número creciente de fuentes de datos digitales. Es por ello que el uso de información extraída en los procesos de toma de decisiones se vuelve cada vez más urgente y difícil.

Un problema omnipresente es el hecho de que la mayoría de los datos inicialmente no están estructurados, es decir, el formato de los datos implica poco su significado y se describe usando un lenguaje natural y comprensible para el ser humano, lo que hace que los datos estén limitados en el grado en que se pueden interpretar a máquina. Este problema frustra la automatización de, por ejemplo, los procesos de recuperación de información vital y extracción de información, que se utilizan para la toma de decisiones cuando involucran grandes cantidades de datos; haciendo de esta tarea la más compleja de todas y en la que se centra nuestro trabajo.

1.2. Objetivo

El objetivo principal de este trabajo es explorar el software existente capaz de anotar marcos (o *frames*, en inglés) de cualquier tipo de representación de eventos, y expandir cualquiera de ellos con el fin de ofrecer la posibilidad de trabajar con frames propios de eventos legales.

El segundo y último objetivo, pero no por ello menos importante, sería el de desarrollar un software de documentos legales anotados con eventos para poder probar el sistema, así como un sistema de visualización de dicho software.

[3] Una **plantilla** o *template* es una estructura de tipo marco con slots que representan la información básica del evento. Esta información es del tipo participantes del evento, resultado obtenido, hora y ubicación, entre otras.

1.3. Terminología

En esta subsección se presentan los términos más importantes que se repiten con asiduidad a lo largo de la memoria. Es por ello que se ha decidido reunirlos en una tabla (Ver 1) en la que figuran su nombre tanto en español como en inglés, así como las siglas (tomando como origen el inglés) por las que van a ser identificados de aquí en adelante. Dichos términos serán explicados en profundidad en su (sub)sección correspondiente.

Términos en español	**Términos en inglés**	**Siglas**
Inteligencia Artificial	Artificial Intelligence	AI
Extracción de Información	Information Extraction	IE
Procesamiento del Lenguaje Natural	Natural Language Processing	NLP
Lingüística Computacional	Computational Linguistic	CL
Minería de Textos	Text Mining	TM
Recuperación de Información	Information Retrieval	IR
Reconocimiento de Entidades Nombradas	Named Entities Recognition	NER
Resolución de Correferencia	Coreference Resolution	CR
Extracción de Relaciones	Relationships Extraction	RE
Extracción de Eventos	Events Extraction	EE
Parte Del Discurso	Part-Of-Speech	POS
Etiquetación Automática de Roles Semánticos	Automatic Semantic Roles Labeling	ASRL
Redes Neuronales Recurrentes Segmentacionales	Segmentational Recurrent Neural Network	SegRNN o SRNN
Máquina de Soporte Vectorial	Support Vector Machine	SVM

Tab. 1: Terminología usada en el documento junto con sus siglas

1.4. Estructura

El trabajo presentado en esta memoria se divide en las siguiente partes (o secciones) principales:

- La **sección 2** contiene el Estado del Arte. Esta sección se divide en varias subsecciones principales. En la primera figuran las tareas principales de la extracción de la información, en la segunda se hace referencia a los posibles enfoques para la extracción de eventos; y en la tercera y cuarta subsección figuran los principios básicos de PropBank y FrameNet respectivamente, junto con todo el software encontrado basado en tales principios.

- La **sección 3** contiene el experimento realizado y se divide en varias subsecciones: implementación, código, corpus de estudio, evaluación y resultados obtenidos.
- Finalmente, la **sección 4** contiene las conclusiones del trabajo, así como posibles mejoras a tener en cuenta para cualquier desarrollo o profundización futura del mismo.

2. Estado del Arte

En esta sección se presenta el estado del arte – desde un punto teórico – de las diferentes tareas de la IE, los métodos posibles para la EE y las diferentes herramientas software existentes para la IE en el dominio de los eventos legales. Para ello ha sido necesaria la división de la misma en las siguientes subsecciones:

1. **Principales tareas de la extracción de la información**: reconocimiento de entidades nombradas, extracción de correferencias, extracción de relaciones y extracción de eventos.
2. **Métodos para la extracción de eventos**: basados en el conocimiento, basados en datos e híbridos.
3. **PropBank**: qué es y cuáles son sus principales técnicas existentes.
4. **FrameNet**: qué es y cuáles son sus principales técnicas existentes.

2.1. Principales tareas de la extracción de la información

2.1.1. Reconocimiento de entidades nombradas

Esta tarea aborda el problema de la identificación (detección) y clasificación de tipos predefinidos de **entidades nombradas** tales como organizaciones, personas, nombres de lugares, expresiones temporales, expresiones numéricas y de divisas, etc. La tarea de NER puede, además, incluir la extracción de información descriptiva a partir un texto en relación a las entidades detectadas, rellenando plantillas simples. Por ejemplo, en el caso de las personas puede incluir extraer el puesto de trabajo, la nacionalidad, el género y otras características de la misma. Se ha de destacar que el reconocimiento de entidades nombradas también implica el proceso de **lematización** de dichas entidades, que es clave en lenguas flexivas.

La **lematización** es un proceso lingüístico que consiste en, dada una forma flexionada, es decir, en plural, en femenino, conjugada, etc., hallar el **lema** correspondiente, siendo éste la forma que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra. Por ejemplo el lema de la palabra *perras* es *perro*, mientras que el lema de la expresión *he ido a correr* es *correr*.

2.1.2. Reconocimiento de correferencia

Esta tarea aborda la identificación de diferentes menciones de la misma entidad en el texto. Las menciones de las entidades pueden ser de varios tipos, a saber:

- Nombradas: referidas por su nombre. Por ejemplo *Barack Obama visitará la India en enero de 2015. Mr Obama se reunirá con los principales líderes del partido del Congreso de oposición.*

- Pronominales: referidas por su pronombre. Por ejemplo *Barack Obama visitará la India en enero de 2015. Él se reunirá con los principales líderes del partido del Congreso de oposición.*
- Nominales: referidas por un sintagma nominal. Por ejemplo *Barack Obama visitará la India en enero de 2015. El Presidente de los EEUU se reunirá con los principales líderes del partido del Congreso de oposición.*
- Implícitas: en el caso de que no exista la anáfora. Por ejemplo *Barack Obama visitará la India en enero de 2015. Se reunirá con los principales líderes del partido del Congreso de oposición.*

Por lo que podemos decir que esta tarea es, en cierto sentido, el *hipervínculo del lenguaje natural*. Por un lado, presta un elemento de estilo y cohesión al escritor humano, mientras que por otro, agrega otra dimensión de oscuridad a la comprensión mecánica del lenguaje. Luego, no se trata de una tarea trivial.

En muchas ocasiones esta tarea se ha confundido con la **Resolución de la Anáfora** de manera errónea ya que la correferencia es una relación de equivalencia, mientras que la anáfora no es ni reflexiva, ni simétrica (ni transitiva). A modo de ejemplo, podemos decir que dos frases nominales se relacionan entre sí mediante correferencia si ambas se resuelven en un único referente (sin ambigüedad). Sin embargo, se dice que una frase nominal A es el antecedente anafórico de una frase nominal B si y solo si A es necesaria para la interpretación de B.

2.1.3. Extracción de relaciones

Esta tarea aborda la detección y clasificación de relaciones predefinidas entre dos o más entidades nombradas (NE) en un texto. Los tipos de relaciones son ilimitados y están predefinidos y fijados como parte de la especificación de la tarea. Por ejemplo *employed_at(Aida, Crisser), founded_by(Apple, Steve Jobs).*

Esta tarea se enfrenta a muchos desafíos debido a que hay una gran variedad de relaciones posibles que varían de un dominio a otro, las relaciones no tienen por qué ser binarias, las técnicas de aprendizaje automático supervisadas se enfrentan a una mayor dificultad (normalmente no se dispone de suficientes datos de entrenamiento) y la ambigüedad inherente juega un gran papel en cuanto a lo que una relación "*significa*"(a menudo se refleja en los altos desacuerdos entre anotadores). Finalmente, y dado que la expresión de una relación depende en gran medida del lenguaje, hace que esta tarea sea dependiente del lenguaje.

2.1.4. Extracción de eventos

Esta tarea aborda la identificación de eventos en texto libre y la derivación de información estructurada sobre los mismos, buscando identificar *quién hizo qué a quién, cuándo, dónde, a través de qué métodos y por qué*. La labor de extracción de eventos implica la extracción de varias entidades y relaciones entre ellas.

Un ejemplo podría ser una *adquisición*. Si consideramos la representación <**Company**> <**Buy**><**Company**>, las palabras identificadas en el texto que se refieren a *empresas* están vinculadas al concepto <**Company**>, y (las conjugaciones de) los *verbos* que tienen el *significado* de *adquisición* están asociados a <**Buy**>. Las representaciones de este evento se pueden extraer de los encabezados de los mensajes de noticias como *"Google adquiere Picnik", "Lala comprada por Apple"* o *"Skype vendido a Microsoft"*.

Para poder realizar esta última tarea existen tres enfoques diferentes. Estos enfoques se detallan a continuación.

2.2. Métodos existentes para la Extracción de Eventos

2.2.1. Métodos basados en datos

Este tipo de métodos tienen como objetivo convertir los datos en conocimiento mediante el uso de estadísticas, aprendizaje automático, álgebra lineal, etc. Los inconvenientes principales de estos métodos son que no tratan el significado explícitamente, es decir, descubren relaciones en cuerpos sin considerar la semántica y que requieren una gran cantidad de datos para obtener resultados estadísticamente significativos. Por otro lado, al no basarse en el conocimiento, no se requieren recursos lingüísticos, ni conocimiento experto (dominio).

2.2.2. Métodos basados en el conocimiento

Este tipo de métodos extraen el conocimiento a través de la representación y la explotación del conocimiento experto, generalmente mediante enfoques basados en patrones. Estos patrones expresan reglas basadas intrínsecamente en el conocimiento lingüístico y lexicográfico; así como en el conocimiento humano existente con respecto al contenido del texto que se va a procesar. Esto alivia los problemas de los métodos basados en datos con respecto al significado del texto. La información se extrae de los corpus mediante el uso de patrones lingüísticos predefinidos o descubiertos, que pueden ser tanto patrones *léxico-sintácticos* (que combinan representaciones léxicas e información sintáctica con expresiones regulares) como patrones *léxico-semánticos* (hacen uso de información semántica). La información semántica se suele agregar mediante *gazetteers*, que utilizan el significado lingüístico del texto, o mediante *ontologías*.

Otra de las ventajas es que necesitan menos datos que los métodos anteriores, dado que no necesitan un entrenamiento previo para entrenar su aprendizaje. Además, es posible definir expresiones potentes utilizando elementos léxicos, sintácticos y semánticos, y los resultados son fácilmente interpretables y rastreables. Los patrones son útiles cuando se necesita extraer información muy específica.

Sin embargo, la principal desventaja es que para poder definir patrones que recuperen la información correcta y deseada, se requieren conocimientos léxicos y posiblemente también conocimientos previos de dominio. Otras posibles desventajas se relacionan con la definición y el mantenimiento de patrones, ya que la adquisición de conocimientos se hace más difícil (por ejemplo, en costos y consistencia) cuando los patrones deben ampliarse para cubrir más situaciones debido al hecho de que los patrones son, por lo general, hechos a mano.

2.2.3. Métodos híbridos

Este tipo de métodos combinan el conocimiento y los métodos basados en datos. Como ambos enfoques tienen sus desventajas, la combinación de los dos métodos podría dar los mejores resultados. En general, un enfoque puede ser visto principalmente como datos o impulsado por el conocimiento. Sin embargo, hay un número cada vez mayor de investigadores que combinan ambos enfoques por igual, y que de hecho emplean enfoques híbridos.

Dentro de los *métodos basados en datos* se encuentra **PropBank**; mientras que dentro de los *métodos basados en el conocimiento* se encuentra **FrameNet**. Tanto PropBank como FrameNet son recursos electrónicos basados en marcos semánticos. Estos son los recursos en los que se ha centrado este trabajo para la extracción de los eventos en el dominio *legal*.

2.3. PropBank

2.3.1. ¿Qué es PropBank?

PropBank es un enfoque práctico de la representación semántica en el que se agrega una capa de información del tipo *predicado-argumento*, o etiquetas[4] de roles semánticos a las estructuras sintácticas de **Penn TreeBank**. Esta anotación sintáctica identifica tanto los sujetos como los objetos del verbo, proporcionando etiquetas con las funciones semánticas del tipo *temporal/locativo* sin ser capaz de distinguir los roles desempeñados por el sujeto u objeto gramatical de un verbo. Por lo que se podría decir que está *orientado al verbo* y que no anota eventos o estados de cosas descritas usando sustantivos.

[4] En principio hay un total de 12 etiquetas diferentes disponibles para cada uno de los constituyentes (a saber, DIR, LOC, MNR, TMP, EXT, REC, PRD, PRP, DIS, ADV, MOD y NEG).

Además, estas anotaciones producidas por **PropBank** son a *nivel sintáctico*, ya que tan solo anotan el sentido literal del objetivo, prefiriendo metas pequeñas, incrementales y fáciles de alcanzar. Este proceso de anotación se realiza a partir de un etiquetador automático basado en reglas y corregido a continuación de manera manual. Debido a esto, una ventaja considerable con respecto a **FrameNet** es que necesita una menor comprensión del contexto en el que se encuentra, lo que hace que resulte una tarea menos compleja. Aunque, por contra, los resultados obtenidos contienen menos información.

En cuanto a la creación del conjunto de marcos en **PropBank**, a diferencia de **FrameNet** (que no considera las diferencias sintácticas), éste necesita que coincidan tanto el número de posibles roles semánticos como los significados de los usos para agruparlos en un mismo conjunto. Además, **PropBank** no diferencia entre las *oraciones causativas*[5] e *incoativas*[6].

Finalmente, en cuanto a la metodología usada, se puede destacar que PropBank emplea *backoff* basada en una red que combina las características del modelo. Esta solución restringe el tamaño de los conjuntos de características ya que resulta difícil añadir nuevas características al problema. Las características estándar del problema son: predicado, camino mínimo desde el predicado hasta el constituyente a ser clasificado, tipo de frase (NP, PP, etc.), posición (antes/después del predicado), voz (activa/pasiva), palabra clave y sub-categorización.

Las tareas principales que desempeña **PropBank** en principio son identificar los diferentes argumentos o constituyentes de cada predicado y asignar un rol a cada uno de ellos. En Xue Palmer (2004) se ha realizado un estudio en el que indican la utilidad de cada una de las características anteriores con respecto a cada una de las dos tareas mencionadas. En cuanto a la tarea de identificar los argumentos las características destacadas son el uso del camino y la palabra clave (junto con su etiqueta POS). Además, del uso del tipo de frase junto con el predicado; y si este último está especificado, la distancia también podría ser útil. En cuanto a la tarea de asignar un rol a cada argumento, se destaca el uso del tipo de frase y la palabra clave (sobre todo si se conoce el predicado). Por otra parte tanto el uso del camino, como el de la sub-categorización y la voz (estos dos últimos debido a que son compartidos por toda la oración, es decir, no son discriminantes en ningún aspecto) no tienen ningún interés

[5] Las **oraciones causativas** son aquellas en las que el sujeto de la oración no realiza la acción, sino que provoca que otro lo haga. Por ejemplo: *"El árbitro hizo repetir el lanzamiento."*

[6] Las **oraciones incoativas** son aquellas en las que hay una acción progresiva o se indica el comienzo de una acción o cosa. Por ejemplo: *"Ha comenzado a llover."* o *"Las niñas se echaron a reír."*

2.3.2. ¿Cuáles son sus principales técnicas existentes?

A continuación se detallan algunas de las técnicas existentes basadas en el principio de *Palmer et al.*:

- En Surdeanu et al. (2003) se describe un sistema de IE independiente del dominio con identificación automática de estructuras del tipo *predicado-argumento*, igual que en PropBank. En este sistema se aplican dos métodos diferentes. El primero de ellos es el estadístico usado en PropBank, cuya tarea es identificar los componentes del árbol de análisis correspondientes a los argumentos de cada predicado codificado en PropBank. El segundo método es nuevo y está basado en el aprendizaje inductivo, cuya tarea es reconocer el rol correspondiente de cada argumento. La ventaja principal del aprendizaje inductivo a través de árboles de decisión es que les permite probar fácilmente grandes conjuntos de características y estudiar el impacto de cada característica en el analizador aumentado que genera estructuras de argumentos de predicado. Por ello, utilizan el algoritmo de aprendizaje del árbol de decisión inductivo C5 (Quinlan (2002)), para implementar tanto el clasificador que identifica los constituyentes de los argumentos como el clasificador que etiqueta los argumentos con sus roles. Para ello añaden dos características principales: reducen a 7 etiquetas principales (a saber, PERSON, ORGANIZATION, LOCATION, PERCENT, MONEY, TIME y DATE) los constituyentes y añaden a la palabra principal su respectivo POS. Este nuevo método obtiene resultados más precisos que PropBank para predicados *no conocidos*.

- En Pradhan et al. (2004) se reemplaza el algoritmo de clasificación estadística con uno que usa SVM[7] y luego se agrega al conjunto de características existente. Este nuevo sistema añade, además de las características mencionadas en Surdeanu et al. (2003), algunas características nuevas: agrupación de verbos (en un total de 64 clases usando el modelo de co-ocurrencia probabilística de Hofmann Puzicha (1998)), ruta parcial, información del sentido verbal, palabra principal de las frases preposicionales (por ejemplo, etiquetar *PP-in* en lugar de *PP*), tener en cuenta la primera y la última palabra de un constituyente junto con su POS correspondiente, concatenación del tipo de constituyente y su posición ordinal del predicado, definir la distancia del árbol constituyente y añadir tanto las características relativas de los constituyentes así como las palabras de referencia temporales. En cuanto a la identificación del argumento, obviamente obtiene peores resultados que si se obtuvieran a mano, pero en cuanto a la clasificación de dichos argumentos funciona significativamente mejor. Al intentar hacer ambas tareas el SVM hace un buen trabajo en ambas etapas.

[7] Las **SVM** son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios ATT. Estos métodos están propiamente relacionados con problemas de *clasificación* y *regresión*. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra.

- En Xue Palmer (2004) proponen un nuevo conjunto de características con el fin de explotar mejor la información que proporciona el árbol analizador. Estas nuevas características son:

  - Añadir marcos semánticos que varían en función de la clasificación del constituyente. Un ejemplo a partir de la oración "*The Supreme Court gave states more leeway to restrict abortion*"(Ver Figura 1), un posible marco de *states* podría ser *np_v_NP_np* (otros posibles marcos podrían ser *np_v_CUR_np* si no se identifica la categoría sintáctica o *np_give_CUR_np* si se lematiza el predicado).

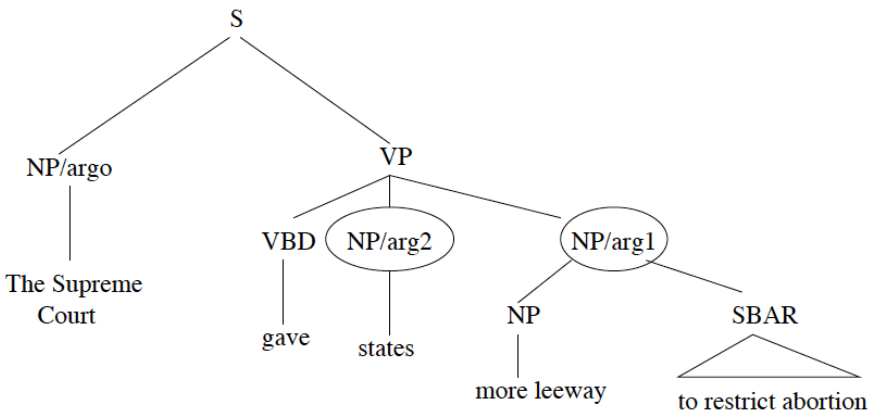

Fig. 1: Oración de ejemplo obtenida de Xue Palmer (2004) donde se pueden apreciar varios NPs (predicados nominales) como pivotes del VP (predicado verbal).

  - Añadir el lema del predicado al tipo de constituyente. Un ejemplo a partir de la oración anterior con respecto a *states* sería *give_NP*.
  - Combinar la palabra clave con el lema del predicado. Un ejemplo a partir de la oración anterior con respecto a *states* sería *give_states*.
  - Combinar la voz y la posición del verbo con respecto al constituyente. Un ejemplo a partir de la oración anterior con respecto a *states* sería *passive_before*.
  - Si el constituyente es un PP, entonces se obvia (deja de pertenecer al constituyente y se añade esta palabra principal a la etiqueta). Esta misma característica ya se tomó en cuenta en Pradham et al. (2004).

- AllenNLP: en Gardern et al. (2018) se lanza una nueva plataforma web de acceso libre[8] basada en la idea de PropBank pero sustituyendo el modelo propuesto por uno basado en BiLSTM (Bidirectional Long Short-Term Memory), que es un tipo especial de RNN. AllenNLP se centra en varias tareas entre

[8] Está disponible de manera online a través del siguiente enlace `https://demo.allennlp.org/semantic-role-labeling`

las que destacan SRL y analizador de constituyentes que divide un texto en constituyentes. Los no terminales en el árbol son tipos de frases y los terminales son las palabras en la oración. Este modelo utiliza incrustaciones ELMo (Peters et al., 2018), que están completamente basadas en caracteres y mejora el rendimiento obtenido con Penn TreeBank hasta el momento.

2.4. FrameNet

2.4.1. ¿Qué es FrameNet?

FrameNet es un recurso electrónico basado en los **marcos semánticos** (o semántica de marcos) creado principalmente por *Charles J. Fillmore* en la Universidad de Berkeley y lanzado en torno al año 1998. Esta semántica está incluida dentro de la **lingüística cognitiva**, ya que no solamente considera los *aspectos formales*, si no que también da cuenta del *lenguaje como facultad inherente* al individuo y, como tal, debe aludir a los aspectos neurolingüísticos, psicolingüísticos, sociolingüísticos y antropolingüísticos que hacen posible el funcionamiento del lenguaje como una herramienta de cognición, representación, comunicación e interacción entre los individuos. Esto quiere decir que la lingüística cognitiva no es una sola teoría del lenguaje, si no un marco flexible que enfatiza en el hecho de que definir una categoría puede implicar describir algunos de sus miembros principales en lugar de dar simplemente una definición abstracta. También subraya que la definición abstracta no tiene que constituir en un conjunto único de características definitorias que pertenezcan única y distintivamente a esa categoría. En resumen, podemos concluir que la **lingüística cognitiva** es "*el estudio del lenguaje natural tratado como un fenómeno mental*".

En cuanto a los **marcos semánticos**, la teoría afirma que las personas entienden el significado de las palabras en gran parte en virtud de los marcos que evocan. Los **marcos** representan fragmentos de historias, que sirven para conectar un grupo de palabras a un conjunto de significados. Es por ello que el estudio de los marcos semánticos intenta definir los marcos y los "participantes/elementosïnvolucrados en cada uno de ellos.

Por supuesto, el proceso de entender una oración en inglés (o en cualquier otro idioma) no solo depende de conocer las palabras y los marcos que evocan, sino también de las construcciones gramaticales que determinan la jerarquía sintáctica de la oración y, a su vez, el orden de las palabras (este concepto se basa en la ***teoría de la gramática de la construcción***, en la que las construcciones no solo definen las relaciones entre los elementos que evocan el marco y los elementos de llenado de roles, sino que también, en muchos casos, tienen un significado propio). El trabajo en el proyecto **FrameNet** siempre ha supuesto la existencia de dicha teoría.

El principal producto de este trabajo (la **base de datos léxica FrameNet**) contiene actualmente más de 13.000 *unidades léxicas* (definidas a continuación), aproximadamente 7.000 de las cuales están completamente anotadas; en más de

1.000 marcos semánticos relacionados jerárquicamente, ejemplificados en más de 200.000 frases anotadas.

Una **unidad léxica (Lexical Unit (LU), en inglés)** es un emparejamiento de una palabra con un significado. Típicamente, cada sentido de una palabra polisémica pertenece a un marco semántico diferente, una estructura conceptual similar a un "*script*"que describe un tipo particular de situación, objeto o evento junto con sus participantes y objetos.

El objetivo del proyecto **FrameNet** era definir los **marcos semánticos**, creando una descripción de cada marco en su conjunto y de cada uno de sus elementos. Dichos marcos consisten en agrupaciones de ideas evocadas por palabras o grupos de palabras que tienen cierta superposición semántica, y dividirlos en grupos para, más tarde, combinarlos en grupos lo suficientemente grandes como para hacer marcos razonables en los que podemos (equivalentemente) llamar a las palabras objetivos, unidades léxicas o elementos que evocan marcos. Al final, se quiere terminar con grupos de palabras de destino en cada marco que tengan un tipo particular de superposición semántica. En el pasado, los criterios para tal agrupación han sido *informales* e *intuitivos*, pero ahora estos criterios son más explícitos. En un sentido práctico, los criterios son de dos tipos:

- Una lista de verificación de características, donde si un criterio de similitud no se cumple, deberíamos poner las palabras en diferentes marcos. Estos criterios son: mismo número y tipo de elementos de marco tanto implícitos como explícitos, las palabras deben denotar la misma parte de la escena y que precisen las mismas relaciones.

- Un principio más difícil de definir para que nuestras agrupaciones sean útiles, especialmente como paráfrasis y como respuestas alternativas a una pregunta. Además, se deben destacar los criterios no utilizados para la división de marcos, a saber, (1) las diferencias gramaticales tales como la formación de la pasiva, la composición de elementos de marco extra-temáticos, las construcciones de tiempo/aspecto y las diferencias POS; (2) los antónimos; y (3) las diferencias de uso tales como deixis, registro, dialecto y evaluación.

Las principales aplicaciones del proyecto **FrameNet** son, entre otras, el reconocimiento de vinculaciones textuales, el parafraseo, el sistema de pregunta-respuesta (ambos resaltados con anterioridad) y la extracción de información, como es el caso en el que se centra este trabajo.

Esta extracción se puede hacer tanto de forma **directa** como por medios de un **ASRL**. Debido a que la etiquetación de forma directa a través de anotaciones FrameNet es muy costosa y lenta, en Gildea Jurafsky (2002) se desarrolla la etiquetación por medios ASRL que produce de forma automática, utilizando técnicas de aprendizaje automático, anotaciones muy similares a las de FrameNet en textos

nuevos nunca antes vistos.

Las tareas principales de un sistema ASRL (o SRL simplemente) son encontrar los constituyentes de las frases relevantes y darles a cada uno de ellos la etiqueta semántica correcta. El primer sistema ASRL consiste en, dada una oración de entrada y un marco de destino, etiquetar los constituyentes (ya sea con roles semánticos abstractos o con específicos del dominio). Este nuevo sistema se basa en clasificadores estadísticos entrenados en aproximadamente 50.000 oraciones anotadas a mano y recuperadas del proyecto **FrameNet**. Cada una de estas oraciones se analizó en un árbol sintáctico del que se extraían varias características léxicas y sintácticas, incluyendo el tipo de frase de cada constituyente, su función gramatical y su posición en la oración. Estas características están derivadas del corpus de **Penn TreeBank** [9].

2.4.2. ¿Cuáles son sus principales técnicas existentes?

A continuación se detallan algunas de las técnicas existentes de ASRL basadas en *Fillmore* y posteriores a la nombrada en la subsección anterior:

- **SEMAFOR**: en Das et al. (2010) nace un nuevo sistema que identifica argumentos de marco semántico utilizando un modelo lineal con características diseñadas a mano basadas en un análisis de dependencia. Este sistema modela instancias nulas incluyendo referencias a argumentos no locales. Sustituye los modelos de Pradham et al. (2004) por dos modelos log-lineales (con un solo conjunto de ponderaciones en cada uno de ellos) para encontrar un análisis semántico de marco completo. En Das et al., 2012 se producen pequeñas variaciones tales como la eliminación de los intervalos de argumentos utilizando heurísticas sintácticas y ?beam search? (o AD3) para decodificar respetando las restricciones; y en Kshirsagar et al. (2015) se extiende de nuevo el modelo mediante el uso de anotaciones ejemplares de FrameNet, características de la guía PropBank y la jerarquía FrameNet. Finalmente cabe destacar que este proyecto está escrito en Java y es de libre acceso [10].

- **Framat**: en Roth Lapata (2015) se desarrolla un nuevo sistema que agrega características basadas en el contexto de la oración y el discurso para mejorar un sistema SRL adaptado para los marcos semánticos utilizando un modelo global con ranking. Para ello definen las siguientes características: (1) a nivel del discurso utilizan directamente el conocimiento del discurso en forma de cadenas de referencias, estas se generalizan mejor que las características léxicas y semánticas tradicionales; (2) a nivel de oraciones modelan las propiedades de una estructura de marco como un todo, las características contextuales proporcionan información adicional necesaria para comprender y asignar

[9] **Penn TreeBank** es un corpus lingüístico donde cada frase ha sido parseada - o anotada - con su estructura sintáctica, representada generalmente como una estructura arbórea, empleando en la mayoría de los casos un etiquetado gramatical.

[10] Código disponible en `https://github.com/Noahs-ARK/semafor-semantic-parser`

roles a este nivel; y (3) léxicas que se pueden calcular utilizando métodos de semántica distributiva y una adaptación para modelar el significado de las palabras específicas del documento.

En 2017 sus creadores extienden este modelo mediante incorporaciones de aprendizaje para los caminos de dependencia entre el predicado y sus argumentos. Para ello crean un nuevo modelo conocido como *PathLSTM* basado en mate-tools que modela relaciones semánticas entre un predicado y sus argumentos mediante el análisis de la ruta de dependencia aplicando una LSTM (un tipo específico de NN). Este modelo considera las rutas lexicalizadas, que se descomponen en secuencias de elementos individuales, es decir, las palabras y las relaciones de dependencia en una ruta. Luego se aplican redes de memoria a corto y largo plazo para encontrar una función de composición recurrente que pueda reconstruir una representación apropiada de la ruta completa a partir de sus partes individuales. Al modelar las rutas de dependencia como secuencias de palabras y dependencias, se aborda de manera implícita el problema de la dispersión de datos. Finalmente cabe destacar que este proyecto está escrito en Java y es de libre acceso[11].

- **OPEN-SESAME**: en Swayamdipta et al. (2017) se desarrolla el que es considerado como el primer analizador semántico libre de sintaxis gracias a su modelo *softmax-margin SegRNN*. Este modelo es una modificación de los modelos *SegRNN*[12] que fomentan la recuperación por encima de la precisión abandonando el filtrado sintáctico y las características sintácticas para la identificación de argumentos de marcos semánticos. Finalmente, a este modelo se le añade información semántica usando (1) un enfoque segmentado que incorpora características de dependencia automática o analizadores de estructura de frase, y (2) un enfoque de andamio sintáctico, descartando de esta forma la necesidad de un analizador sintáctico ya que conserva el beneficio de las características sintácticas sin costo computacional mediante la identificación de los constituyentes sin etiqueta entrenándolo mediante el corpus lingüístico **Penn Treebank**. Finalmente cabe destacar que este proyecto está escrito en Python y es de libre acceso[13].

- **TakeFive**: en Alam et al. (2018) se introduce un nuevo sistema de SRL basado en FrameNet. Este método es un método híbrido y transforma un texto en un grafo de conocimiento orientado a marcos semánticos utilizando *Framester*[14]. Para ello realiza un análisis de dependencia, identifica las palabras

[11] Código disponible en `https://github.com/microth/mateplus`

[12] Las **SegRNN (o SRNN)** combinan dos potentes herramientas de aprendizaje automático (1) el aprendizaje de representación y (2) la predicción estructurada. Además, son una variante de los campos aleatorios condicionales semi-Markov ya que definen una distribución de probabilidad condicional sobre el espacio de salida (segmentación y etiquetado) dada la secuencia de entrada.

[13] Código disponible en `https://github.com/swabhs/open-sesame`

[14] Framester es un grafo de conocimiento RDF que actúa como centro entre varios recursos

que evocan marcos léxicos, localiza los roles y rellenos de cada marco, ejecuta técnicas de coerción y formaliza los resultados en grafos de conocimiento. El algoritmo principal de TakeFive se compone de cuatro pasos básicamente: (1) preprocesamiento, en donde se extraen las dependencias y anotaciones de marco utilizando herramientas existentes (tales como CoreNLP y WFD – Word Frame Disambiguation –); (2) detectar roles de interfaz; (3) detectar roles específicos de VerbNet (principalmente semánticos) para un marco determinado; y (4) comprobar la compatibilidad entre la interfaz y los roles semánticamente específicos. Finalmente cabe destacar que este proyecto está escrito en Python y es de libre acceso[15].

Se ha decidido trabajar a partir de TakeFive ya que es el más completo y usable de todos; además de ser el más actual y el que incorpora los conocimientos previos de los otros sistemas.

lingüísticos orientados a predicados proporcionando de esta forma una gran cantidad de asignaciones lingüísticas que ayudan a una alineación semiautomática requiriendo un análisis lingüístico previo de las relaciones y su contexto.

[15] Código disponible en `https://github.com/TakeFiveSRL/TakeFiveSRL`