Universidad Nacional de Educación a Distancia (UNED)

Escuela Técnica Superior de Ingeniería Informática

Máster Universitario en I.A. Avanzada: Fundamentos, Métodos y Aplicaciones

Trabajo Fin de Máster: **Estudio de generación automática de distractores desde fuentes estructuradas y no estructuradas: el caso de psicofarmacología**

AUTOR: Omar Khalil Gómez

DIRECTOR: D. Rafael Martínez Tomás

CO-DIRECTOR: D. José Luis Fernández Vindel

Índice de Contenidos

1. Introducción 6
   1.1. Motivación y Objetivos 6
   1.2. Estructura del documento 9
2. Trabajos Relacionados 9
   2.1. Grafos de Conocimiento 9
      2.1.1. Generación de Grafos de Conocimiento 11
      2.1.2. Grafos de conocimiento en aplicaciones educativas 14
   2.2. Aprendizaje de Características de entidades 16
      2.2.1. Modelo de *embeddings* de palabras: word2vec (Skip-Gram) 17
      2.2.2. Modelo de *embeddings* de grafos de conocimiento: transE 21
      2.2.3. Modelo de *embeddings* Conjunto 22
   2.3. Generación automática de preguntas de hueco en blanco 25
      2.3.1. Identificación de enunciados 27
      2.3.2. Identificación de claves 27
      2.3.3. Propuesta de distractores 28
      2.3.4. Evaluación de sistemas de generación de FIBQs 28
3. Generación del Grafo de Conocimiento 29
   3.1. Metodología 29
   3.2. Creación del modelo semántico inicial 30
   3.3. Pre-procesado de fuentes textuales 34
   3.4. Búsqueda de entidades 35
   3.5. Filtrado de entidades y búsqueda de relaciones 36
   3.6. Consolidación de la Base de Conocimiento 37
   3.7. Resultados de la generación y modelo final 38
4. Generación de modelos de *embeddings* 40
   4.1. Generación de los modelos 41
      4.1.1. Skip-Gram 41
      4.1.2. Modelo transE 43
   4.2. Análisis de los modelos 46
5. Propuesta automática de distractores 49
   5.1. Método de generación de preguntas de hueco en blanco 49
   5.2. Evaluación 51
   5.3. Análisis de los resultados 53
6. Conclusiones y trabajos futuros 54
7. Referencias 55

Índice de Figuras

Figura 1. Ejemplo de pregunta de hueco en blanco 7
Figura 2. Ejemplo de datos semánticos (tripletas) en RDF 10
Figura 3. Mapa de la LOD Cloud (2020). Fuente: https://lod-cloud.net/versions/2020-07-27/lod-cloud.png 11
Figura 4. Esquema de un sistema educativo según el marco de Semantic Web-Based Education. Fuente: [26] 15
Figura 5. Arquitectura del modelo de lenguaje neuronal 17
Figura 6. Función de coste inicial del modelo Skip-Gram 18
Figura 7. Cálculo de probabilidades empleando softmax 18
Figura 8. Coste del modelo Skip-Gram con muestreo de instancias negativas 19
Figura 9. Probabilidad de una palabra de ser propuesta como contexto negativo 20
Figura 10. Intuición sobre las propiedades de analogía de los embeddings de palabras. Fuente: https://www.ed.ac.uk/informatics/news-events/stories/2019/king-man-woman-queen-the-hidden-algebraic-struct 20
Figura 11. Coste de entrenamiento según el modelo transE 21
Figura 12. Pasos del algoritmo de entrenamiento para el modelo transE 22
Figura 13. Probabilidad de observar una tripleta según el modelo conjunto 23
Figura 14. Función z para el cálculo de probabilidades en el modelo conjunto 23
Figura 15. Coste de probabilidad de observar una tripleta en el grafo de conocimiento 23
Figura 16. Coste total del modelo de conocimiento 23
Figura 17. Probabilidad de observar pares de palabras objetivo y contexto en el modelo conjunto 23
Figura 18. Coste total del modelo textual 24
Figura 19. Coste total del modelo de alineamiento basado en menciones 24
Figura 20. Coste total del modelo de alineamientos basado en tripletas nombradas 24
Figura 21. Coste total del modelo conjunto 24
Figura 22. Pseudocódigo de optimización del modelo conjunto 25
Figura 23. Elementos de una pregunta de hueco en blanco 25
Figura 24. Modelo semántico inicial acordado 34
Figura 25. Gramática para la identificación de términos candidatos 35
Figura 26. Ejemplo de JSON simplificado para almacenar los datos sobre entidades identificadas 36
Figura 27. Consulta SPARQL INSERT 38
Figura 28. Consulta SPARQL DELETE 38
Figura 29. Cobertura final del grafo de conocimiento generado 40
Figura 30. Matrices de confusión para el agrupamiento de entidades (palabras) en relación a su tipo 43
Figura 31. Costes de optimización modelo transE con 5 iteraciones sobre el conjunto total de tripletas 44
Figura 32. Matrices de confusión para la tarea de agrupamiento de nodos (entidades) empleando transE 44
Figura 33. Matrices de confusión para la categorización de entidades en el modelo mixto 46
Figura 34. Proyección de embeddings Skip-Gram 47
Figura 35. Proyección de embeddings transE 47
Figura 36. Proyección de embeddings conjuntos (solo palabras tipadas) 48
Figura 37. Proyección de embedding modelo conjunto (sólo nodos) 48

Figura 38. Ejemplo de enunciado afirmativo 50
Figura 39. Ejemplo de enunciado imperativo 50
Figura 40. Ejemplo de conversión de pregunta afirmativa a enunciado 50

Índice de Tablas

Tabla 1. Requisitos de las FIBQs para su empleo en escenarios de evaluación 26
Tabla 2. Entidades de interés tras primera entrevista 30
Tabla 3. Fuente de datos: Wikidata 31
Tabla 4. Fuente de datos: UMLS 31
Tabla 5. Fuente de datos: DrugBank 31
Tabla 6. Fuente de datos: PubChem 32
Tabla 7. Fuente de datos: DisGeNet 32
Tabla 8. Fuente de datos: CTD 32
Tabla 9. Fuente de datos: MeSH 32
Tabla 10. Fuente de datos: UniProt 33
Tabla 11. Human Disease Ontology 33
Tabla 12. Symptom Ontology 33
Tabla 13. Mapeo de tipos del modelo semántico acordado y las fuentes de búsqueda 36
Tabla 14. Número de entidades por tipo del grafo generado 39
Tabla 15. Número de enlaces por propiedad del grafo generado 39
Tabla 16. Configuraciones para el modelo Skip-Gram (word2vec) 42
Tabla 17. Coste final modelo Skip-Gram 5 iteraciones 42
Tabla 18. Variación de parámetros para el modelo conjunto 44
Tabla 19. Resultados de pureza y coste global en modelo conjunto 45
Tabla 20. Puntuaciones de las medidas de similitud en evaluación directa de similitudes 49
Tabla 21. Formulario de evaluación de FIBQs para el primer ejercicio 51
Tabla 22. Ejemplo de formulario del segundo ejercicio de evaluación - comparación de modelos 52

1. Introducción

La manera en la que se facilita la educación ha visto una evolución desde los modelos más tradicionales hasta otros más actuales como son la educación impartida on-line o los Cursos Masivos en Línea y Abiertos. En este tipo de entornos, se cuenta con un gran número de estudiantes, y por tanto los instructores pueden encontrarse con una mayor carga de trabajo a la hora de desempeñar sus tareas habituales.

Una de las tareas más complejas y que más tiempo conllevan bajo este tipo de entornos, es la generación de actividades formativas que faciliten la evaluación del conocimiento o competencias adquiridas. Por tanto, surge la necesidad de explotar las Tecnologías de la Información para aliviar dicha carga.

Este trabajo se basa en la idea de aprovechar los recursos educativos disponibles de una asignatura para la generación semiautomática de actividades educativas. En concreto, se pretende abordar varias fases de la generación automática de preguntas de hueco en blanco siguiendo un enfoque basado en técnicas de representación vectorial de entidades.

Para abordar este planteamiento, se desea mostrar cómo, partiendo de una serie de recursos textuales, podemos generar de manera semiautomática distintos modelos de similitud conceptual que serán empleados para la propuesta automática de distractores en preguntas de hueco en blanco generadas partiendo de un banco de preguntas de tipo test. En concreto, el trabajo está enfocado en mostrar la experiencia llevada a cabo en el contexto de la asignatura de Psicofarmacología, impartida en el Grado en Psicología ofertado por la Universidad Nacional de Educación a Distancia (UNED).

Veremos cómo, mediante la metodología propuesta, podemos generar un Grafo de Conocimiento partiendo de las entidades de interés de la asignatura previamente extraídas de las fuentes textuales. Analizaremos la riqueza de dicho Grafo de Conocimiento, representado mediante datos en formato semántico y enlazable, y daremos indicaciones sobre sus posibles empleos en ámbitos educativos.

Por otro lado, estudiaremos la utilidad de medidas de similitud vectorial basada en distintos modelos *embeddings* para la propuesta de distractores, teniendo en cuenta un conjunto de preguntas de test de años anteriores sobre el que se han identificado nuevas claves. Distinguiremos tres modelos en base al conocimiento explotado: un modelo de *embeddings* de palabras, un modelo de *embeddings* de grafos y un modelo conjunto o mixto apoyado en alineamientos.

Se realizará un análisis comparativo de la calidad de los distractores propuestos en cada caso, tratando de validar el método propuesto para su generación. Finalmente, se darán indicaciones para mejorar la calidad de las preguntas de hueco en blanco obtenidas para futuros desarrollos.

1.1. Motivación y Objetivos

Los cambios en la educación vienen determinados por condiciones económicas, sociales y tecnológicas [1]. Estos aspectos afectan la manera en la que se desempeñan las funciones por parte de las organizaciones educativas como las universidades, siendo algunos de los hechos más significantes la globalización (mayores oportunidades de acceso a la educación) y la flexibilidad demandada (menores restricciones temporales y espaciales) [2].

Estos factores han hecho que en los últimos años se haya visto una creciente demanda de la educación a distancia y la educación on-line, dando lugar a la popularización de los MOOCs, por

lo que actualmente encontramos cursos en los que participan cientos o incluso miles de estudiantes. De este modo, los instructores de este tipo de cursos deberían dedicar una gran cantidad de esfuerzo para generar y renovar las actividades formativas, así como para llevar a cabo la evaluación consiguiente. Por tanto, surge la necesidad de disponer de herramientas tecnológicas que faciliten las tareas mencionadas.

Uno de los tipos de actividades formativas y de evaluación más empleadas en distintos niveles de educación son las preguntas de tipo test. Este tipo de actividad es muy empleado puesto que a priori requiere de una menor carga administrativa y porque facilita una respuesta de evaluación inmediata [3]. Los ejercicios o preguntas de hueco en blanco (FIBQs de ahora en adelante por sus siglas en inglés: "*Fill-in-the-blank question*") son un tipo específico de ejercicios de tipo test en los que disponemos de los siguientes elementos básicos:

- Enunciado o pregunta: el enunciado de este tipo de ejercicios suele estar comprendido de una o varias sentencias afirmativas.
- Clave: el enunciado del ejercicio incluye un hueco en blanco correspondiente con la clave o respuesta correcta.
- Conjunto de distractores: son las respuestas propuestas no válidas que se incluyen en el listado de posibles respuestas junto con la clave.

(ENUNCIADO) Los déficit de _____ se relacionan con un procesamiento ineficaz en el córtex cingulado anterior:

a) Atención Selectiva **(CLAVE)**
b) Concentración **(Distractor 1)**
c) Aprendizaje **(Distractor 2)**
d) Vigilia **(Distractor 3)**

Figura 1. Ejemplo de pregunta de hueco en blanco

Este tipo de ejercicios son empleados en distintos ámbitos, siendo uno de los más empleados para la evaluación del aprendizaje de lenguas extranjeras, aunque también podemos encontrar su aplicación en las ramas de ciencias y tecnología. De esta forma, el tipo de conocimiento o inferencia que se requiere puede variar entre dominios de aplicación. En este trabajo nos centramos en aquellos dominios en los que el tipo de procesos mentales demandados al alumno esté enfocado en la conceptualización de una serie de entidades, así como el conocimiento de las relaciones que existen entre sí. Es decir, ámbitos en los que normalmente se desea que se establezcan similitudes o diferencias entre distintos tipos de entidades categorizables atendiendo a distinta tipología o relaciones de las entidades. Los ejemplos más claros de este tipo de ámbitos los podremos encontrar en áreas de biología, química, geología e informática, aunque creemos que el método podría ser de interés en otros dominios relacionados con las humanidades.

En este sentido, existe una gran cantidad de literatura dedicada a la generación (semi)automática de FIBQs basadas en entidades, pudiendo encontrar trabajos que se enfocan en distintas partes del proceso de generación:

1. Trabajos enfocados en la generación de enunciados.
2. Trabajos enfocados en la generación de distractores.
3. Trabajos enfocados en el control de dificultad.

4. Trabajos enfocados en la secuenciación y el *feedback*.

Este trabajo se centrará en la propuesta automática de distractores basados en entidades, aunque se abordarán ciertos aspectos de la generación de enunciados. Por tanto, se pretende disponer de un método adecuado de generación de distractores que justifique la implementación de un prototipo más elaborado.

El contexto o dominio de aplicación corresponderá con el de la asignatura de Psicofarmacología, asignatura de 6 créditos impartida durante el tercer curso del programa de Grado en Psicología ofertado en la Universidad Nacional de Educación a Distancia (UNED), y para la que disponemos de los siguientes recursos educativos:

- El libro en formato PDF recomendado en la asignatura: "Psicofarmacología esencial de Stahl: Bases neurocientíficas y aplicaciones básicas" 4ª edición del 2014. Por motivos de copyright, no se facilitará el texto como parte de la entrega del trabajo.
- Un banco de preguntas de tipo test empleadas durante la evaluación de la asignatura.

Teniendo en cuenta estos recursos, exploraremos métodos basado en la representación de entidades en espacios vectoriales para la propuesta de distractores sobre un conjunto de enunciados obtenido del banco de preguntas tipo test. Gracias a estos enfoques, podremos extraer, de manera no supervisada, una representación vectorial de cada entidad o concepto de interés que nos permitirá el cálculo de medidas de similitud entre entidades y/o grupos de entidades.

En concreto, se comparará la utilidad del empleo de medidas de similitud basadas en distintos modelos de *embeddings* para abordar a la generación automática de distractores. De entre dichos enfoques de Aprendizaje de Características, emplearemos los que pertenecen a las técnicas de representación de características sobre fuentes textuales y grafos (conocidas comúnmente como *word embeddings* y *knowledge graph embeddings* correspondientemente). Para ello, se dispondrá de un modelo Skip-Gram [4] generado desde las fuentes textuales, así como un modelo traslacional transE [5] que explotará un grafo de conocimiento en formato semántico y enlazable, extraído automáticamente desde fuentes estructuradas partiendo de una serie de términos identificados en las fuentes textuales. Mientras que en el enfoque textual las entidades se corresponden con términos textuales, en el enfoque traslacional las entidades se corresponden con nodos de un grafo.

Al disponer de distintas fuentes de conocimiento que podrían aportar distintas perspectivas, surge la idea de poder representar tanto entidades textuales como entidades del grafo en un mismo espacio vectorial. Así, surgen varios modelos de *embeddings* conjuntos, que ha demostrado obtener mejores representaciones en varios ámbitos y en distintas tareas. Por tanto, validaremos la aplicabilidad de uno de este tipo de modelos [6], y compararemos la calidad de los distractores generados frente a los modelos mencionados.

Una vez identificados nuestros objetivos y necesidades, presentamos las principales preguntas de investigación del trabajo:

- **P1. ¿Podemos emplear la medida de similitud basa en modelo conjunto para la generación automática de distractores en FIBQs?**
- **P2. ¿Qué medida de similitud (modelo) resulta el más adecuado de cara a la generación automática de distractores en FIBQs?**

Para lo cual se plantearán una serie de objetivos:

- **O1.** Describir una metodología para la generación de un grafo de conocimiento de la asignatura empleando herramientas Web y de Procesamiento del Lenguaje Natural (PLN). Aplicar el método sobre los recursos textuales de psicofarmacología.
- **O2.** Obtener representaciones vectoriales de entidades siguiendo el enfoque Skip-Gram, transE y modelo conjunto.
- **O3.** Comprobar la efectividad del modelo conjunto de cara a la tarea de generación automática de distractores.
- **O4.** Comparar la efectividad de los distractores generados siguiendo el modelo conjunto frente a los distractores generados siguiendo los enfoques Skip-Gram y transE.

1.2. Estructura del documento

La segunda sección estará dedicada a resumir los principales trabajos relacionado con el objetivo de conocer las técnicas computacionales y de la Inteligencia Artificial que se han empleado para el alcance de los objetivos que proponemos.

Seguidamente, la tercera sección muestra el método empleado para la generación del grafo de conocimiento de la asignatura con el soporte de técnicas de PLN y el acceso a distintas bases de datos de interés. Se mostrará el papel fundamental que desempeña el experto docente para la guía de la extracción de conocimiento. Por tanto, este apartado estará orientado a abordar el objetivo **O1**.

En el cuarto apartado se definen los modelos vectoriales que se emplearán para representar distintas entidades recopiladas de las fuentes de datos disponibles, de modo que se facilite el cálculo de medidas de similitud entre entidades, abordando el objetivo **O2**.

El quinto apartado muestra los pasos que se han llevado a cabo para generar los distractores, incluyendo explicaciones sobre cómo se han obtenido los enunciados de los que partimos. Además, se detallará el diseño de la evaluación, y se analizarán los resultados obtenidos. A este punto, se abordarán los objetivos **O3** y **O4.**

Finalmente, se dedicará el último apartado para dar indicaciones sobre el alcance de las preguntas e indicar las ventajas e inconvenientes de los métodos de cara a la posible introducción de mejoras en trabajos futuros**.**

2. Trabajos Relacionados

2.1. Grafos de Conocimiento

La Web Semántica surgió bajo la idea de poder describir distintos tipos de recursos empleando la infraestructura de la web. Estos tipos de recursos se pueden corresponder con recursos accesibles en la web o físicos, o bien pueden representar entidades de un dominio o ámbito concreto. La manera de describir estos recursos se realiza mediante el empleo de enlaces entre recurso que llevan una semántica asociada. De esta forma, se crearía una gran red semántica con distintos tipos de recursos, identificados mediante URIs, y descritos en base a una serie de propiedades. Así, se daría lugar a una base de conocimiento añadida sobre la web tradicional o web 2.0 [7].

Esto es posible gracias al stack tecnológico de la web semántica [8], pero también a la aplicación de las mejores prácticas de publicación de datos en la Web Semántica, lo que se conoce como los principios de los Datos Enlazados (Linked Data) [9]. En un escenario ideal de la Web Semántica, las descripciones estarían disponibles a través de datos o tripletas de datos facilitados según el modelo RDF, en alguno de los formatos soportados (RDF/XML, Turtle, N3,

Nquads, RDfa, etc.), y posiblemente ofrecidos a través de algún punto de consulta SPARQL. En dicho escenario sería indispensable el empleo de vocabularios (RDFS) u ontologías (OWL) para garantizar aspectos de interoperabilidad e integración y favorecer la inferencia de nuevo conocimiento, pudiendo este último beneficio verse acrecentado con el empleo de reglas (RIF). Por tanto, las tecnologías de la Web Semántica se fundamentan en los mecanismos de representación e inferencia del conocimiento como son las Lógicas Descriptivas y las Reglas de Asociación [8].

A continuación se muestra un ejemplo de datos en RDF para mostrar la unidad mínima de información en la web semántica (la tripleta) y el empleo de URIs para la identificación de recursos:

```
@PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

@PREFIX pf: < http://data.ia.uned.es/PF/>

pf:amoxapina rdf:type pf:Droga;

        pf:contribuye pf:trastornos_del_movimiento;

        owl:sameAs <http://wikidata.org/entity/Q58356>;

        rdfs:label “Loxapina”@es, “Amoxapina”@ es.
```

Figura 2. Ejemplo de datos semánticos (tripletas) en RDF

En el ejemplo anterior se emplean un total de 7 recursos (4 propiedades y 4 entidades o conceptos) identificables unívocamente con las URIs resultantes de concatenar la URI del prefijo con el resto del texto. En dicho ejemplo se muestra la descripción de la entidad “amoxapina”, y reflejaría hechos como que su empleo contribuye a la aparición de “trastornos del movimiento” o como que es un tipo de droga o fármaco. Este hecho se representa mediante una tripleta sujeto-predicado-objeto, que forma la unidad mínima de información de datos semánticos en RDF. En total, se muestran 5 tripletas en el ejemplo. El tipo de propiedades y tipos de entidades se encuentran definidos en la especificación del vocabulario u ontología, en el que se le da una semántica asociada a dichos elementos.

En el escenario ideal, también se emplearía con frecuencia el enlace del tipo owl:sameAs entre recursos para establecer que varios recursos se refieren a la misma entidad, concepto o cosa. Sin embargo, la práctica [10] nos enseña que el mundo de los Datos Enlazados hay una gran variedad de vocabularios y ontologías empleadas sobre mismos dominios, y que la Web Semántica se está centralizando en torno a los principales grandes conjuntos de datos semánticos como Wikidata. Además, ciertos conjuntos de datos han sido generados desde fuentes tanto estructuradas como no estructuradas, empleando algoritmos de Aprendizaje Automático o de Extracción y Recuperación de la Información en textos, por lo que podremos encontrar conjuntos de datos no manualmente curados.

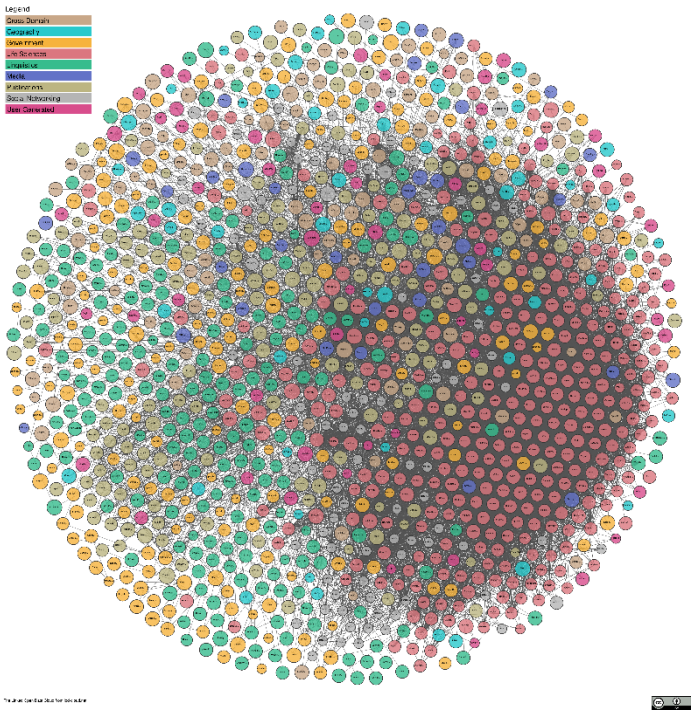


Figura 3. Mapa de la LOD Cloud (2020). Fuente: https://lod-cloud.net/versions/2020-07-27/lod-cloud.png

Impulsados por la actividad investigadora, en los últimos años se ha venido empleando el término grafo de conocimiento para referirse a este tipo de conjuntos de datos en formato semántico, entre otros, y que cuentan con un modelo o una semántica mínima asociada en cuanto a los tipos de entidades y propiedades que contienen [11]. En dichos trabajos, se trata de destacar los beneficios del empleo de grafos de conocimiento con varios propósitos [12] [13]:

- Mejorar la calidad de los patrones a emplear en tareas de minería de datos.
- Mejorar aplicaciones de Recuperación de la Información y Sistemas de Recomendación, así como de *Question Answering*.
- Proponer métodos para el refinado automático de los datos del grafo de conocimiento.

Existe una diversa topología de grafos de conocimiento disponibles en la web que pueden resultar de provecho para favorecer el desarrollo de distintas aplicaciones o agentes inteligentes. Sin embargo, la construcción de grafos de conocimiento puede requerir el empleo de herramientas avanzadas dependiendo del tipo de datos que se dispone y el propósito del grafo. Dedicaremos los siguientes apartados para mostrar los principales métodos de generación de grafos de conocimiento y las aplicaciones de los grafos de conocimiento en ámbitos educativos.

2.1.1. Generación de Grafos de Conocimiento

Desde el punto de vista de la Ingeniería del Conocimiento y los Sistemas Expertos, los grafos de conocimiento se pueden entender como Bases de Conocimiento. Generalmente, este tipo de grafos se construiría mediante un proceso de elicitación del conocimiento, en el que el grupo de ingenieros se reúne con los expertos del dominio para poder llevar a cabo una tarea de modelado de pericia para determinar los contenidos de la Abox (conjunto de aserciones, individuales) y la Tbox (componente terminológico, ontología). De este modo, una validación del grafo de conocimiento empleado se puede realizar atendiendo a su capacidad de responder a una serie de preguntas de competencia que, en caso de haber seguido los estándares del *stack*

tecnológico de la Web Semántica, podrían estar representadas a modo de una batería de preguntas SPARQL.

En [14] se identifica una metodología de ingeniería para la creación de Grafos de Conocimiento en contextos empresariales. Los autores resaltan la existencia de diferencias entre Grafos de Conocimiento y Bases de Conocimiento, indicando que las primeras no requieren de tanto poder expresivo, y por consiguiente el proceso de ingeniería del vocabulario o *schema* del grafo de conocimiento puede realizarse con una metodología más ágil que las empleadas en proyectos de Ingeniería Ontológica, pero aun así el rol de Ingeniero de Conocimiento sigue siendo fundamental. La metodología propuesta se basa en un proceso iterativo compuesto de tres fases: (1) el análisis de requisitos, durante la cual se analiza y restringe el dominio mediante preguntas de competencia; (2) la conceptualización del vocabulario, durante la cual se identifican clases y relaciones presentes en el dominio en un formato que sirve de mediador entre ingenieros y expertos; y (3) la implementación en lenguaje máquina, que requiere de la reutilización de vocabularios existentes. El proceso iterativo va modificando el vocabulario a medida que se dispone de datos que lo completen, lo cuales van guiando los posibles cambios.

El Consorcio de Ontologías Biomédicas Abiertas (OBO, por las siglas en inglés de "The Open Biomedical Ontologies") diseñó una solución para la integración de ontologías del ámbito biomédico basándose en herramientas que facilitan el desarrollo colaborativo de ontologías. En [15], explican las experiencias obtenidas durante la integración de ontologías biomédicas en un solo marco conceptual. Esto fue posible gracias a la creación de un portal web para la modificación, revisión y curado de ontologías por parte de distintas comunidades de investigadores. Para ello, se crearon distintas comunidades acorde a ciertas particiones de los dominios de conocimiento que abordan, de manera que cada comunidad es capaz de crecer independientemente. La integración de las distintas ontologías que conforman OBO se hizo posible gracias a la creación de modelos específicos de relaciones entre ontologías y tareas específicas que sirvieron de guía a los investigadores para el enlazado de los modelos.

También podríamos considerar como grafos de conocimiento a aquellos conjuntos de datos albergados en bases de datos no relacionales (NoSQL), siempre y cuando las relaciones entre elementos tengan una semántica asociada interpretable también por humanos, y esos elementos sean categorizables. Para estos casos, podemos encontrar herramientas que facilitan la generación personalizada de grafos de conocimiento desde bases de datos tanto relacionales como no. Además, también se dispone de herramientas de creación de grafos partiendo de otros grafos de conocimiento ya existentes, o conjuntos de datos en otros formatos más tradicionales como JSON, XML, XSLT o HTML. Así, podemos ver que la generación de un grafo de conocimiento desde datos estructurados o semi-estructurados requiere de herramientas capaces de integrar datos de fuentes heterogéneas.

En [16] se detalla la construcción de un grafo de conocimiento sobre recetas y la información sobre los alimentos que la componen para facilitar servicios de información dietética. Los autores identifican una serie de conjuntos de datos semi estructurados presentes en la web que no están enlazados a pesar de que contienen diversa información relacionada sobre recetas, alimentos, e información tanto composicional como nutricional. La información inicialmente está contenida en tres repositorios de datos, en los que dos de ellos, dedicados a la información de recetas y la clasificación general de tipos de alimentos, se encuentran disponibles en formato semántico y enlazados con grafos de conocimiento del ámbito de los Datos Enlazados (DBPedia y Wikidata). Por otro lado, el *dataset* referente a la información nutricional se encuentra accesible para la recuperación de información en formato tabular. Gracias al enlazado externo,

los autores recuperan una serie de etiquetas alternativas sobre alimentos. Después, dichas etiquetas son empleadas para buscar menciones en el *dataset* sobre información nutricional, que es recuperado mediante técnicas de *scraping*. De esta forma, crean enlaces con este nuevo conjunto de datos, el cual es transformado finalmente a RDF con el empleo de una herramienta de conversión de datos desde fuentes tabulares [17].

El trabajo de [18] resumen los pasos llevados a cabo para crear un grafo de conocimiento en el ámbito de la bioinformática. Los autores hablan de la creación de un *dataset* como mezcla o *mash-up* de varios conjuntos de datos accesibles en la web a través de tres grandes fases: (1) Llevar a cabo un ejercicio de modelado ontológico para establecer los principales tipos de entidades contenidos en los *datasets* a integrar, se investigan los portales (en concreto las páginas HTML o datos XML que proveen) para identificar también posibles relaciones entre dichos elementos; (2) el desarrollo de una serie de *scripts* para la generación de tripletas en RDF, capaces de recuperar y explotar los datos que proveen los conjuntos de datos a integrar en distintos formatos (XML, SQL, HTML); y (3) la normalización de los identificadores o URIs para la identificación unívoca de los elementos y el enlazado entre conjuntos de datos.

Por otra parte, gracias a los avances en los campos de PLN dedicados a la Extracción de la Información, podemos generar grafos de conocimiento partiendo de fuentes textuales, aunque su calidad suele estar muy condicionada a la configuración concreta de cada problema y la cantidad de datos etiquetados que se disponen [13]. Esto es así cuando se emplean las técnicas de Extracción de la Información dedicadas a la Extracción de Entidades o la Extracción de Propiedades, lo que conlleva afrontar problemas de Enlazado y Resolución de Entidades, así como llevar a cabo una evaluación formal de los resultados extraídos, usualmente mediante precisión y cobertura dependientes de la disposición de datos etiquetados.

El trabajo de [19] detalla cómo se ha generado un grafo de conocimiento de metadatos sobre patrimonio cultural que parte de una serie de textos escritos en lengua francesa que son inicialmente traducidos al inglés. Con el empleo de sistemas de extracción de entidades y relaciones previamente entrenados para lengua inglesa, realizan un etiquetado de los textos para obtener un conjunto inicial de entidades y relaciones. Posteriormente, al comprobar la riqueza de las anotaciones, se identificaron una serie de carencias, es decir, entidades o relaciones que no quedaron adecuadamente identificadas. Por esto, los autores generaron un nuevo sistema de identificación en base a nuevas características de las entidades y propiedades restantes, que fue entrenado tomando las etiquetas de los resultados iniciales. Gracias a esto, en una segunda vuelta, la evaluación de la extracción mejoró y se optó por la representación en RDF de los elementos extraídos.

La generación de Grafos de Conocimiento desde texto también se basa en el empleo de grandes lexicones o diccionarios de estructuras lingüísticas. Así, los autores de [20] emplean el uso de una base de datos semántica de marcos para poblar bases de conocimiento. En su enfoque, emplean la semántica de marcos para identificar estructuras lingüísticas y los participantes o roles presentes. Para ello, inicialmente crean las anotaciones de todas aquellas unidades léxicas que se han identificado con su rol asociado dentro del marco semántico. Posteriormente, estas anotaciones se representan en RDF y son sometidas a un proceso de transformación con el empleo de reglas SPARQL hasta que se pasa de una estructura RDF de menciones y roles a una estructura propia del grafo de conocimiento. Este grafo queda enlazado con fuentes externas, en concreto con DBPedia, ya que el formato de los marcos asocia roles con tipos de entidades de dicho grafo de conocimiento.

Por otra parte, el paradigma de OpenIE establece que podemos generar grafos de conocimiento sobre un texto del que a priori no se han establecido los tipos de entidades que se incluyen de una manera no supervisada [21]. Para conocer más sobre los métodos de creación de grafos desde fuentes textuales, se recomienda consultar [13].

La evaluación de la calidad de los grafos de conocimiento generados automáticamente desde texto también puede ser evaluada empleando preguntas de competencia, aunque el ruido que este tipo de técnicas conlleva sobre el grafo final, hace que sea necesario evaluar la calidad de la generación. Si bien esto es de especial importancia para mejorar los algoritmos de generación, sobre todo desde texto, la evaluación no se contempla en la mayor parte de trabajos de generación de grafos de conocimiento a gran escala [22]. Es decir, normalmente no se realiza una evaluación acerca del porcentaje de tripletas correctas que contiene un KG. Una evaluación manual obviamente no es recomendada por el esfuerzo requerido, y normalmente la extracción se realiza sin la existencia de un *gold standard*. Por este motivo, algunos enfoques se basan en facilitar técnicas avanzadas de *sampling* de tripletas para su evaluación [22].

Finalmente, es habitual someter los grafos de conocimiento que incluyen ruido a procesos de refinado posteriores para la corrección, borrado o inclusión de nuevas tripletas. Dichos métodos se basan en el empleo de modelos de representación de grafos [23] y modelos predictivos de enlaces para los cuales se evalúa su precisión [24].

2.1.2. Grafos de conocimiento en aplicaciones educativas

El uso de ontologías es muy extendido para afrontar ciertos aspectos de interoperabilidad o mejorar los sistemas de recuperación de información de las organizaciones educativas. Cuando atendemos a aquellas aplicaciones que se dedican a dar soporte a los procesos de aprendizaje enseñanza, nos adentramos en el mundo del e-learning o de la educación basada en web, en la que es muy común encontrar el empleo de modelos o conjuntos de datos semánticos para su funcionamiento [25].

Un marco general para el desarrollo de sistemas e-learning basados en web semántica es facilitado en [26]. Según este tipo de marco aplicativo, encontraríamos varios tipos de ontología dedicados a modelar distintos aspectos de los procesos enseñanza/aprendizaje como:

- Ontologías de dominio: representan los conceptos, relaciones y teorías esenciales de un dominio de interés. Estas ontologías son clave para poder llevar una traza del conocimiento que se va profundizando por parte de los estudiantes. En [26], dan ideas para generar este tipo de ontologías de una manera poco expresiva pero efectiva mediante mapas conceptuales o *topic maps*.
- Ontologías de tarea: representan las características de aquello que se desea que el estudiante sea capaz de completar o resolver. Por tanto, incluye una serie de problemas, escenarios, preguntas o explicaciones para formalizar las actividades de la experiencia de aprendizaje y los actores involucrados.
- Ontologías de estrategia de aprendizaje: estas modelan las acciones pedagógicas y el comportamiento que se debe seguir en base a los posibles estados del estudiante durante la experiencia de aprendizaje.
- Ontologías de modelo de estudiante: modelan las acciones e interacciones del estudiante con el sistema, así como el estado del mismo en relación con los materiales consultados, tareas llevadas a cabo así como el rendimiento del estudiante a la hora de enfrentarse a las tareas de una experiencia de aprendizaje.

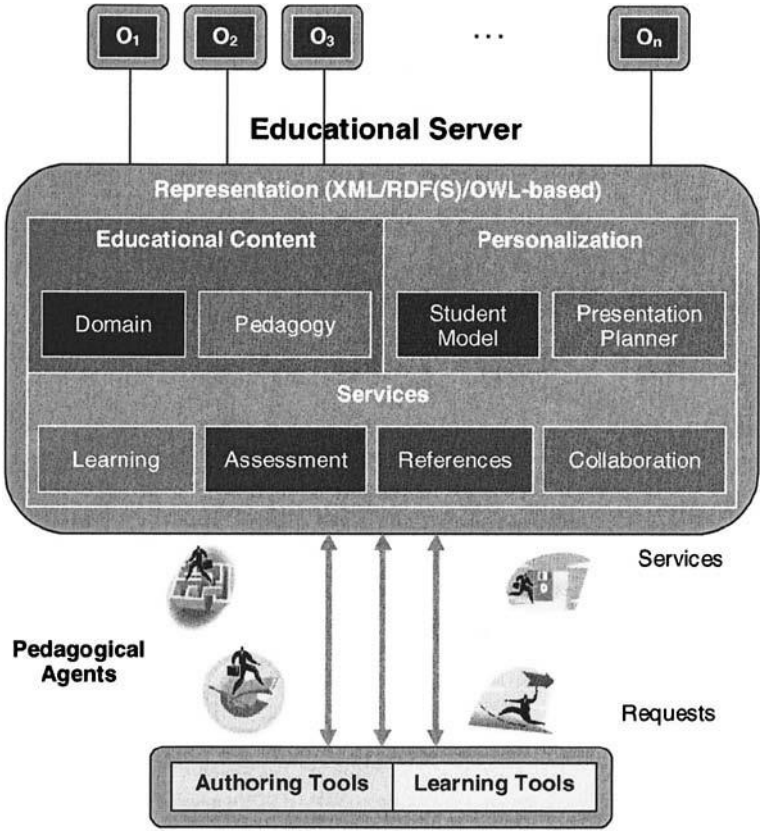


Figura 4. Esquema de un sistema educativo según el marco de Semantic Web-Based Education. Fuente: [26]

De este modo, resulta imprescindible el empleo de ontologías de dominio bajo sistemas educativos amoldados a este marco teórico para poder facilitar una mayor personalización y precisión a la hora de generar experiencias de aprendizaje acorde a las características de los estudiantes. Es decir, si el sistema es consciente del conocimiento (o parte del conocimiento) que se desea transmitir durante las experiencias de aprendizaje, podremos ofrecer recursos educativos específicos, mantener una traza de la consecución de objetivos de aprendizaje con precisa, y generar procesos de evaluación personalizados [26].

Por tanto, la primera aplicación que surge es poder emplear grafos de conocimiento como ontologías de dominio en sistemas e-learning, complementando a los tradicionalmente empleados mapas conceptuales o *topic maps*. En la actualidad, podemos encontrar varios ejemplos de sistemas e-learning en los que la ontología de dominio se puede entender como un grafo de conocimiento que resulta imprescindible para alcanzar los objetivos mencionados.

Por ejemplo, en [27], se propone el empleo de grafos de conocimiento para poder analizar de manera precisa los caminos de aprendizaje de los alumnos de un curso de matemáticas. Estos caminos de aprendizaje pueden entenderse como una secuencia de tareas con recursos educativos asociados, incluyendo tareas de evaluación. Gracias al empleo de un grafo de conocimiento en el dominio de matemáticas, los autores pueden llevar a cabo análisis más precisos sobre aquellas partes del dominio que resultan más relevantes para determinar el rendimiento de los alumnos, y así poder identificar dependencias entre elementos del grafo que finalmente son empleadas para identificar aquellos caminos más o menos adecuados (en función del rendimiento de los alumnos que ya lo siguieron).

Otro trabajo [28] está orientado a facilitar una herramienta de recomendación e integración de material educativo. Dicha herramienta permite a los instructores la creación de recursos

educativos y su etiquetado en base a una serie de conceptos que se desean abordar y que forman parte del grafo de conocimiento del dominio del curso (o experiencia). Por otro lado, la herramienta también es capaz de extraer conceptos de recursos educativos existentes y relacionarlos con el grafo de conocimiento. De esta forma, se construye un sistema recomendador de recursos educativos capaz de sugerir elementos en base a nuevos conceptos que se desean abordar.

Los autores de [29] muestran cómo emplear un grafo de conocimiento creado desde fuentes textuales para mejorar la adaptabilidad de un sistema mentor. Dicho sistema se basa en el grafo de conocimiento textual, al que se le añade nuevo conocimiento siguiendo un proceso de ingeniería entre instructor y técnico, para identificar carencias de conocimiento antes de comenzar una tarea propuesta y así proponer acciones personalizadas en base a las carencias de conocimiento identificadas en un estudiante.

Por otra parte, cuando disponemos de un grafo de conocimiento que refleja el conocimiento que se desea transmitir de una manera precisa, este puede constituir, por sí mismo, una herramienta de gran valor en contextos educativos. Por tanto, creemos que en ciertos dominios podrían emplearse interfaces de edición de grafos de conocimiento con distintos propósitos:

- Permitir a los alumnos navegar por los conocimientos de la asignatura a través de un modelo de datos que se asemeje al tipo de esquemas o mapas conceptuales que los estudiantes pueden manejar.
- Facilitar a los profesores de una herramienta de integración y etiquetado de sus contenidos.
- Permitir a instructores y alumnos realizar consultas sobre el tipo de conocimiento que deben adquirir para la resolución de sus dudas.
- Facilitar el desarrollo de aplicaciones de evaluación basadas en preguntas de tipo test.

Finalmente, si contamos con que los grafos de conocimiento empleados se encuentran en formato semántico y publicados acorde a los estándares de los Datos Enlazados, dispondremos de mayores facilidades para su enriquecimiento [30], lo que puede originar nuevos usos del grafo.

2.2. Aprendizaje de Características de entidades

El éxito de las tareas de Minería de Datos depende en gran medida de la adecuación de los datos o muestras que se emplean para la generación de los modelos de Aprendizaje Automático empleados en dichas tareas. Por este motivo, no es de extrañar que se empleen métodos de pre-procesado o limpieza de las muestras, así como métodos de transformación con el objetivo de que las muestras contengan conocimiento codificado de valor para una tarea en concreto [31].

Tradicionalmente, las características eran codificadas acorde al conocimiento previo que se dispone sobre los datos o el problema a resolver, lo que se conoce como Ingeniería de Características (o Feature Learning). Sin embargo, en los últimos años, se ha ido optando por el empleo de modelos para el aprendizaje automático de estas características, lo que ha supuesto un gran avance en la IA dado que las tareas de Minería de Datos son menos costosos y menos dependientes de las decisiones de ingeniería, lo que fomenta la comparación efectiva de resultados en aplicaciones y el entendimiento general de qué consideramos como buenas características [31].

En este trabajo exploramos técnicas de aprendizaje de características de entidades, que pueden ser entendidas como términos o palabras siguiendo un enfoque textual; o bien puede corresponderse con los nodos de un grafo de conocimiento. Los métodos más populares para aprender características de dichos elementos son los modelos de *embeddings* de palabras y los modelos de *embeddings* de grafos de conocimiento. Ambos métodos explotan las propiedades de las fuentes para generar representaciones vectoriales de palabras, o bien de nodos y aristas. Disponer de este tipo de representaciones hace que podamos emplear algoritmos de Aprendizaje Automático de una manera inmediata, y de este modo emplearlos en tareas extrínsecas con gran facilidad.

Especial interés tienen los modelos de *embeddings* conjuntos, capaces de aprovechar tanto texto como grafos de conocimiento para la generación de representaciones vectoriales de palabras y nodos en un mismo espacio vectorial, lo que introduciría nuevas capacidades como el cálculo de similitudes entre entidades-palabra y entidades-nodo. Sin embargo, es necesario conocer cómo se aprovechan ambos tipos de conocimiento para comprender el rendimiento de las representaciones generadas con estos modelos en distintas tareas extrínsecas.

Dedicamos los siguientes tres apartados para mostrar tres modelos de aprendizaje de características acorde al tipo de conocimiento o fuente de datos empleada.

2.2.1. Modelo de *embeddings* de palabras: word2vec (Skip-Gram)

Disponer de representaciones vectoriales de palabras es imprescindible para poder afrontar distintas tareas del ámbito del PLN como la traducción, el reconocimiento del habla o la identificación de entidades nombradas. Si bien la tónica habitual era la de codificar una serie de características de las palabras con conocimiento lingüístico experto, el paradigma ha ido optando por confiar en modelos que generan estas características de manera automática [31].

Uno de los primeros enfoques que se aplicaron para generar este tipo de representaciones para palabras de un texto, consistía en emplear un modelo neuronal probabilístico, con una capa oculta y una capa de salida que sería entrenado para predecir la palabra que acompañaría a una secuencia de palabras dada:

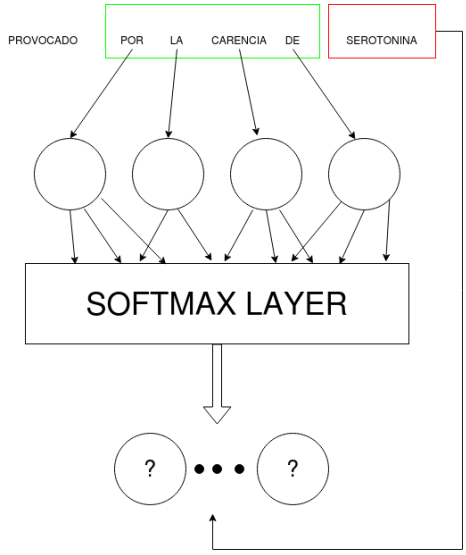


Figura 5. Arquitectura del modelo de lenguaje neuronal

El de modelos de lenguaje de [32] se basa en la idea de entrenar un modelo en una tarea de predicción, lo que origina una matriz de pesos entre las capas de entrada y la oculta, que se van

ajustando a medida que el sistema aprende y se minimiza el coste de predicción. Es esta misma matriz de pesos la que, después del entrenamiento, obtendremos como representaciones de palabras. De este modo, dispondríamos de un vector de características por cada una de las palabras que se intentan predecir (vocabulario de un corpus dado). Así, podríamos emplear una codificación vectorial *one-hot* de cada palabra de forma que, al multiplicar dicho vector por la matriz de pesos de la capa intermedia, se recuperaría la representación vectorial aprendida de la palabra. Este enfoque, en la práctica, es sustituido por funciones de mapeo de identificadores a posiciones de la matriz, cosa que se emplea tanto en los métodos de *embeddings* de palabras como el resto de métodos que consideraremos en este trabajo. Sin embargo, lo interesante de este modelo es que las representaciones de palabras con contextos de predicción similares producen (pesos) vectores similares, lo que constituye una hipótesis de la semántica distribucional.

En la práctica, el modelo propuesto en [32] tiene el problema de que en la capa de salida se debe realizar una clasificación entre todas las palabras posibles de nuestro vocabulario, lo que lo hace impracticable su aplicación frente a grandes *corpora* en los que el conjunto de palabras distintas, conocido como vocabulario, contiene un número muy elevado de elementos. A medida que se investigó en el aprendizaje de representaciones, se comenzaron a emplear modelos aún más simples que demostraron ser más útiles. Este fue el ejemplo de los trabajos [33] y [4], que marcaron un antes y un después en el campo del aprendizaje de características y, en concreto en el área de Procesamiento del Lenguaje Natural. En estos trabajos se presentan los dos principales modelos de *embeddings* de palabras popularmente conocidos como word2vec: el modelo CBOW y el modelo de Skip-Gram, siendo este último el que detallaremos a continuación y pondremos en práctica más adelante.

En primer lugar, el modelo Skip-Gram [4] se basa en la inclusión de una ventana bidireccional para contemplar las palabras posteriores como parte del cálculo de las probabilidades durante el entrenamiento. La idea general es entrenar un sistema para predecir palabras de contexto dada una palabra objetivo, lo que se corresponde con la maximización del coste:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

Figura 6. Función de coste inicial del modelo Skip-Gram

Que corresponde a la probabilidad logarítmica media, siendo T el número total de palabras distintas del vocabulario. De esta forma, dada una palabra objetivo *wt,* queremos maximizar la probabilidad logarítmica de cada palabra *wt+j* que se encuentra dentro de la ventana de contexto bidireccional de tamaño *c*. Así, dicha probabilidad se define de acuerdo a:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top} v_{w_I}\right)}$$

Figura 7. Cálculo de probabilidades empleando softmax

Que se corresponde con el empleo de una capa lineal con función de activación *softmax*, donde *vwI* se corresponde con el vector o *embedding* de la palabra *wI*, mientras que *v′w* se

corresponde con los parámetros o pesos de la capa de clasificación. Así, lo que se pretende maximizar es el producto escalar de las palabras objetivo *vwI* frente a las palabras de contexto *v′w.* De este modo, podemos pensar que los vectores de las palabras que pertenecen a un mismo contexto se situarían cerca en el espacio vectorial, mientras que las palabras que no comparten contextos similares, quedarían alejadas entre sí.

Sin embargo, es necesario llevar a cabo una normalización de las probabilidades que, según este planteamiento, debería tener en cuenta todas las posibles palabras del vocabulario (ver denominador), lo que hace que el objetivo sea computacionalmente complejo de obtener. Por tanto, los autores del modelo Skip-Gram proponen medidas para mitigar el coste computacional del factor de normalización en la capa de salida tipo *softmax*.

De entre dichas optimizaciones, se propuso el empleo de clasificadores jerárquicos en estructura de árbol y técnicas de muestreo de instancias negativas, siendo este último enfoque el considerado como más adecuado dados los resultados empíricos del empleo de las representaciones en tareas de Minería de Textos [4]. Esta política de *sampling* (Noise Contrastive Estimation) permite cambiar el objetivo de la clasificación desde un modo de uno a varios, a disponer de distintos clasificadores binarios. Por lo tanto, en cada época de entrenamiento sólo sería necesario modificar los pesos de varios de los clasificadores acorde a la maximización de la siguiente expresión:

$$\log \sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^{\top} v_{w_I}) \right]$$

Figura 8. Coste del modelo Skip-Gram con muestreo de instancias negativas

Que se corresponde con la aproximación al problema siguiendo un enfoque de regresión logística en el que se mantiene el objetivo de maximizar el producto escalar de las palabras objetivo y contexto (primer sumando), a la vez que se desea minimizar el producto escalar de dicha palabra objetivo frente a un conjunto de palabras de contexto negativas (segundo sumando) que son las generadas por la política de *sampling* que genera ejemplos negativos suponiendo una cierta distribución $Pn(w)$. Así, en la capa de salida, la clasificación se realiza tan sólo teniendo en cuenta un subconjunto de clases. Este subconjunto de clases se corresponde con el de la palabra objetivo y una serie de palabras contexto generadas mediante la técnica de muestreo de instancias negativas. Por tanto, a pesar de que la manera adecuada de generar representaciones sería empleando el coste incluido en la Figura 7, podemos simular un comportamiento parecido empleando regresión logística para determinar qué pares de palabras objetivo-contexto pertenecen a nuestro conjunto de datos (corpus de referencia) y qué pares no pertenecen.

Más concretamente, el método de entrenamiento empleando *sampling* de instancias negativas funciona del siguiente modo:

1. Dada una palabra del vocabulario, se buscan todas las palabras que aparecen dentro del contexto o ventana de la palabra, generando un par positivo de palabra objetivo y palabra contexto.
2. Se generan de manera aleatoria, por cada par identificado previamente, k nuevos pares objetivo-contexto (según los autores de entre 2 a 5 muestras), con la peculiaridad de

que los nuevos contextos no se dan en el corpus de referencia (muestreo de instancias negativas).

3. Se actualizan los pesos en la capa de salida. En concreto, se actualizan únicamente los pesos correspondientes a los clasificadores binarios de la palabra objetivo y todas las palabras contexto, tanto positivas como negativas, de acuerdo al nuevo objetivo de entrenamiento.

Para poder llevar a cabo una política de entrenamiento como la propuesta, los autores del modelo Skip-Gram acuerdan una serie de parámetros para controlar la cantidad de veces que una misma palabra se escoge como objetivo, así como para establecer el número de instancias negativas k necesarias por cada instancia positiva. Para ello, la manera de llevar a cabo este *sampling* de palabras se realiza de manera heurística, empleando la frecuencia de las palabras para no favorecer que las palabras que más aparecen en el corpus sean las más probables de escoger como palabra objetivo:

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

Figura 9. Probabilidad de una palabra de ser propuesta como contexto negativo

Una vez que sometemos un corpus de referencia al modelo Skip-Gram, podemos observar cómo, para palabras de contextos similares, obtenemos representaciones más similares o cercanas dentro del espacio vectorial de los *embeddings*. Además, otra de las propiedades de las representaciones generadas es su capacidad de establecer analogías mediante la adición y sustracción de los vectores de palabras. Los autores del modelo muestran cómo, para ciertas palabras, es posible llevar a cabo inferencias de analogía según ilustramos a continuación:

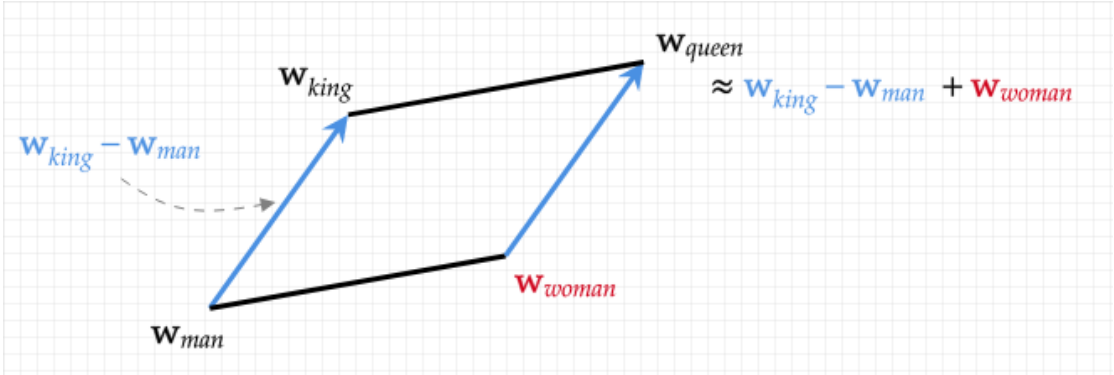


Figura 10. Intuición sobre las propiedades de analogía de los embeddings de palabras. Fuente: https://www.ed.ac.uk/informatics/news-events/stories/2019/king-man-woman-queen-the-hidden-algebraic-struct

Podemos pensar que esto es posible ya que se espera que vector de la palabra rey ($\mathbf{w}king$) y el vector de la palabra reina ($\mathbf{w}queen$) guarden ciertas similitudes (sus características son similares), de modo que la resta de estos dos vectores resaltaría las diferencias de los rasgos aprendidos para dichas palabras. La intuición reside en que, a pesar de que los rasgos no son interpretables (sino más bien considerados como variables latentes), la diferencia entre rasgos de las palabras mencionadas resultaría en la identificación de los rasgos encargados de codificar el conocimiento abstracto sobre el género (masculino o femenino). Por este motivo, empíricamente se ha demostrado la adecuación de las operaciones de sustracción y adición con *embeddings* de palabras para la inferencia de analogías.

Esta intuición es aplicable a otros modelos de *embeddings*, que basan su éxito en las ideas del modelo Skip-Gram, como veremos a continuación en el caso de los *embeddings* de grafos de conocimiento.

2.2.2. Modelo de *embeddings* de grafos de conocimiento: transE

Del mismo modo que podemos aprender representaciones vectoriales de palabras dado un corpus, podemos aprender representaciones vectoriales de los nodos y aristas de un grafo. Este tipo de métodos pertenecen al aprendizaje de características de datos multi-relacionales, aunque también existen métodos más cercanos a las áreas de aprendizaje de características aplicado a redes u otras estructuras representables mediante grafos.

En concreto, el método de *embeddings* de grafos de conocimiento que exponemos y que emplearemos más adelante, pertenece a la familia de métodos traslacionales. Este tipo de modelos se basan en la idea de modelar las relaciones entre nodos como traslaciones en el espacio vectorial donde quedan representados los nodos. Esta interpretación geométrica hace que los modelos sean intuitivos, y nos permite establecer similitudes con el modelo de *embeddings* de palabras anteriormente descrito.

Aquí describimos el modelo transE presentado en [5], que al igual que sucedió con los modelos anteriormente vistos, la presentación de este modelo a la comunidad científica marcó un antes y un después debido a la calidad de los resultados obtenidos con las representaciones generadas por dicho método en tareas de predicción de enlaces, mejorando el rendimiento de aplicaciones como los Sistemas de Recomendación.

La idea central del modelo transE es considerar que, si una determinada tripleta pertenece al grafo de conocimiento, entonces las representaciones vectoriales de la entidad objeto deben situarse cerca de la representación de la entidad sujeto más el vector que representa el tipo de relación. De lo contrario, si una determinada tripleta no pertenece al grafo de conocimiento, se deseará que la suma de las representaciones del vector sujeto y el vector relación queden lejos de la entidad sujeto. Esto es posible debido considerando que el objetivo del modelo se basa en minimizar la siguiente expresión:

$$\mathcal{L} = \sum_{(h,\ell,t)\in S} \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h}+\boldsymbol{\ell},\boldsymbol{t}) - d(\boldsymbol{h'}+\boldsymbol{\ell},\boldsymbol{t'})\right]_+$$

Figura 11. Coste de entrenamiento según el modelo transE

Donde $\boldsymbol{h}$, $\boldsymbol{l}$ y $\boldsymbol{t}$ se corresponden con los vectores entidad (*head*), propiedad (*link*) y objeto (*tail*) correspondientes con las tripletas del grafo de conocimiento, y $\boldsymbol{h'}$, $\boldsymbol{t'}$ se corresponden con corrupciones de la tripleta original que no pertenecen al conjunto de tripletas del grafo; γ es el tamaño de un margen de distancia; d se corresponde con la distancia Manhattan o distancia Euclídea; y el símbolo + representa la parte positiva de aquello que está entre corchetes.

Por tanto, la idea es que las relaciones se correspondan con traslaciones de los *embeddings*, para lo cual es necesario emplear una función de distancia, que puede corresponderse con la distancia de Manhattan o distancia euclídea. Gracias a este cálculo de distancias, podemos definir el coste global de optimización como la suma de diferencias entre las distancias de las tripletas originales y las distancias de las tripletas corruptas. De este modo, a medida que la distancia en las tripletas originales es menor, menor será el coste global de la función. De lo contrario, cuando la distancia entre tripletas originales sea mayor que la distancia entre tripletas

corruptas, implicará que el margen no se está respetando y obtendremos valores positivos del coste a minimizar.

Del mismo modo que ocurría con el modelo SKip-Gram, se emplea una política de generación de instancias negativas para obtener tripletas corruptas y garantizar la eficiencia del algoritmo, cuyo proceso se detalla a continuación [5]:

1. Inicializar los vectores entidades y relaciones de manera aleatoria
2. Por cada entidad:
    a. Normalizar el vector entidad
    b. Obtener un conjunto inicial de tripletas en los que participe el vector entidad
    c. Por cada tripleta:
        i. Se obtiene un conjunto de tripletas corruptas
        ii. Se añaden las tripletas corruptas al conjunto inicial de tripletas de la entidad
    d. Actualizar las representaciones acorde a la expresión de la Figura 11.

Figura 12. Pasos del algoritmo de entrenamiento para el modelo transE

Así, la optimización en transE se realiza mediante *batches*, partiendo de una inicialización aleatoria de los vectores de entidades y relaciones. El proceso continúa hasta llegado un número de épocas establecidas o bien acorde al rendimiento del modelo alcanzado sobre un conjunto de tripletas de validación frente a la tarea de predicción de enlaces.

Los autores del modelo transE llevaron a cabo una serie de experimentos en los que descubren que este método genera representaciones con las que se obtuvieron mejoras en tareas de predicción de enlaces frente a los métodos propuestos hasta el momento. Así, a partir de este modelo, surgieron otros métodos traslacionales más complejos, pero todos ellos basados en las mismas ideas de traslación, que a fin de cuentas, se basan en las ideas de las propiedades exhibidas por los *embeddings* de palabras del modelo anteriormente visto, algunos de los cuales generan representaciones con las que se obtienen aún mejores resultados dependiendo de la naturaleza del grafo del conocimiento. Para conocer el resto de modelos traslacionales, así como las ventajas del empleo de estos modelos en tareas de predicción, se recomienda [24].

2.2.3. Modelo de *embeddings* Conjunto

Dadas las comunalidades entre las bases de los modelos de *embedding* presentados, la pregunta que surge es ¿podemos obtener mejores representaciones entrenando *embeddings* de palabras y *embeddings* de grafos simultáneamente en el mismo espacio vectorial? Los resultados obtenidos empleando ciertos modelos mixtos o conjuntos muestran que estos modelos generan representaciones vectoriales con las que se obtienen mejoras frente a tareas de clasificación de tripletas (decidir si una tripleta pertenece o no a un grafo) y tareas de extracción de entidades y relaciones, en comparación con los modelos entrenados únicamente con *embeddings* de grafos y *embeddings* de palabras respectivamente [6] [34] [35].

La principal idea de estos modelos reside en llevar a cabo una optimización conjunta de las distintas representaciones, de manera que se puedan calcular distancias entre entidades que correspondan a palabras y entidades que correspondan a nodos. En concreto, en [6], se propone un modelo probabilístico compuesto de tres componentes: el modelo textual, el modelo de conocimiento y el modelo de alineamientos.

El modelo de conocimiento emplea una versión probabilística del modelo transE explicado previamente. Para cada tripleta, definiremos la probabilidad condicional de una tripleta en el grafo en base a la siguiente expresión:

$$\Pr(h|r,t) = \frac{\exp\{z(\mathbf{h},\mathbf{r},\mathbf{t})\}}{\sum_{\tilde{h}\in\mathcal{I}} \exp\{z(\tilde{\mathbf{h}},\mathbf{r},\mathbf{t})\}}$$

Figura 13. Probabilidad de observar una tripleta según el modelo conjunto

Donde ***h*** y t se corresponden con las representaciones vectoriales de las entidades o nodos del grafo; ***r*** con las relaciones, y la función z se define como:

$$z(\mathbf{h},\mathbf{r},\mathbf{t}) = b - \tfrac{1}{2}\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|^2$$

Figura 14. Función z para el cálculo de probabilidades en el modelo conjunto

Como podemos observar, el modelo de conocimiento se basa en la intuición de clasificación del modelo de *embeddings* de palabras que vimos en la Figura 7, con la particularidad de que en lugar de considerar el producto escalar de pares de vectores, consideramos el ranking de tripletas acorde a traslaciones de los vectores entidad, lo que concuerda con la idea principal del modelo transE visto anteriormente. Así, se define el coste final de observar una tripleta del siguiente modo:

$$\begin{aligned}\mathcal{L}_f(h,r,t) = \log\Pr(h|r,t) + \log\Pr(t|h,r) \\ + \log\Pr(r|h,t)\end{aligned}$$

Figura 15. Coste de probabilidad de observar una tripleta en el grafo de conocimiento

Lo que da lugar al coste total del modelo de conocimiento, consistente en la maximización de las probabilidades de observar una tripleta:

$$\mathcal{L}_K = \sum_{(h,r,t)\in\Delta} \mathcal{L}_f(h,r,t)$$

Figura 16. Coste total del modelo de conocimiento

Como hemos visto, el cálculo de este coste considerando la expresión de la Figura 13 resulta impracticable en grafos de conocimiento con un gran número de tripletas, por lo que también se emplean métodos de muestreo de tripletas corruptas para su cálculo [6].

Por otro lado, en el modelo textual se realiza la asunción de que, dados dos pares de palabras pertenecientes a la misma ventana, existe una relación no explícita entre dichas palabras. Así, se formula el coste del modelo textual de una manera muy similar al modelo de Skip-Gram:

$$\Pr(w|r_{wv},v) \triangleq \Pr(w|v) = \frac{\exp\{z(\mathbf{w}',\mathbf{v})\}}{\sum_{\tilde{w}\in\mathcal{V}} \exp\{z(\tilde{\mathbf{w}}',\mathbf{v})\}}$$

Figura 17. Probabilidad de observar pares de palabras objetivo y contexto en el modelo conjunto

De este modo, aunque debemos considerar la adaptación de la formulación anterior a su versión optimizada, se define el coste total del modelo textual del siguiente modo:

$$\mathcal{L}_T = \sum_{(w,v)\in\mathcal{C}} n_{wv} \log \Pr(w|v)$$

Figura 18. Coste total del modelo textual

De esta forma, el modelo textual resultante guarda similitudes con el modelo Skip-Gram previamente descrito. Ahora bien, disponiendo únicamente de estos modelos, las computaciones de distancias entre palabras y nodos carecen de sentido. El método propuesto en [6] para alinear ambos tipos de entidades en el mismo espacio consiste en:

- Emplear alineamientos basados en menciones: Normalmente se da el caso de que las entidades del grafo pueden corresponder con una o varias menciones (entidades) textuales. La idea de este tipo de alineamiento consiste en simular el coste del modelo textual por cada par de mención-nodo en lugar de por cada par de menciones:

$$\mathcal{L}_{AA} = \sum_{(w,v)\in\mathcal{C},v\in\mathcal{A}} \log \Pr(w|e_v)$$

Figura 19. Coste total del modelo de alineamiento basado en menciones

  Como podemos imaginar, el efecto de minimizar esta función hará que la distancia entre pares de mención-nodo se reduzca para aquellos casos en los que ambas menciones (la inicial y la que se sustituye por su nodo) coinciden dentro de una ventana definida.
- Emplear alineamiento en base a la creación de tripletas nombradas: Este modo de alineamiento se basa en generar nuevas tripletas sustituyendo entidades de nodos por menciones y añadiendo las nuevas tripletas al grafo:

$$\mathcal{L}_{AN} = \sum_{(h,r,t)\in\Delta} \mathbf{I}_{[w_h\in\mathcal{V}\wedge w_t\in\mathcal{V}]}\cdot\mathcal{L}_f(w_h,r,w_t)+$$
$$\mathbf{I}_{[w_h\in\mathcal{V}]}\cdot\mathcal{L}_f(w_h,r,t)+\mathbf{I}_{[w_t\in\mathcal{V}]}\cdot\mathcal{L}_f(h,r,w_t)$$

Figura 20. Coste total del modelo de alineamientos basado en tripletas nombradas

Finalmente, el coste global de maximización del modelo resulta en:

$$\mathcal{L} = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$$

Figura 21. Coste total del modelo conjunto

Donde el coste del modelo de alineamientos, $\mathrm{L}A$, puede corresponderse con uno de los dos métodos de alineamiento propuestos, o con la suma de los costes de ambos modelos. Para cuya optimización podemos emplear un algoritmo basado en mini-*batches*. Una implementación de este tipo se puede encontrar en github (mglau/pTrasnE), que emplea el algoritmo del descenso de gradiente a tal propósito, y cuyo pseudocódigo se muestra a continuación:

1. Dividir el conjunto de tripletas y pares de palabras objetivo-contexto en varios sub-conjuntos o batches de manera aleatoria.
2. Inicializar los pesos de entidades, relaciones y palabras de manera aleatoria.
3. Por cada batch <TRIPLETAS, PARES CONTEXTO-OBJETIVO>:
    a. Generar un conjunto de tripletas corruptas por cada tripleta del batch.
    b. Generar un conjunto de pares de palabras mediante muestreo de instancias negativas.
    c. Generar un conjunto de tripletas nombradas en las que las entidades quedan representadas mediante los vectores palabra correspondientes a menciones de la entidad del grafo.
    d. Generar un conjunto de pares <entidad objetivo – palabra contexto> empleando las menciones textuales de las entidades del grafo.
    e. Optimizar el modelo acorde a la expresión de la figura 21 tomando los conjuntos a-d como entrada.
4. Repetir 3 hasta completar todos los batches y un número de épocas definido.

Figura 22. Pseudocódigo de optimización del modelo conjunto

2.3. Generación automática de preguntas de hueco en blanco

Dada la gran adopción de preguntas de test como medio de evaluación en diversos niveles de educación, y los esfuerzos requeridos para la creación de este tipo de ejercicios por parte de los instructores, surgen métodos computacionales para la generación automática de tests.

Un tipo específico de preguntas de test son las preguntas de hueco en blanco o FIBQs (Fill-in-the-blank questions). Este tipo de preguntas está compuesto por los siguientes elementos:

(ENUNCIADO) La carbamazepina, en comparación con _____ tiene efectos adversos sobre la médula ósea y puede causar toxicidad fetal provocando defectos en el tubo neural

e) El valproato **(CLAVE)**
f) La cocaína **(Distractor 1)**
g) El LSD **(Distractor 2)**
h) El Valium **(Distractor 3)**

Figura 23. Elementos de una pregunta de hueco en blanco

Aunque es habitual encontrar este tipo de preguntas en ejercicios para el aprendizaje de lenguas extranjeras, también resultan interesantes en otras áreas de las ciencias y la tecnología. En particular, nos centramos en aquellos métodos de generación de FIBQs basadas en las entidades de interés de un dominio concreto. Por tanto, aquellas áreas en las que la topología y relaciones de las entidades conforman un conocimiento valioso en el área, podrán verse beneficiadas por los métodos aquí propuestos. Ejemplos de estas áreas pueden ser la biología, la geología, química o informática, aunque creemos que estos métodos también pueden resultar de interés en pequeñas partes de materias pertenecientes a otras ramas de conocimiento.

Los métodos de generación de FIBQs basados en entidades, generalmente abordan una serie de fases en las que se divide la generación:

1. Identificación de enunciados. En esta fase se identifican aquellas frases o enunciados informativos, o que contienen información considerada como relevante y susceptible de ser preguntada/reconocida.
2. Identificación de claves. En esta fase se identifica una de las posibles entidades contenidas en el enunciado, que se propone como el hueco en blanco y por tanto la respuesta correcta.
3. Propuesta de distractores. Consiste en generar una serie de entidades que actúan como respuestas incorrectas del enunciado, puestas para despistar o distraer al evaluado.

Los principales enfoques empleados para de generación automática de FIBQs son [36] [37]:

- Enfoque ontológico: a este tipo de enfoque pertenecen los métodos que explotan principalmente el conocimiento contenido en una Base o Grafo de Conocimiento con cierto nivel de expresividad. Aunque no es imprescindible, el empleo de ontologías expresivas resulta beneficioso para la extracción de patrones ontológicos que pueden corresponderse con preguntas. Del mismo modo, también existen varias propuestas para la generación de distractores que han sido evaluadas de manera satisfactoria en ámbitos como la lengua y la historia [38] la geografía y la informática [39] [40] así como el médico [41]. Normalmente estos métodos se basan en la definición de plantillas para cada patrón ontológico de preguntas.
- Enfoque textual: a este tipo de enfoques pertenecen los métodos que extraen enunciados desde fuentes textuales [36]. Aunque no necesariamente, estos métodos pueden confiar en fuentes ontológicas o grafos de conocimiento durante las fases de generación. Sin embargo, pueden requerir de una fase añadida a la propuesta de distractores, consistente en garantizar la adecuación léxica y la concordancia gramatical del enunciado con respecto a los distractores.

A pesar de que las fases incluyen una cierta dificultad al proceso, existen otra serie de aspectos que las preguntas generadas deben cumplir [42]:

Tabla 1. Requisitos de las FIBQs para su empleo en escenarios de evaluación

REQUISITOS DE LAS FIBQs	
1	Los enunciados deben ser relevantes, e incluir una clave que no resulte en una pregunta imposible de contestar, ni tampoco evidente.
2	Los distractores propuestos deben añadir dificultad a la pregunta. Distractores demasiado obvios hacen que las preguntas queden invalidadas.
3	Las preguntas deben ser gramaticalmente correctas.
4	Es una capacidad deseable de estos sistemas el poder contar con mecanismos de control de dificultad de las preguntas.

En este trabajo, se emplearán los enfoques textuales apoyados en grafo de conocimiento para la identificación de claves y la propuesta de distractores, por lo que las siguientes secciones estarán dedicadas a mencionar algunas de las técnicas más empleadas para abordar los requisitos 1 y 2 de la tabla anterior. Acabaremos este apartado dedicándole una sección a los modos de evaluación empleados para determinar el cumplimiento de los requisitos.

2.3.1. Identificación de enunciados

Esta es la primera fase del proceso de generación, para la que suponemos que se ha podido llevar a cabo una fase previa de pre-procesado del texto que se empleará como fuente. Durante esta fase, se extraen una serie de frases o conjuntos de frases que contienen información susceptible de convertirse en enunciados. Podemos encontrar distintas estrategias en la literatura para afrontar este objetivo [36]:

- Basadas en la aparición de términos o entidades clave: algunas heurísticas empleadas son la aparición de determinados n-gramas o la cantidad/riqueza terminológica incluida en las frases. En [43], se diseña un conjunto de reglas basadas en conteos de palabras, nombres y presencia de superlativos,
- Basadas en información lingüística: es posible emplear técnicas de *shallow parsing* para la extracción de patrones sintácticos que conllevan la recuperación de frases afirmativas. Por otra parte, también podemos encontrar trabajos que explotan marcos semánticos para el etiquetado de entidades acorde a los distintos roles contemplados en dichos marcos. Los autores de [44] proponen un sistema de identificación de frases informativas basado en comparación de árboles sintácticos, de manera que se extraen dichos árboles partiendo de un conjunto existente de preguntas, y seguidamente comparan nuevas estructuras frente a este conjunto de referencia para decidir cuáles recuperar de una fuente textual. Un método similar es empleado en [45], en el que primeramente se extraen roles semánticos de un conjunto de preguntas existentes y posteriormente las nuevas preguntas son comparadas con patrones tanto semánticos como sintácticos gracias al empleo de etiquetadores de árboles sintácticos. La información semántica de un conjunto de enunciados preestablecido es explotada en [46] empleando un modelo de *embeddings* de frases basado en un modelo bi-LSTM (Long Short Term Memory), de manera que las frases que superan un umbral de similitud son seleccionadas como informativas.
- Basadas en sumarización: Algunos trabajos se basan en el empleo de un sumarizador de texto para la extracción de las frases más significativas de un texto. En [47] emplean un sumarizador basado en conteos de palabras para identificar las frases clave en fragmentos de texto recopilados desde Wikipedia.
- Basadas en aprendizaje automático: Cuando se dispone de un conjunto de frases etiquetadas, es posible emplear técnicas de aprendizaje automático para inferir nuevas cláusulas.

Todas estas técnicas tienen sus ventajas e inconvenientes. En primer lugar, las reglas heurísticas pueden dar lugar a enunciados excesivamente amplios y perder coherencia, mientras que las técnicas basadas en información lingüística pueden dar lugar a sentencias demasiado cortas [36]. Por otra parte, las técnicas basadas en sumarización pueden constituir un punto intermedio entre los métodos anteriores, aunque su calidad depende en gran medida del éxito de medidas de similitud de frases. Finalmente, el enfoque basado en aprendizaje automático puede resultar el más exitoso, aunque aparte de requerir de un conjunto de frases etiquetadas (como informativas o no) también depende del tipo de características que se extraen.

2.3.2. Identificación de claves

Una vez que se dispone de un conjunto de enunciados, se identifica una palabra, posiblemente multi-término, para considerarla como respuesta válida y por tanto identificar el hueco en blanco. Los métodos empleados durante esta fase pueden variar, al igual que los métodos de la fase anterior.

Por un lado, encontramos trabajos que realizan una identificación de la clave en base a la aparición de palabras clave, o que bien realizan algún tipo de ranking basado en conteos y frecuencias de aparición como TF-IDF.

La información lingüística también puede ser empleada para identificar claves acorde a ciertos patrones sintácticos o semánticos. Entre estos métodos, destacamos los basados en entidades nombradas, o aquellos que restringen el conjunto de claves en base a información de la entidad en un grafo de conocimiento u ontología. En [42], se construyen una serie de reglas en base a las etiquetas gramaticales de las palabras del enunciado para crear un conjunto inicial de candidatos del que finalmente se extrae aquel candidato que se encuentre lo más cerca posible de la raíz del árbol sintáctico de la frase.

Una mezcla de características de frecuencia, así como léxicas (abreviaciones, mayúsculas, *stop-words*), sintácticas (etiqueta gramatical), semánticas (Etiquetado de roles semánticos) y estructuradas (enlaces de Wikipedia) es explotada para identificar los huecos sobre preguntas obtenidas de artículos de Wikidata [47].

Finalmente, del mismo modo que podemos emplear algoritmos de Aprendizaje Automático para identificar enunciados, podremos explotar conjuntos de enunciados con la clave etiquetada para inferir nuevas claves en nuevos enunciados.

2.3.3. Propuesta de distractores

Una vez identificado el enunciado y la clave, en esta fase se generan los distractores que formarán las respuestas incorrectas de las preguntas. De entre los métodos propuestos para abordar esta tarea destacan aquellos:

- Basados en hipótesis distribucional: como el empleo de medidas de similitud basadas en *embeddings* de palabras. De esta forma, los distractores propuestos guardan cierta medida de similitud semántica con la clave. En [48], se parte de un conjunto de enunciados y claves identificadas manualmente en el ámbito de las ciencias, posteriormente, se generan modelos de similitud de palabras basados en técnicas de Análisis Semántico Latente (LSA, Latent Semantic Analysis) que son empleados para la generación de distractores candidatos que son filtrados acorde a distintas políticas como la similitud morfológica con la clave o el filtrado de distractores empleando una Base de Conocimiento sobre la que se aplica el algoritmo Page-Rank para obtener finalmente las entidades relevantes .
- Basados en conocimiento estructurado: esos métodos explotan los datos albergados en lexicones, grafos de conocimiento, ontologías u otras estructuras de datos. Una práctica habitual consiste en emplear como distractores aquellas entidades que pertenecen a un mismo grupo de sinónimos o a un mismo tipo de entidades que la clave. En [44] llevan a cabo un etiquetado de entidades nombradas del texto y se seleccionan aquellos distractores que pertenezcan a la misma clase que la clave.
- Basados en patrones lingüísticos: consideran como distractores aquellas palabras con mismo rol semántico o sintáctico que la clave identificada.

2.3.4. Evaluación de sistemas de generación de FIBQs

La evaluación de estos sistemas puede realizarse de manera desacoplada de las tareas posteriores del proceso de generación, aunque lo normal es evaluar las preguntas de manera extrínseca por parte de los expertos instructores del dominio de aplicación. De entre los rasgos sujetos a evaluación, podemos considerar:

- La evaluación de la relevancia de los enunciados extraídos.
- La evaluación de la adecuación de la clave escogida.
- La evaluación de la adecuación de los distractores generados.
- La evaluación sobre consistencia léxico-gramatical de las preguntas generadas.
- La evaluación sobre la dificultad de las preguntas generadas.

Por ejemplo, en [45], lo autores llevan a cabo una evaluación de su sistema en el ámbito biomédico mediante la presentación de formularios al experto instructor. En su trabajo, se centran en evaluar consistencia gramatical, relevancia de las preguntas, aceptación de distractores y utilidad global de las preguntas. El formulario se diseña con escalas discretas de 0 a 5 para puntuar las preguntas. Los resultados muestran una buena aceptación global de media (3.37 sobre 5) aunque se dan problemas gramaticales en la pregunta y los distractores, y la relevancia de las preguntas no logra la puntuación media.

Una evaluación exclusiva de la calidad de los distractores se lleva a cabo en [48], en la que se realiza una evaluación empírica de una serie de preguntas previamente etiquetadas (con su clave) en un contexto real en el dominio de ciencias y en el nivel de Educación Secundaria. La calidad de los distractores queda entonces evaluada en base a las características de dificultad (proporción de estudiantes que contestaron adecuadamente) y discriminación (los distractores son efectivos si las preguntas difíciles son contestadas correctamente por los estudiantes de mayor rendimiento). Los resultados muestran que una relativamente pequeña cantidad de las preguntas, en torno a un 20%, tienen una dificultad baja (el 90% o más de los alumnos la han acertado), aunque el número de preguntas difíciles no sobrepasa el 15% (menos del 30% de estudiantes la han acertado).

En [49] generan de manera completamente manual, un conjunto de 76 preguntas y sus distractores, las cuales son sometidas a evaluación por parte de 5 expertos para examinar la validez de las preguntas, así como de la clave y el enunciado. Los resultados muestran que un 84 % de las preguntas resultan válidas para su aplicación en el dominio de historia y nivel educativo de primaria.

EL trabajo en [50] muestra los resultados de evaluación de un sistema de generación de FIBQs aplicada al dominio de biología en cuanto a validez de las preguntas y distractores. Para ello, dos evaluadores llevan a cabo las mismas evaluaciones en paralelo, identificándose un alto porcentaje de claves, así como porcentajes considerables de preguntas válidas (en torno al 75%) y un conjunto de distractores válidos mediocre (sobre 60% y 67% dependiendo del evaluador).

3. Generación del Grafo de Conocimiento

3.1. Metodología

La construcción de un Grafo de Conocimiento se puede entender como la construcción de una Base de Conocimiento. Es decir, un proceso de elicitación y de ingeniería, en el que el conocimiento experto de un área determinada queda codificado mediante un modelo que puede requerir del empleo de distintos formalismos de Representación del Conocimiento para su adecuada codificación. Así, desde la perspectiva de los Sistemas Expertos y la Ingeniería del Conocimiento, es común comenzar planteándose una serie de preguntas de competencia, con las cuales determinar el alcance en base a aquellas preguntas que se espera pueda contestar dicho artefacto.

En nuestro caso, deseamos reducir esta carga de trabajo inicial, optando por la automatización en la medida de lo posible de la generación del Grafo de Conocimiento gracias al empleo de las

tecnologías PLN, y sobre todo de las fuentes de datos disponibles en la Web tradicional y en la Web Semántica. Sin embargo, es necesario que el experto del área, en este caso el docente, tenga una participación directa en la fase inicial de la generación del grafo para asegurar que esta queda enfocada de manera adecuada al dominio de aplicación, es decir, para establecer un mínimo *schema* o vocabulario. Para ello, se ha seguido una metodología ágil basada en [14] considerando la existencia de un rol de Ingeniero en Web Semántica, proponiendo la realización de tres entrevistas básicas con el experto para acordar los grandes rasgos del modelo semántico del Grafo de Conocimiento. En concreto, se han determinado una serie de fases clave en esta etapa inicial:

1. Primera entrevista con el personal docente. Análisis de los principales grandes tipos de entidades de interés de la asignatura y discusión de posibles aplicaciones. El personal docente establece una serie de ítems o entidades de interés de su asignatura.
2. Estudio de fuentes de datos relevantes por parte del ingeniero. Se estudian minuciosamente las fuentes de datos estructurados o semi-estructurados que se pueden emplear para el poblado de la Base de Conocimiento.
3. Segunda entrevista con el personal docente para una primera aproximación formal al dominio. Contraste de ideas con el personal docente para acordar un modelo semántico inicial mediante mapas conceptuales o esquemas. Además, en esta entrevista se le ofrece al experto un catálogo de datos para que decida su utilidad o interés.
4. El ingeniero diseña un flujo de recuperación de datos acorde al modelo semántico y las fuentes de datos identificadas como relevantes. Se genera el Grafo de Conocimiento.
5. Tercera entrevista para conocer el alcance de los datos que se han podido recopilar. Se analiza la posible utilidad de los mismos y se muestra el modelo semántico final en base a la disponibilidad final de los datos.

El proceso de generación requiere de un diseño preciso de flujo de recuperación de datos para poder ofrecer una primera versión de la Base de Conocimiento lo más completa posible, así como el empleo de herramientas de Recuperación de la Información facilitadas por los sistemas que albergan los conjuntos de datos que se contemplaron. Para ello, emplearemos un método similar a [18], con la peculiaridad de que en una primera etapa, confiaremos en los Grafos de Conocimiento disponibles para la identificación automática de entidades sobre el texto, siguiendo un enfoque similar al presentado en [16].

Los siguientes apartados reflejan la adopción de la metodología propuesta y detallan el proceso de recuperación.

3.2. Creación del modelo semántico inicial

En primer lugar, se identificó un conjunto inicial de tipos de entidades bajo la guía del personal docente de la asignatura. Como se ha indicado previamente, se llevó a cabo una primera entrevista para conocer el propósito de la asignatura y los principales tipos de entidades de interés. Durante esta fase se recopilaron las siguientes entidades como relevantes en el contexto de la asignatura de Psicofarmacología:

Tabla 2. Entidades de interés tras primera entrevista

TIPO DE ENTIDAD
Fármacos
Receptores
Acciones farmacológicas
Trastornos psicológicos

Tratamientos recomendados
Efectos adversos o nocivos
Regiones o partes del cerebro
Funciones (ej. cognitivas)

De cara a la segunda entrevista, el Ingeniero Semántico localizó varias fuentes de datos y modelos semánticos existentes que cubrieran las necesidades de datos en la mayor medida posible. De entre las fuentes identificadas, que pasarían a formar parte del catálogo de datos, se encuentran las siguientes:

Tabla 3. Fuente de datos: Wikidata

WIKIDATA			
El propósito de Wikidata es facilitar una Base de Conocimiento generalista editable por humanos y por sistemas. Esta contiene conocimiento sobre entidades de distinta índole, lo que lo convierte en una herramienta muy potente para el soporte en tareas de identificación o enlazado de entidades. En un principio, toda entrada en Wikipedia dispone de su representación en el grafo de conocimiento.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
GENERALES	RDF / SPARQL	LIBRE (CC)	WIKIMEDIA FOUNDATION

Tabla 4. Fuente de datos: UMLS

UMLS			
El propósito de UMLS (Unified Medical Language System) es facilitar una serie de sistema terminológicos y computacionales para dar soporte a la interoperabilidad entre sistemas y servicios relacionados con el ámbito de la bioinformática. De esta forma, UMLS facilita servicios de búsqueda e identificación de entidades llevando a cabo una consolidación de conceptos en base a los términos referidos a dichos conceptos en distintas bases de datos externas. UMLS emplea a tal propósito un modelo semántico para la caracterización de los conceptos y sus posibles relaciones.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
BIOINFORMÁTICA	API REST / JSON	PARTICULAR (NO COMERCIAL)	NATIONAL INSTITUTE OF HEALTH

Tabla 5. Fuente de datos: DrugBank

DRUGBANK			
El propósito de Drugbank es la creación de una base de datos que recopile información detallada sobre drogas o fármacos, con el objetivo de favorecer aplicaciones del ámbito de la bioinformática como la identificación de interacciones entre drogas o el descubrimiento de nuevos fármacos.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
DROGA, ACCIÓN FARMACOLÓGICA	BBDD en XML	LIBRE (CC)	ORIGINARIAMENTE, CANADIAN

			INSTITUTES OF HEALTH RESEARCH

Tabla 6. Fuente de datos: PubChem

PUBCHEM			
Pubchem agrupa una serie de información detallada sobre distintos tipos de moléculas y su composición. Sin embargo, este repositorio realiza una recopilación de información de fuentes externas para relacionar otros tipos de entidades con los compuestos químicos como enfermedades asociadas, productos farmacológicos, o referencias a la literatura.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
PRINCIPALMENTE MOLÉCULAS, ESTRUCTURAS + ENFERMEDADES, PATENTES, REFERENCIAS	REST API / JSON	LIBRE	NATIONAL INSTITUTE OF HEALTH

Tabla 7. Fuente de datos: DisGeNet

DISGENET			
El propósito de este Grafo de Conocimiento es facilitar el descubrimiento de relaciones entre genes y enfermedades. Para lo cual, esta fuente de datos lleva a cabo una consolidación de datos curados disponibles en otras fuentes de datos externas y lleva a cabo tareas de predicción para inferir nuevos enlaces entre las entidades.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
GEN ENFERMEDAD	REST API / SPARQL / RDF / JSON	NO COMERCIAL	CONSORCIO DE GRUPOS DE INVESTIGACIÓN

Tabla 8. Fuente de datos: CTD

CTD			
El propósito de CTD (Comparative Toxicogenomics Database) es facilitar una base de datos que contiene información relevante para obtener conclusiones sobre aquellos factores o exposiciones que afectan a la salud de los humanos. Esta base de datos contiene una serie de relaciones curadas entre distintos tipos de entidades.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
DROGA, GEN, ENFERMEDAD, FENOTIPOS, EXPOSICIONES	PROGRAMÁTICO MEDIANTE FORMULARIOS, CSV, XML, TSV	PARTICULAR (NO COMERCIAL)	MDI BIOLOGICAL LABORATORY

Tabla 9. Fuente de datos: MeSH

MESH

El propósito de MeSH (Medical Subject Headings) es facilitar un tesauro de conceptos y términos del ámbito de la biomedicina que es empleado para organizar la literatura del campo.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
BIOINFORMÁTICA, BIOMÉDICA	SPARQL / RDF / REST API / XML	PARTICULAR (NO COMERCIAL)	NATIONAL INSTITUTE OF HEALTH

Tabla 10. Fuente de datos: UniProt

UNIPROT El propósito de Uniprot es facilitar una base de datos sobre secuencias de proteínas y ciertos aspectos de las mismas, como sus funciones, composición o tipología.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
PROTEÍNAS, GENES, INFORMACIÓN COMPOSICIONAL Y FUNCIONAL	REST API / JSON / SPARQL	LIBRE (SUJETA A ATRIBUCIÓN)	EMBL-EBI, SIB SWISS INSTITUTE OF BIOINFORMATICS, GEORGETOWN UNIVERSITY

Tabla 11. Human Disease Ontology

DO El propósito de Disease Ontology (DO) es el desarrollo de una ontología curada y estandarizada sobre enfermedades del ser humano.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
ENFERMEDADES	RDF-XML / OWL /OBO	LIBRE	UNIVERSITY OF MARYLAND

Tabla 12. Symptom Ontology

SO El propósito de Symptoms Ontology (SO) es facilitar un conjunto de datos estructurados sobre las relaciones entre enfermedades y observaciones clínicas, así como síntomas.			
TIPOS DE ENTIDAD	TIPO DE ACCESO	TIPO DE LICENCIA	ORGANIZACIÓN
OBSERVACIÓN, SÍNTOMA, ENFERMEDAD	RDF-XML / OWL /OBO	LIBRE	OBO FOUNDRY

Por otra parte, de cara a facilitar un modelo semántico inicial, se barajaron los siguientes vocabularios y taxonomías:

- Vocabularios empleados en el portal BIO2RDF: Este portal de datos enlazados alberga una gran cantidad de datos de distintas fuentes del ámbito de la bioingeniería y la bioinformática. Su principal problema es que lleva sin actualizarse desde la versión 3 en 2014.

- Árbol conceptual de MESH (Medical Subject Headings): Este tesauro de conceptos contempla un glosario de términos médicos muy extenso en forma de taxonomía que resulta útil para conocer las familias de los tipos de entidades propuestas (subtipos y supertipos de fármacos, receptores, etc.)
- Modelo semántico de UMLS (Unified Medical Language System): una parte del sistema de UMLS lo conforma un modelo semántico de tipos y relaciones posibles entre entidades de dichos tipos, estableciendo un vocabulario en el que los dominios y los rangos de las propiedades están definidos.

Por los motivos expuestos, durante la segunda entrevista, ofrecimos al personal docente un elenco de posibles tipos de entidades y relaciones que podrían ser de interés basándonos en una simplificación del modelo semántico de UMLS acorde a las entidades de la Tabla 2. El modelo semántico o *schema* acordado queda representado en el siguiente diagrama:

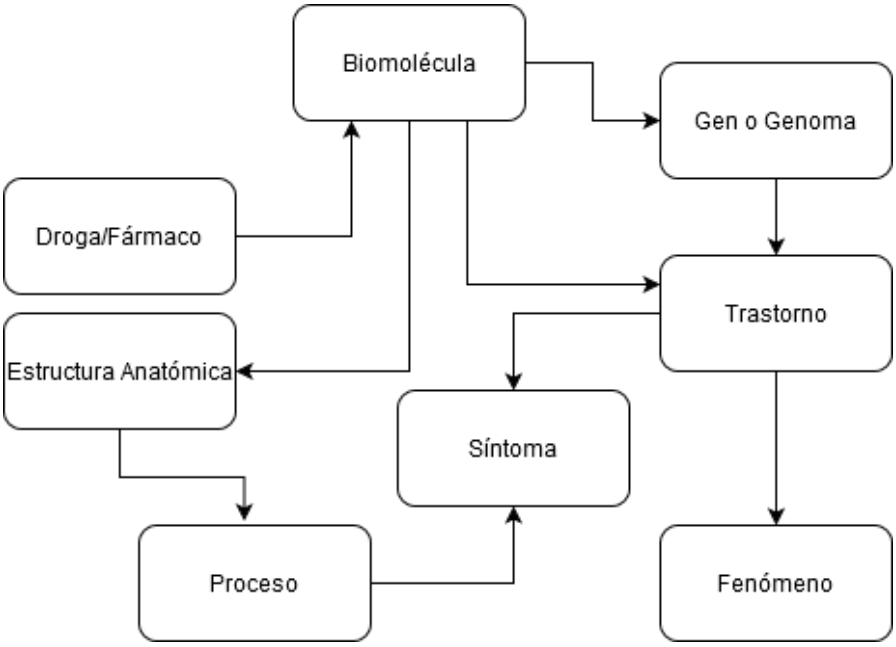


Figura 24. Modelo semántico inicial acordado

A este punto, se diseñó el flujo de recuperación del Grafo de Conocimiento y el personal docente quedó a la espera de los resultados. Los detalles de este flujo quedan descritos en los siguientes apartados.

3.3. Pre-procesado de fuentes textuales

Esta es la primera fase de cara a la generación del Grafo de Conocimiento y consiste en disponer de los recursos textuales de la asignatura con sus palabras separadas en *tokens* y etiquetadas acorde a su clase gramatical o *part-of-speech* con el empleo de herramientas de PLN.

Recordemos que en el contexto de la asignatura disponemos del libro recomendado y un banco de preguntas de test empleadas durante la evaluación oficial de la asignatura. Inicialmente, se dispone del libro en formato electrónico PDF y las preguntas quedan recopiladas en un documento Word. Por tanto, para disponer de texto plano, es necesario llevar a cabo una fase de pre-procesado, en la que el texto se ha extraído del libro con la librería Python PDFMiner, y un *parser* programado para distinguir entre las distintas secciones de texto empleando la librería Python pytransitions. De este modo, obtenemos capítulos, encabezados a distintos niveles, e información de si el texto pertenece a la descripción de una figura o tabla.

En segundo lugar, cada una de las secciones de texto se segmenta en una secuencia de frases y *tokens* con la librería Stanza que también es empleada para obtener el texto etiquetado acorde a las clases gramaticales contempladas en este sistema. Para cada uno de los *tokens*, almacenamos también información sobre el lema de la palabra, su género y su número.

En cuanto a las preguntas de test de años anteriores, estas se incluyen en un fichero de texto plano y se etiquetan del mismo modo con la librería Stanza.

3.4. Búsqueda de entidades

Esta fase tiene como objetivo la extracción de una serie de entidades nombradas relevantes desde el texto. En contraposición a los enfoques más puramente basados en Minería de Textos y Recuperación de Información, nosotros empleamos un método basado en diccionario que se apoya en las herramientas de búsqueda existentes de las bases de datos identificadas, por lo que primeramente realizamos una búsqueda de términos candidatos que someteremos a una búsqueda web para la resolución y enlazado de entidades.

Los términos candidatos se corresponden con aquellas secuencias de *tokens* extraídas mediante técnicas de análisis sintáctico superficial teniendo en cuenta la siguiente gramática, para la que se obtiene un conjunto inicial de 16.064 términos candidatos:

```
BaseNP:   {<NOUN|PROPN><ADJ|NOUN|PROPN|NUM>*}

PrepPhrase: {<ADP><DET>*<BaseNP>}

TERM:   {<BaseNP><PrepPhrase>*}
```

Figura 25. Gramática para la identificación de términos candidatos

Por cada uno de los 16.064 términos candidatos, extraídos empleando la librería NLTK, se automatizan una serie de búsquedas frente a las bases de datos más generalistas de nuestro catálogo de datos (UMLS y Wikidata). La política que se ha seguido para la resolución de entidades (de cuando considerar una entidad por reconocida) una vez disponemos de los términos candidatos, es la siguiente:

- Búsqueda con UMLS: se realiza una búsqueda terminológica a través del servicio terminológico ofrecido a través del API REST de UMLS. Si se encuentran resultados empleando la búsqueda exacta, se toma como entidad el primero de los resultados.
- Búsqueda con Wikipedia/Wikidata: se realiza una búsqueda por título de página de Wikipedia. Si se encuentra un resultado empleando la búsqueda exacta, se toma como entidad, de lo contrario:
    - Si el término contiene información del tipo (“receptor”, “proteína”, “trastorno”, etc.): se realiza una búsqueda personalizada de entidades por tipo empleando el API de búsqueda de Wikidata. Se toma como entidad la entidad correspondiente a la etiqueta más similar empleando la distancia Levenshtein, siempre y cuando sea menor de 3 caracteres para palabras mayores de 6 caracteres.
    - Si el término no contiene información del tipo: se realiza el mismo tipo de búsqueda de Wikidata, pero sin incluir la restricción de tipo.

El flujo de búsqueda se ha desarrollado empleando la librearía Scrapy. Con el empleo de esta librería, generamos una serie de documentos JSON que dan forma a un conjunto de datos

intermedio, definida mediante la librería tinyDB, que incluye información básica sobre cada una de las entidades reconocidas (su tipo en el grafo de conocimiento del que se ha recuperado, identificadores del elemento en fuentes externas, y etiquetas):


```
{
  "word_form": "monoamina",
  "prov_spider": "wikipedia_search_spider",
  "title": "Tiramina",
  "id": "http:\/\/www.wikidata.org\/entity\/Q165930",
  "labels": [
    {
      "lang": "en",
      "label": "2-(4-Hydroxyphenyl)ethylamine"
    },
    {
      "lang": "en",
      "label": "2-(4'-Hydroxyphenyl)ethylamine"
    }
  ],
  "types": [
    {
      "label": "heteroatomic molecular entity",
      "uri": "http:\/\/www.wikidata.org\/entity\/Q11173",
      "lang": "en"
    },
    {
      "label": "compuesto quimico",
      "uri": "http:\/\/www.wikidata.org\/entity\/Q11173",
      "lang": "es"
    }
  ],
  "external_ids": [
    {
      "id": "CHEMBL11608",
      "source": "ChEMBL ID"
    },
    {
      "id": "2150",
      "source": "Guide to Pharmacology Ligand ID"
    }
  ]
}
```


Figura 26. Ejemplo de JSON simplificado para almacenar los datos sobre entidades identificadas

El tiempo de ejecución para el total de 16.064 términos fue de 3 horas y 20 minutos para el caso de la búsqueda con UMLS, y de casi dos horas para el caso de la búsqueda con WIkidata. Esto es debido a las políticas de empleo del API REST de UMLS, que emplean fuertes restricciones de peticiones simultáneas, así como el método de control de licencia y clave, que conlleva el empleo de una petición HTTP adicional por cada ítem que se desea recopilar.

A este punto, se dispone de un conjunto de términos y sus entidades asociadas que no necesariamente son relevantes, sobre todo debido al empleo de Wikipedia para la resolución de entidades, puesto que UMLS está más especializado. Por tanto, es necesario realizar un filtrado de las entidades, para lo cual emplearemos un mapeo de los tipos del modelo de datos de Wikidata y el modelo semántico de UMLS a los tipos incluidos en nuestro vocabulario acordado con el docente.

3.5. Filtrado de entidades y búsqueda de relaciones

Partiendo del conjunto inicial de términos y entidades, en esta fase se filtran aquellas entidades que no corresponden a los tipos incluidos identificados en el modelo semántico inicial. Este filtrado se realiza por tipos, teniendo en cuenta la correspondencia de tipos que se detalla a continuación:

Tabla 13. Mapeo de tipos del modelo semántico acordado y las fuentes de búsqueda

CLASES DEL MODELO ACORDADO	TIPOS UMLS ASOCIADOS	TIPOS WIKIDATA ASOCIADOS
Droga/Fármaco	Pharmacologic Substance	Drug

	Hazardous or Poisonous Substance	Pharmaceutical Drug Medication Medicine Pharmaceutical Product Prohibited Substance Antidote
Biomolécula	Amino Acid, Peptide, or Protein Receptor Hormone	Protein Protein Family Group or Class of Proteins Enzyme
Síntoma	Pathologic FUnction Sign or Symptom Finding Individual Behavior Cell or Molecular Dysfunction	Health Problem Clinical Finding Physiological Condition
Trastorno	Disease or Syndrome Mental or Behavioral Dysfunction Neoplastic Process Anatomical Abnormality Congenital Abnormality	Disease Illness Syndrome Disorder Rare Disease
Proceso	Mental Process	Emotion Biological Process
Fenómeno	Phenomenon or Process	Disease Attributes
Estructura	Body Part, Organ or Organ Component Cell Component Cell Tissue	Anatomical Structure Organ System Organ Part Cell Cell Type Brain Region Tissue

Si una entidad pertenece a alguno de estos tipos, esta pasará a ser empleada por parte del flujo de recuperación de relaciones de las fuentes identificadas, recorriendo el camino adecuado dependiendo del tipo identificado. De nuevo, se empleó la librería Scrapy para automatizar la extracción de datos partiendo de los identificadores disponibles. Además, una vez recuperada la estructura de datos JSON de cada entidad con sus propiedades, se realiza una conversión a RDF empleando la librería RdfLib de Python. Las tripletas generadas acorde al modelo semántico propuesto finalmente se cargan en un almacén de RDF (Apache Jena Fuseki). El diseño final del flujo se realiza empleando la disponibilidad de identificadores externos para cada uno de los conjuntos de datos listados en el catálogo.

El tiempo requerido por esta fase ha sido de 44 minutos para las entidades identificadas con el empleo de Wikidata y de 3 horas y cuarto para las entidades identificadas con el empleo de UMLS.

3.6. Consolidación de la Base de Conocimiento

La Base de Conocimiento inicialmente no contempla las propiedades finales, sino que estas propiedades quedan reflejadas mediante el enlace con identificadores textuales de las entidades que se enlazan (valores literales RDF). Por tanto, para poder expresar adecuadamente las propiedades, se deben consolidar los identificadores con las URIs de los recursos.

Para ello, se dispone de una batería de consultas SPARQL que generan las propiedades y borran los valores literales posteriormente. Gracias a los identificadores externos de cada entidad, también seremos capaces de identificar aquellas entidades que reflejan el mismo concepto y generar un enlace del tipo owl:sameAs. Un ejemplo de consulta SPARQL de creación de propiedades es el siguiente:

```
PREFIX fp: <http://data.ia.uned.es/PF/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

INSERT {
  ?a fp:interviene ?d
}
WHERE{
   ?a a fp:Biomolecula;
       fp:interviene ?id.
   ?d a fp:Trastorno;
      ?prop ?id.
  FILTER(CONTAINS(STR(?prop),"_id"))
}
```

Figura 27. Consulta SPARQL INSERT

Esta consulta crea enlaces del tipo “interviene” para reflejar que existe una cierta intervención de un tipo de biomoléculas en el desarrollo de una enfermedad. La consulta genera un enlace entre el recurso biomolécula y el recurso cuyo identificador coincide con el valor literal de identificador asociado a la misma propiedad. Por tanto, es necesario eliminar los valores literales anteriores para eliminar aquellas propiedades que no enlazan con alguna de las entidades que hayamos recopilado:

```
PREFIX fp: <http://data.ia.uned.es/PF/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

DELETE {
  ?a fp:interviene ?id
}
WHERE{
   ?a a fp:Biomolecula;
       fp:interviene ?id.
  FILTER(ISLITERAL(?id))
}
```

Figura 28. Consulta SPARQL DELETE

Existen un total de 8 consultas tipo INSERT y 6 de tipo DELETE. De entre las consultas de tipo INSERT también cabe destacar la empleada para la generación de enlaces del tipo owl:sameAs cuando dos entidades comparten algún identificador.

3.7. Resultados de la generación y modelo final

Seguidamente se facilita información cuantitativa sobre el grafo de conocimiento generado:

Tabla 14. Número de entidades por tipo del grafo generado

Tipo de entidad	Número de entidades
Biomolécula	399
Síntoma	214
Droga	204
Trastorno	180
Gen	117
Proceso	80
Estructura	74
Fenómeno	8

Tabla 15. Número de enlaces por propiedad del grafo generado

Tipo de relación	Dominio-Rango	Número de enlaces
gen_asociado	Síntoma-Gen	2381
proceso	Biomolécula-Anotación Molecular	2239
contribuye	Droga-Trastorno	1454
componente	Biomolécula-Anotación Molecular	1097
trata	Droga-Trastorno	1092
función	Biomolécula-Anotación Molecular	868
sintomatología	Trastorno-Síntoma	568
sustrato	Droga-Biomolécula	331
interviene	Biomolécula-Gen	157
antagnoista	Droga-Biomolécula	155
gen	Biomolécula-Gen	129
agonista	Droga-Biomolécula	64
inductor	Droga-Biomolécula	30
agonista_parcial	Droga-Biomolécula	10

Debemos tener en cuenta que este grafo contiene cierto ruido, sobre todo debido al método de identificación de entidades. Sin embargo, pensamos que la calidad de las propiedades es alta, debido al empleo de conjuntos de datos manualmente curados durante la extracción.

Finalmente, en una tercera entrevista, le mostramos los resultados al experto docente para, en primer lugar, hacernos una idea de aquellos datos del modelo que han podido ser recopilados. En concreto, el siguiente diagrama muestra aquellas partes del esquema para las que se dispone de datos y aquellas para las que no ha podido ser posible empleando el catálogo de datos establecido:

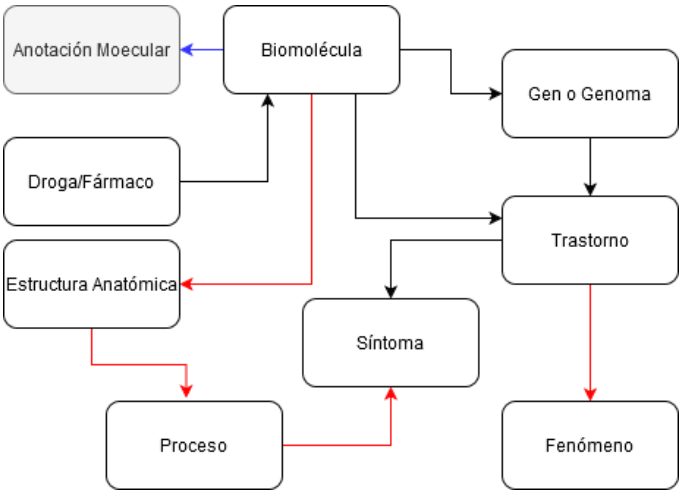


Figura 29. Cobertura final del grafo de conocimiento generado

Como podemos observar en la figura anterior, existe un gran número de enlaces (en rojo) para los cuales no se ha podido recopilar relaciones partiendo de nuestro catálogo de datos. Además, hemos llevado a cabo la recopilación de un tipo adicional de entidad, correspondiente con las anotaciones que se facilitan en Uniprot sobre componentes y funciones moleculares, dado que se identificó este tipo de entidad durante el proceso de recopilado pero no fue contemplado durante la segunda entrevista.

En la última y tercera entrevista, se muestran los resultados cuantitativos y se barajan posibles aplicaciones futuras con el docente, de entre las que se han identificado como valiosas:

- La generación de preguntas de hueco en blanco.
- La disposición de un interfaz de navegación de los contenidos de la asignatura.
- La disposición de un interfaz de consulta para recuperar datos desde el grafo de conocimiento.

En este trabajo, abordaremos la generación de preguntas de hueco en blanco, para lo que aprovecharemos el conocimiento extraído. Sobre esta aplicación, el experto docente señaló el subconjunto de tipos de entidades que estarían sujetas a este tipo de aplicación, resultando en las 7 siguientes:

- Droga
- Biomolécula
- Síntoma
- Trastorno
- Proceso
- Fenómeno
- Estructura

4. Generación de modelos de *embeddings*

En este apartado se describen los modelos de *embeddings* generados para la representación vectorial de entidades como palabras y elementos del grafo. En concreto, deseamos emplear las representaciones aprendidas por estos modelos para el empleo del cálculo de similitud en entidades, que formará nuestro método de generación de distractores a la hora de generar preguntas de hueco en blanco en el siguiente capítulo.

Por tanto, detallamos a continuación el proceso llevado a cabo para la generación de modelos de aprendizaje de características, en los que consideraremos el modelo Skip-Gram de word2vec [4], un modelo traslacional transE de *embeddings* de grafo de conocimiento [5], así como un modelo de optimización conjunta de palabras y elementos del grafo [6].

Principal atención tiene el análisis de la bondad de las representaciones, sobre todo en relación con una determinada configuración o valores de los hiperparámetros de los modelos. Para llevar a cabo este ajuste, así como el análisis de las representaciones, nos apoyaremos en técnicas de visualización, así como modos de evaluación intrínseca de los modelos atendiendo a trabajos [24].

Si bien la evaluación intrínseca de los modelos de *embeddings* se puede realizar en tarea de predicción, los autores de [51] sugieren el empleo de:

- Evaluación en la tarea de analogía. Para lo cual se debería disponer de un conjunto curado de analogías que nosotros no disponemos. Además, resulta complicado realizar una evaluación de este tipo cuando la naturaleza de los datos hace posible que se asocie más de una entidad por cada analogía.
- Evaluación en tarea de *clustering*. La idea es medir el poder de las representaciones vectoriales para caracterizar los distintos tipos de entidades (palabras o nodos). Los autores sugieren emplear el algoritmo de agrupación k-medias estableciendo el número de grupos al número de tipos distintos de entidades que se consideran. Trataremos de llevar a cabo esta tarea para categorizar las entidades teniendo en cuenta los tipos de las mismas en el grafo de conocimiento generado en el apartado 3.
- Evaluación en tarea de similitud. En [52] Podemos encontrar un caso de evaluación intrínseca basado en etiquetado manual de palabras similares. De esta forma, se generarían palabras similares a una dada en base a distintos modelos y el experto iría decidiendo cuáles de las palabras propuestas es más similar a la dada. Lo habitual es generar tres tipos de comparaciones por cada entidad dada acorde a distintos rangos (ej. la más similar, la tercera más similar y la sexta más similar). Para ello, diseñaremos una evaluación manual que será revisada por el experto docente.

4.1. Generación de los modelos

En este apartado detallamos diversas configuraciones de los modelos de aprendizaje de características. El objetivo es poder hacernos una idea sobre aquellos parámetros que resultan adecuados para el entrenamiento de los modelos.

4.1.1. Skip-Gram

Para poder entrenar este modelo, inicialmente partimos del conjunto de *tokens* o palabras procesadas de los recursos textuales de la asignatura. Para este conjunto de palabras, se han agrupado en un solo *token* aquellos términos multi-palabra correspondientes con las entidades extraídas durante la generación del gafo de conocimiento. Así, tenemos una referencia consistente de la aparición de las entidades de interés. De esta forma, el tamaño del vocabulario empleado es de 8732 palabras, previo filtrado de palabras según categorías gramaticales (se descartan determinantes, adjetivos, puntuaciones, conjunciones, verbos auxiliares y pronombres). A pesar de disponer de palabras que no se corresponden con ninguno de los tipos identificados en el grafo de conocimiento, estas se emplearán durante el entrenamiento, mientas que mostraremos resultados para aquellas representaciones de palabras que contengan una entrada en el Grafo de Conocimiento generado en el apartado 3.

En primer lugar, con una configuración dada, realizamos un estudio del coste total por cada iteración, para un total de 15 iteraciones. Se observó que pasadas las 4 primeras épocas, el coste comenzaba a subir, por lo que se ha decidido emplear un máximo de 5 iteraciones o épocas.

Generaremos un total de 15 modelos de *embeddings* de palabras empleando una implementación del algoritmo de SKip-Gram descrito en el apartado 2.2.1. Para ello emplearemos la librería Gensim con *sampling* de muestras negativas y los siguientes posibles parámetros:

Tabla 16. Configuraciones para el modelo Skip-Gram (word2vec)

Parámetro	Valores
Tamaño de la ventana	5, 10, 15
Dimensiones de las representaciones	50, 100, 150, 200, 250, 300

A continuación, se muestran los resultados del coste final de los modelos acorde a las configuraciones:

Tabla 17. Coste final modelo Skip-Gram 5 iteraciones

DIM	**50**			**100**		
WIN	5	10	15	5	10	15
VAL	1.315.400,25	1.973.797,875	2.472.150,5	1.319.615,875	1.977.187,625	2.472.692,25
DIM	**150**			**200**		
WIN	5	10	15	5	10	15
VAL	1.383.252,25	1.986.415,875	2.459.331,5	1.264.243,75	1.976.840,25	2.467.527,25
DIM	**250**			**300**		
WIN	5	10	15	5	10	15
VAL	1.260.244,375	1.977.530,75	2.463.029,5	1.266.600,625	1.976.045,25	2.480.017,25

Seguidamente, sometemos las representaciones a la tarea de categorización de entidades. Para ello, inicialmente obtenemos los tipos de las entidades desde el grafo de conocimiento generado para la asignatura. A continuación, considerando k=7 (recordemos que tenemos 7 tipos principales de entidades en el grafo) y las configuraciones, realizamos la tarea de agrupación con el empleo de la librería Scikit-Learn. Una vez generados los modelos, obtenemos el mismo valor de pureza en todas y cada una de las configuraciones: 0.2247765006385696.

A continuación mostramos las matrices de confusión para varios de los casos contemplados, considerando la asignación de *clusters* [0-> Fármaco/droga, 1-> Biomolécula, 2-> Síntoma, 3-> Trastorno, 4-> Proceso, 5-> Fenómeno, 6-> Estructura]:

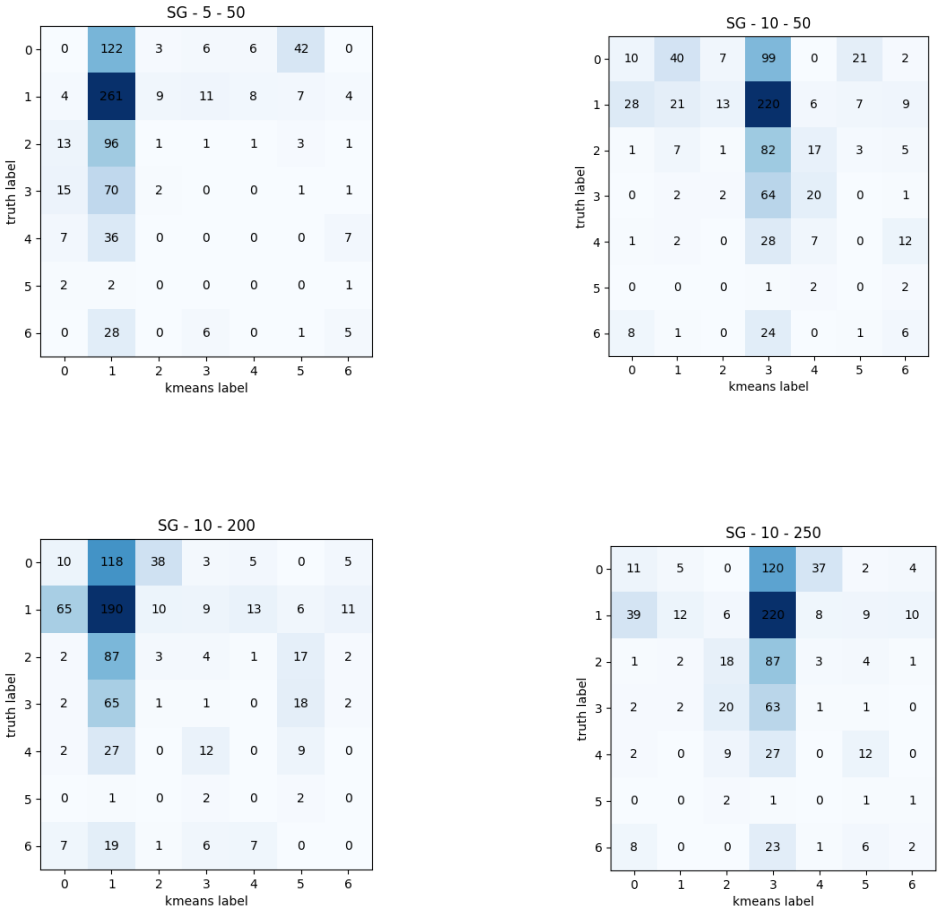


Figura 30. Matrices de confusión para el agrupamiento de entidades (palabras) en relación a su tipo

4.1.2. Modelo transE

Para poder generar *embeddings* de las entidades y propiedades o aristas del grafo de conocimiento, en primer lugar se realiza un volcado de la base de conocimiento en formato textual TSV con tres columnas dedicadas a cada tripleta (entidad sujeto, propiedad, entidad objeto). A pesar de que emplearemos el grafo de conocimiento al completo para el aprendizaje de representaciones, únicamente tendremos en cuenta los 7 tipos de entidades que resultan de interés para la generación de distractores.

En el caso de este modelo, exploramos el empleo de la medida de distancia Manhattan frente a la distancia euclídea como parte del algoritmo explicado en la sección 2.2.2. También exploramos la variación de la dimensión de las representaciones y mostramos el coste total de optimización resultante, considerando dimensiones de 50, 100, 150, 200, 250 y 300 características. Los siguientes resultados muestran el coste total de optimización sobre todas las tripletas del grafo del conocimiento, considerando un total de 5 iteraciones:

DIM	**50**		**100**	
DIS	Manhattan	Euclidean	Manhattan	Euclidean
VAL	0.00232	0.00326	0.00132	0.00288
DIM	**150**		**200**	
DIS	Manhattan	Euclidean	Manhattan	Euclidean
VAL	0.000834	0.00256	0.00061	0.00227
DIM	**250**		**300**	

DIS	Manhattan	Euclidean	Manhattan	Euclidean
VAL	0.000488	0.00203	0.00041	0.00237

Figura 31. Costes de optimización modelo transE con 5 iteraciones sobre el conjunto total de tripletas

Una vez obtenidas las distintas representaciones, se lleva a cabo la tarea de agrupación para la categorización de las entidades (nodos), observando siempre el mismo resultado de pureza: 0.1780199

A continuación se muestran algunas de las matrices de confusión de los modelos frente a la tarea de agrupamiento ([0-> Fármaco/droga, 1-> Biomolécula, 2-> Síntoma, 3-> Trastorno, 4-> Proceso, 5-> Fenómeno, 6-> Estructura]):

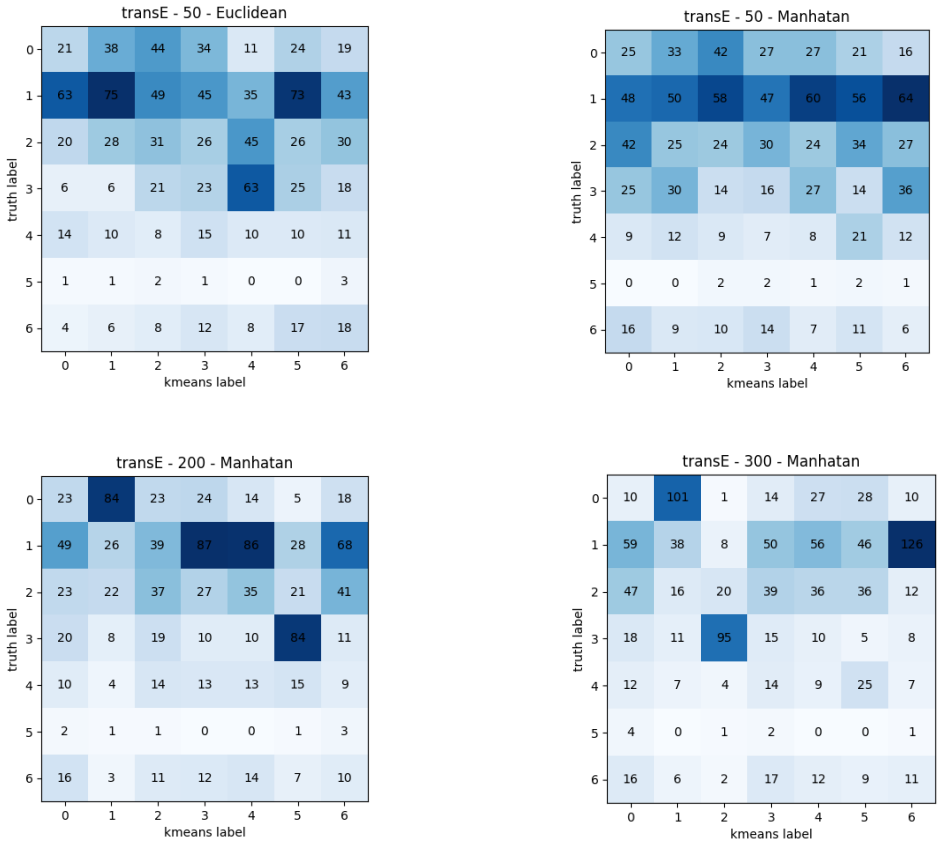


Figura 32. Matrices de confusión para la tarea de agrupamiento de nodos (entidades) empleando transE

4.1.3. Modelo conjunto

Para poder llevar a cabo una optimización de las representaciones vectoriales correspondientes tanto a palabras como a nodos, generamos varios modelos conjuntos o mixtos acorde a la propuesta descrita en el apartado 2.2.3. Por tanto, los datos para el entrenamiento de este modelo están formados por los conjuntos de datos empleados en los dos modelos anteriores, empleando todo el texto y todas las tripletas del grafo de conocimiento.

En este caso, exploraremos el efecto del tamaño de la ventana así como de las dimensiones de las representaciones vectoriales del siguiente modo:

Tabla 18. Variación de parámetros para el modelo conjunto

Parámetro	Valores

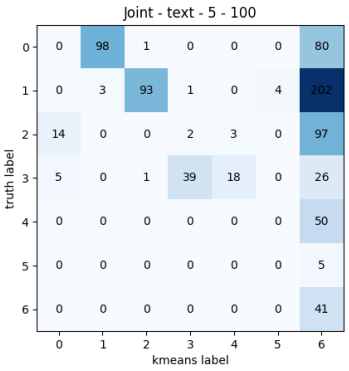
Tamaño de la ventana	5, 10, 15
Dimensiones de las representaciones	100, 150, 200, 250, 300

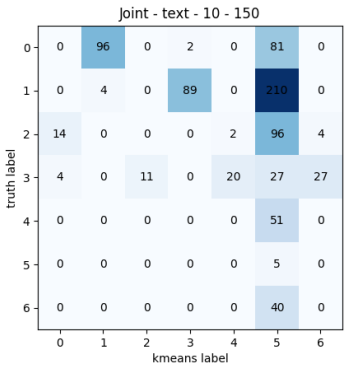
Empleando una configuración por defecto de la implementación de [6] (15 épocas) sobre la que se han llevado a cabo ciertas modificaciones de las estructuras principales de datos empleadas el dicho desarrollo, obteniéndose los siguientes costes y valores de pureza en la tarea de categorización:

Tabla 19. Resultados de pureza y coste global en modelo conjunto

DIM	**100**			**150**		
WIN	5	10	15	5	10	15
VAL	5.2794	5.3163	5.5168	5.2786	5.2344	5.5054
PUR	0.2286	0.2247	0.2273	0.2260	0.2286	0.2286
DIM	**200**			**250**		
WIN	5	10	15	5	10	15
VAL	5.4062	5.596	5.0163	5.2306	5.708	5.5279
PUR	0.2234	0.2286	0.2286	0.2158	0.2298	0.2298
DIM	**300**					
WIN	5	10	15			
VAL	5.2535	5.6369	5.6753			
PUR	0.2273	0.2247	0.2260			

A continuación mostramos varios ejemplos de las matrices de confusión obtenidas durante la tarea de *clustering*, teniendo en cuenta la agrupación de las entidades que se corresponden con nodos del grafo ([0-> Fármaco/droga, 1-> Biomolécula, 2-> Síntoma, 3-> Trastorno, 4-> Proceso, 5-> Fenómeno, 6-> Estructura]):





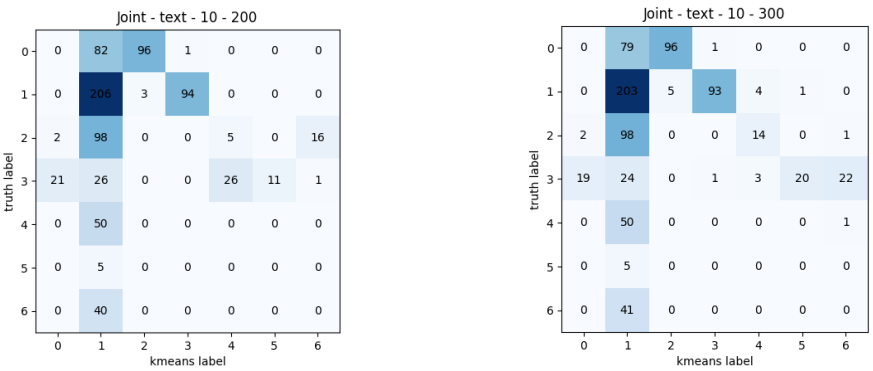


Figura 33. Matrices de confusión para la categorización de entidades en el modelo mixto

4.2. Análisis de los modelos

En primer lugar, debemos considerar que el coste de optimización no es una medida adecuada para garantizar la bondad de las representaciones, puesto que, a pesar de que los objetivos de los modelos se basan en tareas de predicción "simuladas" (recordemos que en realidad los modelos realizan una aproximación de las predicciones empleando técnicas más óptimas pero no exactamente equivalentes), nuestro interés no radica en los pesos de la capa predictiva, sino en los de la capa intermedia.

Por otro lado, el empleo de técnicas de *clustering* puede resultar orientativo, ya que una de las propiedades que se desean de las características es que las entidades similares queden cerca unas de otras en el espacio vectorial. Si atendemos a estos resultados de pureza obtenidos, podríamos caer en el error de suponer que el modelo Skip-Gram es el que mejor caracterización de las entidades logra. Sin embargo, si atendemos a las matrices de confusión, el resultado se debe al poco balance o la poca discriminación que se realiza entre grupos.

Para verificar esta hipótesis, generamos una serie de visualizaciones en 3D de los *embeddings* con el empleo de la librería Tensorflow, la cual realiza un cómputo PCA de las representaciones para su proyección en un espacio de dimensiones reducido. Como podemos ver a continuación, el modelo de Skip-Gram resulta en una gran esfera en la que las entidades si tipo asignadas (en rojo) son las predominantes, corroborando los resultados de sus matrices de confusión:

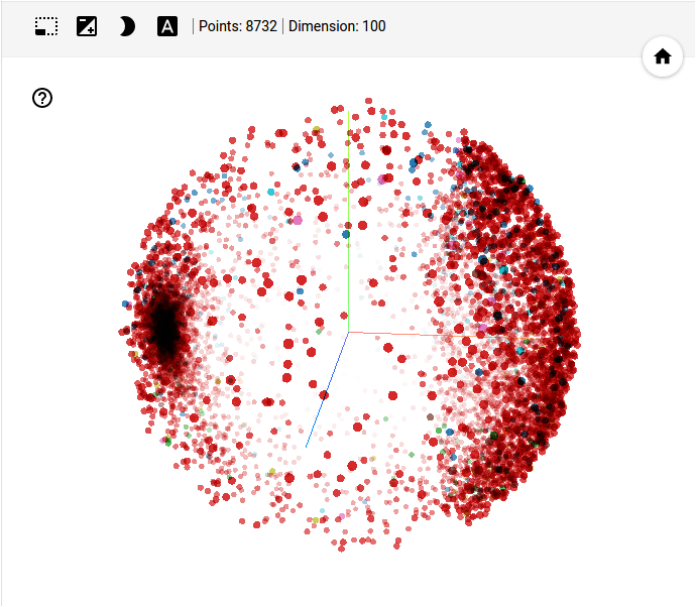


Figura 34. Proyección de embeddings Skip-Gram

Por otro lado, podemos observar como el modelo transE sí que realiza una categorización de entidades más razonable, a pesar de que la puntuación de pureza sigue siendo poco elevada:

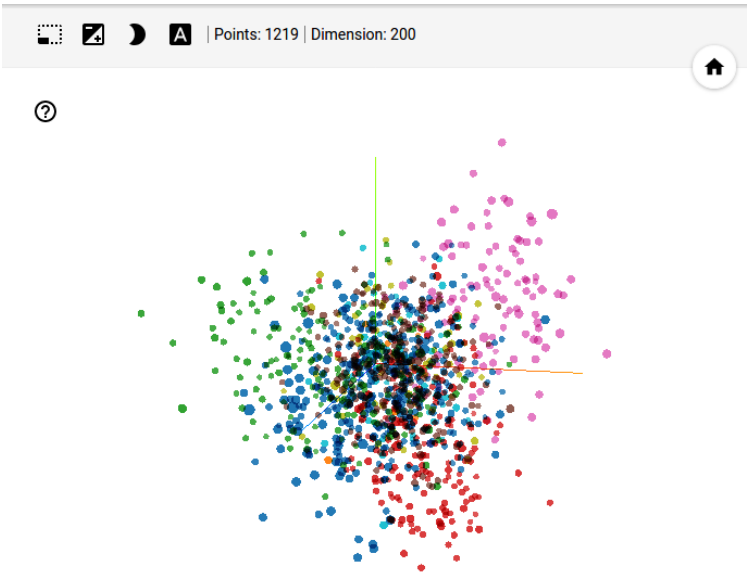


Figura 35. Proyección de embeddings transE

Algo parecido sucede a la hora de visualizar los *embeddings* del modelo conjunto. La siguiente imagen muestra el agrupamiento que se realiza teniendo en cuenta aquellas palabras que tienen una entrada en el grafo de conocimiento:

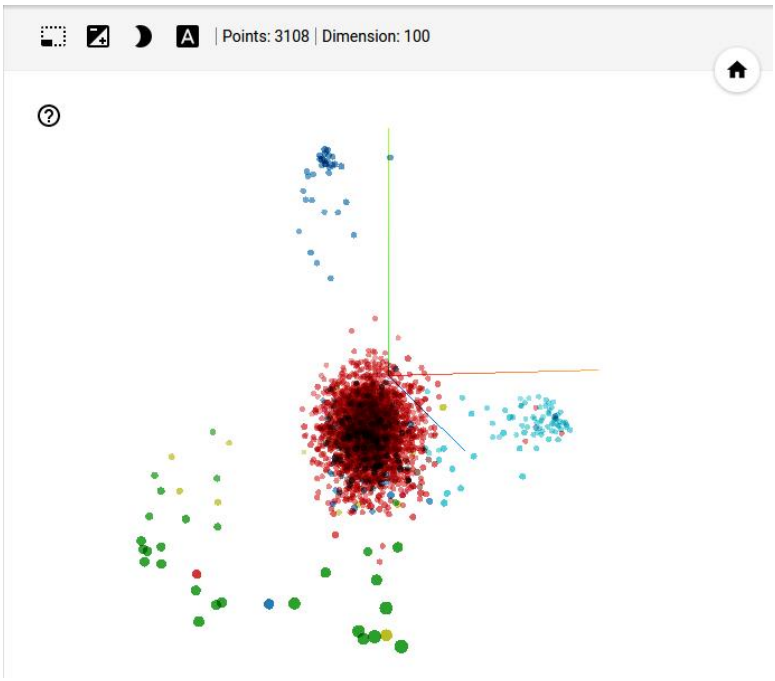


Figura 36. Proyección de embeddings conjuntos (solo palabras tipadas)

Finalmente, mostramos el mejor modelo de *embeddings* de cara a la tarea de agrupamiento según la medida de pureza, que resultan ser las representaciones de los nodos según el modelo conjunto:

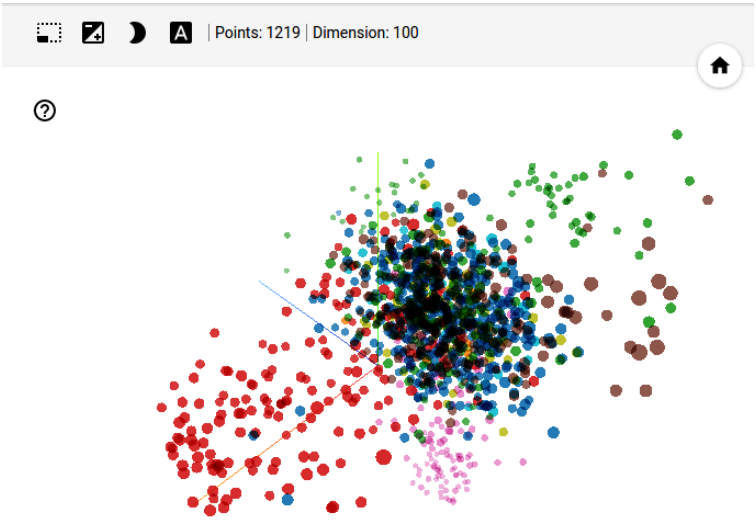


Figura 37. Proyección de embedding modelo conjunto (sólo nodos)

Finalmente, como parte de la evaluación intrínseca de los modelos, se lleva a cabo un estudio de similitudes para saber cuál de los modelos genera, en un principio, las entidades más similares a un conjunto preestablecido de entidades [52]. La experimentación se lleva a cabo del siguiente modo:

1. Se propone un conjunto de entidades de referencia.

2. Por cada entidad de referencia, cada modelo genera las entidades más similares en los rankings 1, 3 y 6. Generando tres ejercicios de evaluación por cada entidad.
3. Se le pide al experto docente que seleccione, por cada ejercicio de evaluación, la entidad que más similitudes guarda con la entidad de referencia.
4. Contamos el número de veces que cada modelo ha sido seleccionado como el mejor. Es decir, el número de veces que el experto ha declarado como más similar las entidades generadas por cada modelo.
5. Comparamos la puntuación final de la evaluación.

Acorde a las pautas establecidas, empleamos los modelos Skip-Gram, transE y conjunto (empleamos las similitudes basadas en texto, que puede que nos devuelvan una entidad más similar correspondiente a una palabra o un nodo), obteniéndose las siguientes puntuaciones:

Tabla 20. Puntuaciones de las medidas de similitud en evaluación directa de similitudes

Skip-Gram	**transE**	**Conjunto (sim. palabra)**
25/69	22/69	22/69

La evaluación manual llevó un total de 2 horas y media. El experto resaltó la dificultad del ejercicio, dado que en muchos casos era muy complicado decidirse por uno de los tres casos propuestos, por lo que se tuvo que descartar en lugar de escoger en la mayor parte de los casos. Estos resultados nos indican que no existen diferencias notables entre la capacidad de agrupar entidades de semántica similar entre modelos, por lo que emplearlos de manera directa no resultará beneficioso dependiendo de la tarea extrínseca que se desee afrontar.

Por estos motivos, de cara a la propuesta automática de distractores, tendremos que realizar un filtrado de las entidades similares a la clave según el tipo semántico para poder garantizar que los distractores son coherentes.

5. Propuesta automática de distractores

En este apartado se muestra el experimento de la propuesta de distractores sobre un conjunto de enunciados creado partiendo de un banco de preguntas de test y preguntas de años anteriores de la asignatura de psicofarmacología.

Mostraremos la política de generación de enunciados, así como la de identificación de las claves, para seguidamente evaluar los distractores generados acorde a las medidas de similitud o distancia teniendo en cuenta una única configuración por cada uno de los modelos de *embeddings* generados.

Se analizarán los resultados para poder contestar a las preguntas de investigación planteadas.

5.1. Método de generación de preguntas de hueco en blanco

Inicialmente partimos de un conjunto de preguntas de test oficiales empleadas para la evaluación de las pruebas presenciales de la asignatura de psicofarmacología. Dichas preguntas están compuestas por un enunciado y dos distractores. Algunas de las preguntas se basan en enunciados afirmativos, mientras que una parte de las preguntas quedan expresadas mediante enunciados imperativos o interrogativos.

En relación al receptor NMDA, se hipotetiza que los síntomas positivos de la esquizofrenia son debidos a una: a) hiperfunción de este receptor; b) hipofunción de este receptor.

Figura 38. Ejemplo de enunciado afirmativo

Indique en qué triada se ha incluido un anticonvulsivante que, además, actúa como probado estabilizador del ánimo: a) Mirtazapina-ácido valproico-citalopram; b) Monoterapia de segunda línea.

Figura 39. Ejemplo de enunciado imperativo

Realizamos un descarte de los enunciados imperativos, y llevamos a cabo una conversión automática de las preguntas afirmativas a enunciados, almacenando información del tema de la asignatura al que pertenecen. Para ello, concatenamos la clave de las preguntas al enunciado de la pregunta.

En relación al **receptor NMDA**, se hipotetiza que los **síntomas positivos** de la **esquizofrenia** son debidos a una **hipofunción** de este **receptor**.

Figura 40. Ejemplo de conversión de pregunta afirmativa a enunciado

A este punto, dada la calidad del conjunto de enunciados, realizamos la suposición de que podemos generar nuevas preguntas de hueco en blanco para cada una de las entidades que se identifiquen en el enunciado. Es decir, para cada uno de los términos para los que se identifica una entrada o entidad en el grafo de conocimiento. Sin embargo, para aumentar la calidad de las claves seleccionadas, llevamos a cabo una reducción del conjunto de claves candidatas teniendo en cuenta que:

- Las claves candidatas sólo pueden aparecer una vez en el enunciado.
- Las claves candidatas correspondientes con términos multi-palabra sólo pueden aparecer si en el enunciado no aparece ninguna de las palabras que lo forman.
- Las entidades asociadas a las claves candidatas sólo aparecen una vez en el enunciado.

Mientras que los dos primeros puntos se abordan directamente sobre el texto de los enunciados, para abordar el tercer punto es necesario consultar los enlaces del tipo owl:sameAs recopilados en el grafo de conocimiento. Teniendo en cuenta todo lo anterior, recopilamos un conjunto inicial de 156 enunciados, lo que, considerando las posibles claves, da lugar a un total de 321 preguntas distintas de hueco en blanco.

Posteriormente, el método que empleamos para la propuesta automática de distractores se basa en emplear la distancia entre representaciones como medida de similitud. De esta forma, por cada clave se identificarán las tres entidades más cercanas a dicha clave, siempre y cuando coincidan con el tipo de la clave en el grafo de conocimiento. Así podemos garantizar cierta coherencia de los distractores debido a que el empleo aislado de las medidas de similitud no resulta adecuado tal y como se pudo comprobar durante la generación de los modelos.

5.2. Evaluación

Se diseñan dos ejercicios de evaluación para contestar a las preguntas de investigación formuladas.

En un primer ejercicio, emplearemos el modelo conjunto para la propuesta de distractores y compararemos los conjuntos generados con una propuesta aleatoria. Como el modelo conjunto contiene tanto palabras como nodos representados en el mismo espacio vectorial, propondremos los distractores generados en base a la similitud frente a la entidad de la clave textual, así como frente a la entidad de la clave en el grafo. Además, ya que realizamos un filtrado de distractores según el tipo o clase de la clave en el grafo, emplearemos la misma estrategia para el caso de la propuesta de distractores aleatoria.

Para este primer propósito, generamos un conjunto de 25 enunciados extraídos con el empleo de un sumarizador de texto basado en funciones submodulares [53]. Dado que los enunciados pertenecen a distintos temarios de la asignatura, impondremos un factor de novedad alto a la hora de emplear el sumarizador para favorecer la aparición de preguntas variadas. Una vez extraído el conjunto de enunciados, llevamos a cabo la selección de claves acorde al método descrito.

La evaluación de este primer ejercicio se lleva a cabo con el empleo de formularios. Para cada pregunta, generamos un formulario con información sobre la pregunta y los tres conjuntos de distractores generados (como hemos indicado, uno según el modelo conjunto pero basado en palabras, otro según el modelo conjunto pero basado en nodos, y un tercer conjunto aleatorio). Así, la evaluación de cada una de las preguntas conlleva:

1. Escoger entre los tres niveles de pregunta:
    - La pregunta queda invalidada: si el docente considera que aquello que se deja en blanco hace que la pregunta carezca de sentido.
    - La pregunta es válida pero no relevante: si el docente considera la pregunta apropiada para el aprendizaje de psicofarmacología en general, pero no particularmente en su asignatura.
    - La pregunta es relevante: si el docente consideraría útil la pregunta en el ámbito general y el de su asignatura.
2. Sólo en el caso de que la pregunta sea válida, por cada uno de los tres conjuntos de distractores:
    - El docente deberá escoger entre una de las tres opciones, siendo las dos primeras motivo de invalidez del conjunto de distractores.
    - El docente escogerá la tercera opción si el conjunto de distractores es válido.
3. Sólo en el caso de que la pregunta y alguno de los conjuntos de distractores sea válido, el docente deberá seleccionar el conjunto de distractores que le parezca más afín como la mejor opción.

A continuación, se muestra un ejemplo de pregunta evaluada:

Tabla 21. Formulario de evaluación de FIBQs para el primer ejercicio

P – 152

Un alcohólico crónico necesitaría tomar calmantes -si fuera preciso- en dosis mayores a las habituales requeridas por la población normal debido a la _____ y la dependencia cruzadas que tienen las benzodiacepinas y los barbitúricos y el alcohol **CLAVE: TOLERANCIA**
☐ PREGUNTA INVÁLIDA ☐ La pregunta es válida pero no relevante ☒ La pregunta es relevante

DISTRACTORES - 1	DISTRACTORES - 2	DISTRACTORES - 3
acatisia	Agresión	muerte_neuronal
mueca	Malestar	insomnio
retencion_urinaria	Pérdida de interés	panico
☐ Algún distractor es respuesta correcta ☒ Los distractores son muy obvios ☐ **Los distractores son válidos**	☐ **Algún distractor es respuesta correcta** ☐ **Los distractores son muy obvios** ☒ Los distractores son válidos	☐ **Algún distractor es respuesta correcta** ☒ **Los distractores son muy obvios** ☐ Los distractores son válidos
MEJOR OPCIÓN ☐	MEJOR OPCIÓN ☒	MEJOR OPCIÓN ☐

El experto docente no tiene conocimiento previo sobre los modelos empleados, desconociendo el proceso que ha generado cada uno de los tres conjuntos de distractores.

Un aspecto importante de esta evaluación es que la concordancia gramatical de los distractores no se evalúa. Pensamos que esta es una primera aproximación para determinar la factibilidad del método propuesto, y que podemos modificar los enunciados o los distractores con técnicas de Procesamiento del Lenguaje Natural para apaciguar los problemas de concordancia léxico-gramatical.

En un segundo ejercicio, y con el objetivo de comparar la calidad de los distractores basados en similitud del modelo conjunto frente a los de los modelos Skip-Gram y transE, recopilamos los mejores conjuntos de distractores resultantes del primer ejercicio (si existen), y son sometidos a una reevaluación frente a los dos nuevos conjuntos de distractores. De esta forma, el tipo de formulario empleado es similar al ya mostrado.

Tabla 22. Ejemplo de formulario del segundo ejercicio de evaluación - comparación de modelos

P – 152
Un alcohólico crónico necesitaría tomar calmantes -si fuera preciso- en dosis mayores a las habituales requeridas por la población normal debido a la _____ y la dependencia cruzadas que tienen las benzodiacepinas y los barbitúricos y el alcohol **CLAVE: TOLERANCIA**
☐ **PREGUNTA INVÁLIDA** ☐ **La pregunta es válida pero no relevante** ☒ **La pregunta es relevante**

MEJOR-OPCION	DISTRACTORES - 1	DISTRACTORES - 2
Agresión	Ajuste	Hiperpirexia
Malestar	Visión borrosa	Ideación suicida
Pérdida de interés	fumador	Colapso cardiovascular
	☐ **Algún distractor es respuesta correcta** ☒ **Los distractores son muy obvios** ☐ Los distractores son válidos	☐ **Algún distractor es respuesta correcta** ☒ **Los distractores son muy obvios** ☐ Los distractores son válidos
	MEJOR OPCIÓN ☐	MEJOR OPCIÓN ☐

5.3. Análisis de los resultados

El experto docente completó el primer ejercicio en un periodo de tiempo de a una hora, aproximadamente, mientras que le tomó media hora llevar a cabo el segundo ejercicio.

Comenzaremos comentando los resultados del primer ejercicio. Estos muestran una aceptación general de las preguntas. Es decir, todas las 25 preguntas sometidas a evaluación se consideraron como relevantes en el contexto de la asignatura, a pesar de la sencillez del enfoque de identificación de claves. Por otra parte, 4 de las 25 preguntas no lograron disponer de un conjunto de distractores válido. De las 21 preguntas con conjunto de distractores válidos, 4 de ellas corresponden con el conjunto generado de manera aleatoria. En esos 4 casos podemos observar que la similitud basada en el modelo conjunto genera distractores demasiado obvios. Por el contrario, de las 21 preguntas con distractores válidos, la generación aleatoria produce distractores válidos en dos ocasiones.

Por otra parte, 18 de las 25 preguntas contienen como mejor opción uno de los dos conjuntos de distractores propuestos según el modelo conjunto, de los cuales 11 se han identificado como mejor opción teniendo en cuenta la similitud de palabras y 7 según la similitud de nodos. Cabe resaltar que, de estos 7 casos, el modelo conjunto basado en palabras es capaz de generar distractores válidos para 6 de ellos, quedando invalidado una vez por obviedad. Sin embargo, esto no sucede a la inversa, es decir, de los 11 casos en los que el modelo conjunto basado en palabras se proclama como mejor opción, el modelo conjunto basado en nodos logra distractores válidos en 4 ocasiones.

De este modo, podemos decir que el modelo conjunto basado en palabras genera distractores apropiados en 17 de los 25 casos (68% de los casos), lo cual no es suficiente para poder afirmar la efectividad de medidas de similitud basadas en modelos conjuntos para la propuesta de distractores en preguntas de hueco en blanco.

En cuanto a los resultados del segundo ejercicio, podemos ver cómo, de los 21 casos con distractores válidos del primer ejercicio, con el modelo basado en Skip-Gram producimos mejores distractores en 6 ocasiones, mientras que el modelo basado en transE produce mejores distractores en 4 casos. Por otro lado, encontramos 7 casos en los que el modelo conjunto produce mejores distractores que con los dos modelos restantes. De este modo, no podemos afirmar que el empleo de las medidas de similitud basadas en modelos conjuntos implique una mejor propuesta de distractores frente a los métodos transE y Skip-Gram, a pesar de que el

método conjunto produce 11 conjuntos de distractores válidos frente a los 7 obtenidos con modelo SKip-Gram y los 8 acorde al modelo transE.

Dados los resultados negativos, cabe preguntarse en qué casos fallan las similitudes basadas en modelo conjunto y en qué casos no, así como en qué casos fracasan o tienen éxito las medidas basadas en los modelos de *embeddings* de palabras y *embeddings* de grafos de conocimiento:

- En primer lugar, si atendemos a los tipos de las claves de aquellas preguntas para las que no ha sido posible obtener distractores válidos durante el primer ejercicio, observamos dos problemas. Uno de ellos se debe a que las entidades, si bien su tipo ha sido correctamente identificado en el grafo de conocimiento, no obtiene el carácter de entidad o no con la misma fuerza que otras entidades del mismo tipo. Por ejemplo, "bloqueo" está considerado como un síntoma, sin embargo, dependiendo del contexto, esta palabra puede referirse o no a un síntoma real. Entonces, la propuesta de síntomas más propiamente dichos para sustituir la palabra "bloqueo" puede resultar en una propuesta de distractores demasiado obvios. Por otra parte, otro de los problemas puede residir en la ambigüedad de tipos de entidades. Por ejemplo, la entidad "noradrenalina" podría considerarse tanto como una biomolécula como un fármaco/droga, por lo que es necesario inferir el tipo en base al contexto antes de la propuesta de distractores.
- En segundo lugar, observamos que las medidas del modelo Skip-Gram obtienen éxito sobre todo cuando la clave es un término de una sola palabra, lo que se nota aún más cuando dicho término es muy frecuente en el corpus (como por ejemplo "miedo" o "monoaminas") o cuando la entidad asociada en el grafo de conocimiento no contiene enlaces suficientes con otras entidades. Esto tiene sentido en el contexto de los *embeddings* de palabras, que como ya hemos visto, favorecen el muestreo de palabras más frecuentes, a pesar de que el modelo cuenta con un mecanismo para suavizar este efecto.
- En tercer lugar, en analogía con el anterior punto, observamos que el modelo conjunto obtiene mayor éxito para proponer distractores cuando la entidad asociada a la clave en el grafo de conocimiento contiene una cantidad considerable de enlaces.

De esta forma, para poder mejorar el sistema es necesario eliminar ambigüedades y mejorar la completitud del grafo, así como explorar métodos más avanzados de *sampling* de muestras en los algoritmos de aprendizaje de características.

6. Conclusiones y trabajos futuros

En este trabajo hemos abordado la generación de distractores para FIBQs con el empleo de medidas de similitud basadas en modelos de *embeddings*. Los resultados muestran que, con la metodología empleada, somos capaces de proponer distractores válidos en un 68% de los casos, lo que no es suficiente como para validar el enfoque, pero sí sería suficiente para el empleo del método de cara al soporte de la creación de preguntas de hueco en blanco basadas en entidades por parte de los docentes.

A pesar de que los mejores distractores generalmente se obtienen con el empleo de *embeddings* según el modelo conjunto, existen varios casos en los que el empleo de las medidas acorde a los modelos Skip-Gram y transE resulta óptimo. De esta forma, no podemos afirmar que el modelo conjunto sea la mejor de las opciones, sino que dependerá de los casos particulares tal y como se ha detallado en la sección 5.

En cuanto a la relación de la calidad intrínseca de los *embeddings* frente a la calidad en la tarea extrínseca de generación de distractores, los experimentos sugieren que la tarea de agrupamiento para la caracterización de entidades según su tipo puede darnos cierta indicación sobre la bondad de las representaciones de cara a la tarea extrínseca planteada. Sin embargo, desaconsejamos el empleo de evaluación intrínseca de similitudes, ya que los modelos no son capaces de generar entidades relacionadas sin un filtrado previo por tipos.

Por otra parte, algunos de los problemas encontrados durante la evaluación de la propuesta de distractores, se correspondieron con ambigüedades a nivel del grafo de conocimiento o bien al nivel del enunciado, por lo que deberíamos disponer de algún método que permitiera resolver estos problemas, como por ejemplo, empleando técnicas de desambiguación de semántica de palabras o empleando modelos de generación de representaciones contextualizados.

Otro método que convendría explorar es la propuesta de medidas de similitud basadas en el grafo de conocimiento, aunque para poder obtener buenas medidas, pensamos que es conveniente ofrecer un nivel más alto de expresividad que el *schema* o vocabulario acordado. Para ello, se podrían recopilar una serie de taxonomías o jerarquías de clases que ofrecieran una mayor granularidad de los tipos de entidades contemplados. Gracias a que el grafo se encuentra en formato semántico enlazable, podríamos llevar a cabo una tarea de enriquecimiento desde Wikidata o desde BioPortal.

De todas formas, el grafo de conocimiento puede constituir por sí mismo un artefacto sobre el que desarrollar otro tipo de aplicaciones educativas, como por ejemplo, interfaces de navegación sobre los contenidos de la asignatura, así como interfaces para facilitar la recuperación de los datos contenidos. De esta forma, un alumno podría apoyarse en dicho sistema para formar una conceptualización acertada de los elementos de la asignatura. Aunque esto podría conllevar la necesidad de llevar a cabo una poda o curado del grafo para una adecuación más focalizada en la asignatura. A tal propósito, cabría pensar si un entorno colaborativo de etiquetado, que agrupara a alumnos y profesores, resultaría de interés.

7. Referencias

[1] Laredo, P., (2007). Revisiting the Third Mission of Universities: Toward a Renewed Categorization of University Activities. Higher Education Policy 20, 441–456. https://doi.org/10.1057/palgrave.hep.8300169

[2] European Commission, Directorate-General for Education and Culture, LSE, Panteia, (2014). Study on innovation in higher education final report. Publications Office, Luxembourg.

[3] E v, Vinu. (2015). A novel approach to generate MCQs from domain ontology: Considering DL semantics and open-world assumption. Journal of Web Semantics. 10.1016/j.websem.2015.05.005.

[4] Mikolov, Tomas & Sutskever, Ilya & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems. 26.

[5] Bordes, Antoine & Usunier, Nicolas & Garcia-Duran, Alberto & Weston, Jason & Yakhnenko, Oksana. (2013). Translating Embeddings for Modeling Multi-relational Data. 2013.

[6] Wang, Zhen & Zhang, Jianwen & Feng, Jianlin & Chen, Zheng. (2014). Knowledge Graph and Text Jointly Embedding. 1591-1601. 10.3115/v1/D14-1167.

[7] Cardoso, Jorge. (2007). The Semantic Web Vision: Where Are We?. Intelligent Systems, IEEE. 22. 84 - 88. 10.1109/MIS.2007.4338499.
[8] Domingue, John & Fensel, Dieter & Hendler, James. (2011). Handbook of Semantic Web Technologies. 10.1007/978-3-540-92913-0.
[9] Best Practices for Publishing Linked Data
[10] Debattista, Jeremy & Lange, Christoph & Auer, Sören & Cortis, Dominic. (2018). Evaluating the Quality of the LOD Cloud: An Empirical Investigation.
[11] Pan, Jeff & Matentzoglu, Nicolas & Jay, Caroline & Vigo, Markel & Zhao, Yuting. (2017). Understanding Author Intentions: Test Driven Knowledge Graph Construction. 10.1007/978-3-319-49493-7_1.
[12] Paulheim, H., (2018). Machine Learning with and for Semantic Web Knowledge Graphs, in: d'Amato, C., Theobald, M. (Eds.), Reasoning Web. Learning, Uncertainty, Streaming, and Scalability: 14th International Summer School 2018, Esch-Sur-Alzette, Luxembourg, September 22–26, 2018, Tutorial Lectures. Springer International Publishing, Cham, pp. 110–141. https://doi.org/10.1007/978-3-030-00338-8_5
[13] Martinez-Rodriguez, Jose & Hogan, Aidan & Lopez-Arevalo, Ivan. (2018). Information extraction meets the Semantic Web: A survey. Semantic Web. 11. 1-81. 10.3233/SW-180333.
[14] Jetschni, Jonas & Meister, Vera. (2017). Schema engineering for enterprise knowledge graphs: A reflecting survey and case study. 271-277. 10.1109/INTELCIS.2017.8260074.
[15] Smith, Barry & Ashburner, Michael & Rosse, Cornelius & Bard, Jonathan & Bug, William & Ceusters, Werner & Goldberg, Louis & Eilbeck, Karen & Ireland, Amelia & Mungall, Christopher & Leontis, Neocles & Rocca-Serra, Philippe & Ruttenberg, Alan & Sansone, Susanna-Assunta & Scheuermann, Richard & Shah, Nigam & Whetzel, Patricia & Lewis, Suzanna. (2007). The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature biotechnology. 25. 1251-5. 10.1038/nbt1346.
[16] Haussmann, Steven & Seneviratne, Oshani & Chen, Yu & Ne'eman, Yarden & Codella, James & Chen, Ching-Hua & Mcguinness, Deborah & Zaki, Mohammed. (2019). FoodKG: A Semantics-Driven Knowledge Graph for Food Recommendation. 10.1007/978-3-030-30796-7_10.
[17] Rashid, Sabbir & Chastain, Katherine & Stingone, Jeanette & Mcguinness, Deborah & McCusker, James. (2017). The Semantic Data Dictionary Approach to Data Annotation & Integration.
[18] Belleau, François & Nolin, Marc-Alexandre & Tourigny, Nicole & Rigault, Philippe & Morissette, Jean. (2008). Bio2RDF: Towards A Mashup To Build Bioinformatics Knowledge System. Journal of biomedical informatics. 41. 706-16. 10.1016/j.jbi.2008.03.004.
[19] Marchand, Erwan & Gagnon, Michel & Zouaq, Amal. (2020). Extraction of a Knowledge Graph from French Cultural Heritage Documents. 10.1007/978-3-030-55814-7_2.
[20] Corcoglioniti, Francesco & Rospocher, Marco & Aprosio, Alessio. (2016). Frame-Based Ontology Population with PIKES. IEEE Transactions on Knowledge and Data Engineering. 28. 3261-3275. 10.1109/TKDE.2016.2602206.
[21] Niklaus, Christina & Cetto, Matthias & Freitas, Andre & Handschuh, Siegfried. (2018). A Survey on Open Information Extraction.
[22] Gao, Junyang & Li, Xian & Xu, Yifan & Sisman, Bunyamin & Dong, Xin & Yang, Jun. (2019). Efficient Knowledge Graph Accuracy Evaluation.

[23]Wang, Quan & Mao, Zhendong & Wang, Bin & Guo, Li. (2017). Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering. PP. 1-1. 10.1109/TKDE.2017.2754499.
[24]Paulheim, Heiko. (2016). Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web. 8. 489-508. 10.3233/SW-160218.
[25]Al-yahya, Maha & George, Remya & Alfaries, Auhood. (2015). Ontologies in E-Learning: Review of the literature. International Journal of Software Engineering and its Applications. 9. 67-84. 10.14257/ijseia.2015.9.2.07.
[26]Devedzic, Vladan. (2006). Semantic Web and Education. 10.1007/978-0-387-35417-0.
[27]Signer, Beat & Ilkou, Eleni. (2020). A Technology-enhanced Smart Learning Environment Based on the Combination of Knowledge Graphs and Learning Paths. 10.5220/0009575104610468.
[28]Grévisse, Christian & Manrique, Ruben & Marino, Olga & Rothkugel, Steffen. (2018). Knowledge Graph-Based Teacher Support for Learning Material Authoring.. 10.1007/978-3-319-98998-3_14.
[29]Meissner, Roy & Köbis, Laura. (2020). Annotated Knowledge Graphs for Teaching in Higher Education: Supporting Mentors and Mentees by Digital Systems. 10.1007/978-3-030-50578-3_43.
[30]d'Aquin, Mathieu. (2016). On the Use of Linked Open Data in Education: Current and Future Practices. 10.1007/978-3-319-30493-9_1.
[31]Bengio, Y. & Courville, Aaron & Vincent, Pascal. (2013). Representation Learning: A Review and New Perspectives. IEEE transactions on pattern analysis and machine intelligence. 35. 1798-1828. 10.1109/TPAMI.2013.50.
[32]Bengio, Y. & Ducharme, Réjean & Vincent, Pascal & Jauvin, Christian. (2006). Neural Probabilistic Language Models. 10.1007/3-540-33486-6_6.
[33]Mikolov, T. & Yih, W.-T & Zweig, G.. (2013). Linguistic regularities in continuous space word representations. Proceedings of NAACL-HLT. 746-751.
[34]Yamada, Ikuya & Shindo, Hiroyuki & Takeda, Hideaki & Takefuji, Yoshiyasu. (2016). Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. 250-259. 10.18653/v1/K16-1025.
[35]Nikolaev, Fedor & Kotov, Alexander. (2020). Joint Word and Entity Embeddings for Entity Retrieval from a Knowledge Graph. 10.1007/978-3-030-45439-5_10.
[36]Rao, Dhawaleswar & Saha, Sujan Kumar. (2018). Automatic Multiple Choice Question Generation from Text : A Survey. IEEE Transactions on Learning Technologies. PP. 1-1. 10.1109/TLT.2018.2889100.
[37]Kurdi, Ghader & Leo, Jared & Parsia, Bijan & Al-Emari, Salam. (2019). A Systematic Review of Automatic Question Generation for Educational Purposes. International Journal of Artificial Intelligence in Education. 10.1007/s40593-019-00186-y.
[38]Alsubait, Tahani & Parsia, Bijan & Sattler, Uli. (2015). Ontology-Based Multiple Choice Question Generation. KI - Künstliche Intelligenz. 30. 10.1007/s13218-015-0405-9.
[39]v, Vinu. (2016). Automated Generation of Assessment Tests from Domain Ontologies. Semantic Web.
[40]Alsubait, T., Parsia, B., & Sattler, U. (2014). Generating Multiple Choice Questions From Ontologies: Lessons Learnt. OWLED.
[41]Leo, Jared & Kurdi, Ghader & Matentzoglu, Nicolas & Parsia, Bijan & Sattler, Uli & Forge, Sophie & Donato, Gina & Dowling, Will. (2019). Ontology-Based Generation of Medical, Multi-term MCQs. International Journal of Artificial Intelligence in Education. 10.1007/s40593-018-00172-w.

[42]Haladyna, T.M. & Downing, S.M. & Rodriguez, M.C.. (2002). A Review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment. Applied Measurement in Education. 15. 309-334.
[43]Kumar, Girish & Banchs, Rafael & D'Haro, Luis. (2015). RevUP: Automatic Gap-Fill Question Generation from Educational Texts. 154-161. 10.3115/v1/W15-0618.
[44]Majumder, Mukta & Saha, Sujan Kumar. (2014). Automatic selection of informative sentences: The sentences that can generate multiple choice questions. Knowledge Management and E-Learning. 6. 377-391.
[45]Afzal, Naveed & Mitkov, Ruslan. (2014). Automatic generation of multiple choice questions using dependency-based semantic relations. Soft Computing. 18. 1269-1281. 10.1007/s00500-013-1141-4.
[46]Rao, Dhawaleswar & Saha, Sujan Kumar. (2018). RemedialTutor: A blended learning platform for weak students and study its efficiency in social science learning of middle school students in India. Education and Information Technologies. 24. 1-17. 10.1007/s10639-018-9813-4.
[47]Becker, Lee & Basu, Sumit & Vanderwende, Lucy. (2012). Mind the gap: Learning to choose gaps for question generation. 742-751.
[48]Aldabe, Itziar & Maritxalar, Montse. (2010). Automatic Distractor Generation for Domain Specific Texts. 27-38. 10.1007/978-3-642-14770-8_5.
[49]Pannu, Sumeet & Krishna, Aishwarya & Shiwani, Kumari & Patra, Rakesh & Saha, Sujan Kumar. (2018). Automatic Generation of Fill-in-the-Blank Questions From History Books for School-Level Evaluation. 10.1007/978-981-10-7871-2_44.
[50]Agarwal, Manish & Mannem, Prashanth. (2011). Automatic gap-fill question generation from text books. 56-64.
[51]Schnabel, Tobias & Labutov, Igor & Mimno, David & Joachims, Thorsten. (2015). Evaluation methods for unsupervised word embeddings. 298-307. 10.18653/v1/D15-1036.
[52]Bakarov, Amir. (2018). A Survey of Word Embeddings Evaluation Methods.
[53]Lin, Hui & Bilmes, Jeff. (2011). A Class of Submodular Functions for Document Summarization.. 1. 510-520.