

Análisis de la técnica de aprendizaje por refuerzo para la generación automática de resúmenes

Máster Universitario en Ingeniería Informática



Trabajo Fin de Máster

Autor:

Carlos Martínez Rueda

Tutor/es:

Elena Lloret Pastor

Septiembre 2018



Universitat d'Alacant
Universidad de Alicante

Justificación y Objetivos

Últimamente, se está hablando largo y tendido y de forma sonada sobre el procesamiento del lenguaje natural (PLN) y hoy en día aún más debido al mundo digital cada vez más creciente.

Este campo de conocimiento de la Inteligencia Artificial se ocupa de investigar la manera de comunicar las máquinas con las personas mediante el uso de lenguajes naturales (aquellos utilizados en la comunicación humana, ya sean escritos, hablados o signados) cómo el español, inglés o chino (Moreno, 2017a).

Existen una gran variedad de aplicaciones para el PLN (Moreno, 2017b), de las cuales destacan:

- Comprensión del lenguaje.
- Recuperación de la información.
- Extracción de la información.
- Búsqueda de respuestas.
- Traducción automática.
- Reconocimiento del habla.
- Síntesis de voz.

Sin ir más lejos, los buscadores de internet cómo Google fueron la primera aplicación masiva del lenguaje natural en las tecnologías de la información donde estos nos permiten leer la información expresada en una lengua y extraer su contenido por medio de palabras clave que introducimos en el buscador.

Otra de las aplicaciones más utilizadas dentro del PLN es la generación automática de resúmenes de texto. Esto es debido a que, en la sociedad actual, la información juega un papel crucial que brinda ventajas competitivas a los usuarios cuando se maneja correctamente.

Sin embargo, debido a la gran cantidad de información disponible, los usuarios no pueden hacer frente a ella y, por lo tanto, los nuevos métodos y enfoques basados en el PLN son esenciales para procesar toda la información de manera efectiva y eficiente.

Las aplicaciones como recuperación de información (IR), respuesta a preguntas (QA) o resúmenes de texto (TS) pueden ayudar a los usuarios a acceder a la información más fácilmente, por un lado, reduciendo el tiempo que tienen que dedicar a la información, y por otro, seleccionando la información más útil para ellos (Lloret y Palomar, 2012).

Siguiendo esta línea se establecen los siguientes objetivos:

- Aprender nociones básicas sobre el procesamiento del lenguaje natural y de las tecnologías del lenguaje humano, en concreto sobre generación de lenguaje natural y generación de resúmenes de texto.
- Aprender conocimientos de las técnicas aprendizaje por refuerzo y aprendizaje por refuerzo profundo.
- Analizar y decidir las tecnologías más apropiadas para desarrollar el trabajo.
- Aprender y utilizar herramientas y librerías específicas de Procesamiento de Lenguaje Natural, así como frameworks para diseñar arquitecturas de aprendizaje automático (por ejemplo, Weka, Keras o TensorFlow).
- Evaluar el rendimiento del método propuesto tanto en términos cuantitativos como cualitativos.
- Realizar una comparación con respecto al estado de la cuestión.

Agradecimientos

Me gustaría agradecer a la Universidad y en concreto a la EPS y sus departamentos asociados todos los años que he pasado cursando tanto el grado de “Ingeniería Informática” como el máster de mismo nombre, así como los profesores que me han impartido alguna asignatura a lo largo de este tiempo.

Hacer una mención especial dentro de este agradecimiento global a profesores como Virgilio Gilart Iglesias, quién me motivó a seguir con mis estudios después del grado y así llegar a cursar el máster que culmina con este trabajo.

También me gustaría agradecer el duro trabajo de la tutora Elena Lloret Pastor, no sólo en el ámbito de este TFM, sino en las asignaturas cursadas impartidas por ella, así cómo la dedicación que ha tenido conmigo, no sólo por ser la tutora de este TFM, sino la ayuda que me ha proporcionado con asuntos relacionados con el máster.

Finalmente agradecer a Diego Mollá su ayuda tanto en el entendimiento de la temática relacionada con este TFM cómo en temas de implementación, sin la cual no podría haber realizado este TFM.

No quería acabar esta sección sin agradecer a mis familiares, amigos y compañeros su apoyo, sin el cual no hubiera podido llegar tan lejos en esta cruzada.

Índices

Índice general

Justificación y Objetivos	2
Agradecimientos.....	4
Índices.....	5
Índice general.....	5
Índice de figuras.....	6
Introducción	7
Objetivos.....	8
Marco teórico	9
Procesamiento del Lenguaje Natural.....	9
Generación de resúmenes automáticos.....	10
Aprendizaje por refuerzo	11
El aprendizaje profundo.....	13
Aprendizaje por refuerzo profundo.....	14
Representación de la información.....	14
Novedad y contribuciones	17
Metodología	18
Tarea BIOASQ.....	18
Base de la investigación	20
Enfoque propuesto	22
Tecnologías utilizadas	23
Experimentación y resultados	25
Pruebas preliminares	25
Pruebas realizadas	26
Evaluación y análisis de los resultados	28
Conclusiones.....	29
Trabajo futuro.....	30
Bibliografía y referencias	31

Índice de figuras

<i>Figura 01: Arquitectura general de un sistema de generación de resúmenes de texto.....</i>	<i>10</i>
<i>Figura 02: Esquema general del aprendizaje por refuerzo.....</i>	<i>12</i>
<i>Figura 03: Esquema ejemplo de una red neuronal con n entradas.....</i>	<i>14</i>
<i>Figura 04: Esquema de los enfoques de Word2vec (CBOW y Skip-gram).....</i>	<i>16</i>
<i>Figura 05: Tabla resumen de los datos disponibles.....</i>	<i>18</i>
<i>Figura 06: Esquema del funcionamiento del sistema.....</i>	<i>20</i>
<i>Figura 07: Representación de la frase de prueba en el modelo por CBOW.....</i>	<i>25</i>
<i>Figura 08: Representación de la frase de prueba en el modelo por Skip-gram.....</i>	<i>26</i>
<i>Figura 09: Resultados Rouge-L al utilizar CBOW.....</i>	<i>27</i>
<i>Figura 10: Resultados Rouge-L al utilizar Skip-gram.....</i>	<i>27</i>
<i>Figura 11: Resultados Rouge-L al utilizar TFIDF.....</i>	<i>28</i>

Introducción

La Inteligencia Artificial (IA) es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente (Romero, 2007).

Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que, si fuera llevado a cabo por una persona, se diría que es inteligente. El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente.

Tanto la técnica de aprendizaje por refuerzo cómo el procesamiento del lenguaje natural del cual se extrae la generación automática de resúmenes de texto se engloba dentro de esta rama de la ciencia.

En concreto, las tareas relacionadas con el PLN y sobre todo la generación automática de resúmenes cobran un papel cada vez más relevante en una sociedad caracterizada por la información. A los usuarios, cada vez se les proporcionan más datos y estos los demandan aún más, sobre todo hoy en día donde la transparencia se ha convertido en un requisito imperativo. De esta forma, el usuario puede acceder a toda esta información pero de la forma y modo en el que desea, sin obtener todos los datos y ser él mismo quien los organice.

Del hecho de combinar ambas técnicas se obtiene la generación automática de resúmenes llevada a cabo y perfeccionada por medio de un aprendizaje por refuerzo positivo o negativo cómo si de una persona se tratara.

Por tanto, en este trabajo se pretende la creación de un sistema que por medio del aprendizaje por refuerzo sea capaz de llevar a cabo tareas de generación de resúmenes de forma automática.

La tarea específica que deberá realizar el sistema será la generación automática de respuestas a preguntas en el ámbito biomédico, utilizando como base de la investigación la tarea BioASQ 2017 (explicada con mayor detalle en el apartado "*Metodología*").

Objetivos

Partiendo de dicha tarea BioASQ, y en concreto su apartado de contestación de preguntas y partiendo del concepto de crear un sistema para ello por medio de aprendizaje por refuerzo, se establece los siguientes objetivos específicos:

- Investigar en que consiste la tarea BioASQ en su versión de 2017 y comprender el funcionamiento de los ficheros proporcionados.
- Investigar la implementación que mejores resultados obtuvo en la tarea previamente menciona que se encuentra disponible en Github (<https://github.com/dmollaaliod/bioasq-rl>) y fue desarrollada por Diego Mollá. Se investigará cómo funciona, y las tecnologías y métodos que utiliza.
- Realizar variaciones en la implementación con el fin de comparar diferentes técnicas para la resolución de dicho problema. En concreto la sustitución de la técnica TFIDF para el cálculo del valor de las palabras por el uso de Word Embedding.
- Investigar en variaciones de la forma de realizar el aprendizaje por refuerzo profundo distintas a la implementación basada en políticas por descenso de gradiente (Policy Gradients).

Marco teórico

Teniendo en cuenta el problema que se desea abordar y la forma en la que se desea resolver, el primer paso consiste en tratar el aspecto del lenguaje natural para identificar y explicar los aspectos relacionados con el fin llevar a cabo la generación de resúmenes.

Procesamiento del Lenguaje Natural

El lenguaje natural (Martín Mateos y Ruiz Reina, 2013), entendido como la herramienta que utilizan las personas para expresarse, posee propiedades que afectan a su procesamiento. Estas propiedades son la variación y la ambigüedad lingüística. Cuando hablamos de la variación lingüística nos referimos a la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea. En cambio, la ambigüedad lingüística se produce cuando una palabra o frase permite más de una interpretación.

Ambos fenómenos inciden en el proceso de recuperación de información, aunque de forma distinta. La variación lingüística provoca el silencio documental, es decir la omisión de documentos relevantes para cubrir la necesidad de información, ya que no se han utilizado los mismos términos que aparecen en el documento. En cambio, la ambigüedad implica el ruido documental, es decir la inclusión de documentos que no son significativos, ya que se recuperan también documentos que utilizan el término, pero con significado diferente al requerido. Estas dos características dificultan considerablemente el tratamiento automatizado del lenguaje.

La arquitectura de un sistema de PLN se sustenta en una definición del lenguaje natural por niveles:

- Fonológico: trata de cómo las palabras se relacionan con los sonidos que representan.
- Morfológico: trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas.
- Sintáctico: trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración y que sintagmas son parte de otros sintagmas.

- Semántico: trata del significado de las palabras y de cómo los significados se unen para dar significado a una oración, también se refiere al significado independiente del contexto, es decir de la oración aislada.
- Pragmático: trata de cómo las oraciones se usan en distintas situaciones y de cómo el uso afecta al significado de las oraciones.
 - Subnivel recursivo: discursivo, que trata de cómo el significado de una oración se ve afectado por las oraciones inmediatamente anteriores.

Generación de resúmenes automáticos

Entendemos por resumen la transformación reductiva de un texto fuente a un texto resumen por reducción de su contenido mediante selección y/o generalización del contenido relevante del texto fuente (Cardoso y Pérez, 2013).

Por generación automática de resúmenes de texto entendemos el proceso por el cual se identifica la información sustancial proveniente de una fuente (o varias) para producir una versión abreviada destinada a un usuario particular (o grupo de usuarios) y una tarea (o tareas).

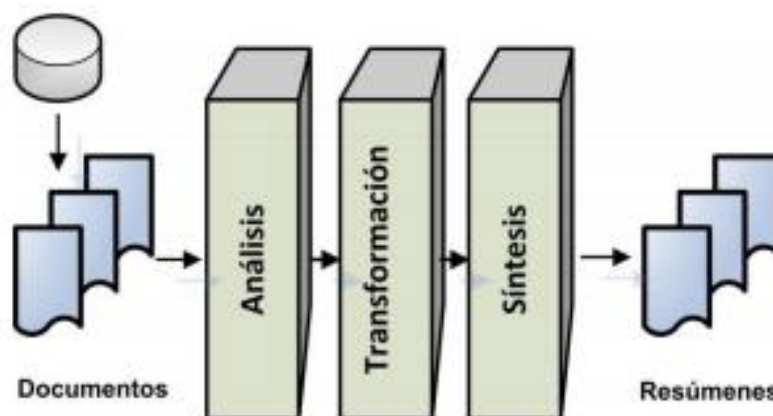


Figura 01: Arquitectura general de un sistema de generación de resúmenes de texto.

Atendiendo al alcance, el resumen puede limitarse a un único documento o a un conjunto de ellos que traten sobre el mismo tema (ACERO, 2001).

Según su propósito los resúmenes se clasifican en:

- Indicativos: anticipar al lector el contenido del texto y ayudarle a decidir sobre la relevancia del documento original.

- Informativos: sustituir al texto completo incorporando toda la información nueva o trascendente.
- Críticos: incorporar opiniones o comentarios que no aparecen en el texto original.

Finalmente, atendiendo al enfoque, podemos distinguir entre resúmenes:

- Genéricos: recogen los temas principales del documento y van destinados a un grupo amplio de personas.
- Adaptados al usuario: el resumen se confecciona de acuerdo con los intereses del lector o grupo de lectores al que va dirigido.

Aunque existen varias técnicas para llevar a cabo este proceso de generación de resúmenes de forma automática, la elegida para llevar a cabo el trabajo es el uso de la técnica denominada aprendizaje por refuerzo, la cuál se va a estudiar y aplicar en la creación del sistema.

Aprendizaje por refuerzo

El aprendizaje por refuerzo (*Reinforcement Learning*) consiste en un área del aprendizaje automático inspirada en la psicología conductista, cuya ocupación es determinar qué acciones debe escoger un agente de software en un entorno dado con el fin de maximizar una "recompensa" (Harmon y Harmon, 1996). Además, presenta una serie de características concretas (Sempere, 2013):

- Es completo (contempla el problema en su totalidad, mediante funciones objetivo).
- Es interactivo (recibe información del entorno y puede modificar el mismo).
- Es no supervisado (sustituye la información supervisada por información del tipo acción/reacción).
- Es dirigido por objetivos (el fin del aprendizaje es alcanzar un máximo).
- El agente debe explotar las acciones que le beneficien y explorar nuevas acciones.

Los elementos que presenta el aprendizaje por refuerzo son:

- La política que define el comportamiento del agente en cada momento.
- La función de recompensa que define el objetivo a alcanzar.

- La función de evaluación que permite establecer la respuesta recibida a partir de cada posible acción.

El objetivo del aprendizaje por refuerzo es extraer qué acciones deben ser elegidas en los diferentes estados para maximizar la recompensa. En cierta forma, buscamos que el agente aprenda lo que se denomina una política, que formalmente podemos verla como una aplicación que dice en cada estado qué acción tomar. Esta política se divide en dos partes: por una parte, cómo de buena cree el agente que es una acción sobre un estado determinado y, por otra, cómo usa el agente lo que sabe para elegir una de las acciones posibles (Sancho, 2017a).

La forma en la que el sistema toma sus decisiones sigue un proceso de decisión basado en los modelos de Markov donde:

- El agente percibe un conjunto finito de estados distintos en su entorno y dispone de un conjunto finito de acciones para interactuar con él.
- El tiempo avanza de forma discreta, donde en cada instante de tiempo, el agente percibe un estado concreto y selecciona una de las posibles acciones y la ejecuta, obteniendo un nuevo estado.
- El entorno responde al agente por medio de una recompensa.
- Tanto la recompensa como el estado siguiente no tienen por qué ser conocidos por el agente y dependen únicamente del estado actual y de la acción tomada.

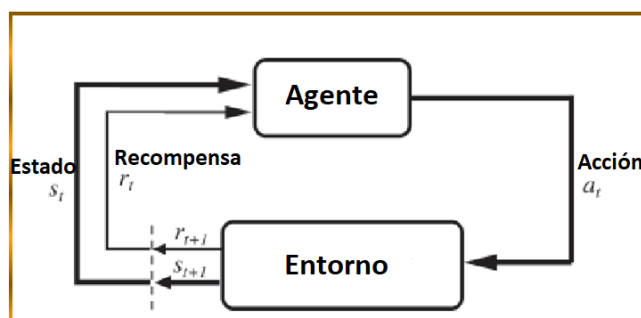


Figura 02: Esquema general del aprendizaje por refuerzo.

La técnica del aprendizaje por refuerzo es una técnica clásica para la toma de decisiones, pero en los últimos años, debido al auge computacional del aprendizaje profundo y de las redes neuronales, ha surgido una variante denominada aprendizaje por refuerzo profundo, la cuál consiste en la combinación de las redes neuronales con el aprendizaje por refuerzo.

El aprendizaje profundo

El aprendizaje profundo (Deep Learning) consiste en el hecho de llevar a cabo el aprendizaje automático por medio del uso de una red neuronal artificial. Estas Redes Neuronales Artificiales (RNA) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas y que constan de un gran número de elementos simples de procesamiento llamados nodos o neuronas organizadas en capas. Las RNA son conocidas dentro del área de la inteligencia artificial como redes de neuronas simplemente (Palmer y Montaña, 1999).

Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. Los pesos representan la información que será usada por la red neuronal para resolver un problema determinado.

De esta forma, las RNA son sistemas adaptativos que aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Mediante este entrenamiento o aprendizaje, las RNA crean su propia representación interna del problema. Posteriormente, pueden responder adecuadamente cuando se les presentan situaciones a las que no habían sido expuestas con anterioridad.

En la Figura 03 se puede apreciar un ejemplo de una red neuronal con n entradas, que consta de (Sancho, 2017b):

- Un conjunto de n entradas X_1, \dots, X_n .
- Los pesos correspondientes a cada entrada w_1, \dots, w_n .
- Una función de agregación, Σ .
- Una función de activación, f .
- Una salida, Y .

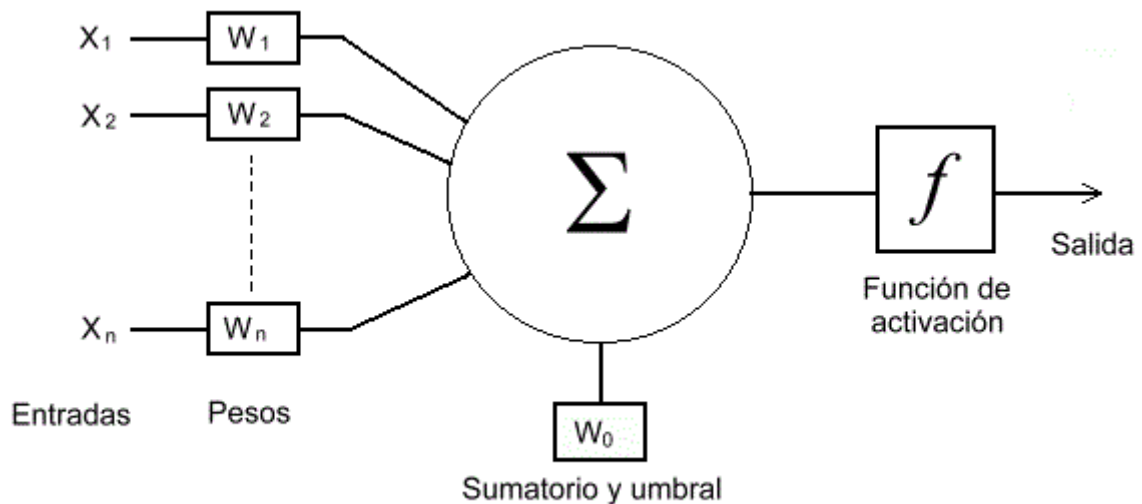


Figura 03: Esquema ejemplo de una red neuronal con n entradas.

Mediante el uso de las redes neuronales, junto con la técnica del aprendizaje por refuerzo, obtenemos una versión mejorada de este denominada aprendizaje por refuerzo profundo (Deep Reinforcement Learning)

Aprendizaje por refuerzo profundo

Si combinamos el aprendizaje por refuerzo (Reinforcement Learning) junto con el uso de redes neuronales artificiales (*Deep Learning*) en sus procesos y algoritmos (entrenamiento y decisión), obtenemos lo que se conoce como aprendizaje por refuerzo profundo (*Deep Reinforcement Learning*) (Li, 2017).

De esta forma, si se utilizan las redes neuronales para representar la función de valor, la política del agente o el modelo del aprendizaje por refuerzo, se obtiene el Deep Reinforcement Learning.

Para llevar a cabo este proceso, se deberá decidir como se representa la información que va a tratar el sistema.

Representación de la información

Para poder utilizar los datos de texto en algoritmos de aprendizaje automático, necesitamos una forma de representar dicha información que pueda entender y utilizar el sistema.

En problemas relacionados con el PLN, se utiliza el concepto de Bag-of-Words para referirse al modelo utilizado para extraer las características de las palabras (D'Souza, 2018). Así se obtiene una representación de la información entendible y usable para los algoritmos de aprendizaje automático. Las representaciones más utilizadas son:

- CountVectorizer: Consiste en contar el número de veces que cada palabra aparece en el conjunto de documentos y calcula su frecuencia.
- TF-IDF Vectorizer: Esta puntuación viene determinada por el producto de multiplicar la frecuencia de una palabra (TF) y su inversa (IDF) dentro del conjunto de documentos.
- Word Embeddings: Cada palabra está representada por un vector teniendo en cuenta su ocurrencia y la información de co-ocurrencia (relación de proximidad de dos o más términos en una unidad de texto) (Naryani, 2017).

Las Word Embeddings son, de hecho, una clase de técnicas donde las palabras individuales se representan como vectores de valores reales en un espacio vectorial predefinido y donde las palabras que tienen el mismo significado tienen una representación similar (Brownlee, 2017a).

Cada palabra se asigna a un vector y los valores del vector se aprenden de una manera que se asemeja a una red neuronal. La clave del enfoque es la idea de usar una representación distribuida densa para cada palabra. Cada palabra está representada por un vector de valor real, a menudo decenas o cientos de dimensiones.

Word Embeddings aprende en función del uso y contexto de las palabras. Esto permite que las palabras que se usan de manera similar den como resultado una representación similar, con el fin de capturar de forma natural su significado.

Hay una teoría lingüística más profunda detrás de este enfoque, es decir, la "hipótesis de distribución" de Zellig Harris (Harris, 1954) que podría resumirse como: las palabras que tienen un contexto similar tendrán significados similares.

Los métodos con Word Embeddings aprenden una representación vectorial de valor real para un vocabulario de tamaño fijo predefinido a partir de un corpus de texto. Cómo método para implementar Word Embedding se utiliza "word2vec" pero existen otros como

GloVe (Global Vector for Word Representation). Se elige el uso de “word2vec” por ser el algoritmo base y principal desarrollado para implementar Word Embeddings.

Word2Vec es un método estadístico para aprender eficientemente Word Embeddings independiente a partir de un corpus de texto. Fue desarrollado por Tomas Mikolov, et al. (2013) en Google como respuesta para hacer más eficiente el proceso de captación de significado.

Existen dos modelos de aprendizaje diferentes que se pueden usar como parte del enfoque de word2vec para aprender Word Embedding:

- Modelo Continuo Bag-of-Words, o modelo CBOW.
- Modelo Skip-Gram.

El modelo CBOW aprende el significado al predecir la palabra actual en función de su contexto y el modelo de Skip-Gram aprende al predecir el significado de las palabras circundantes dada una palabra actual (contexto en función de la palabra actual).

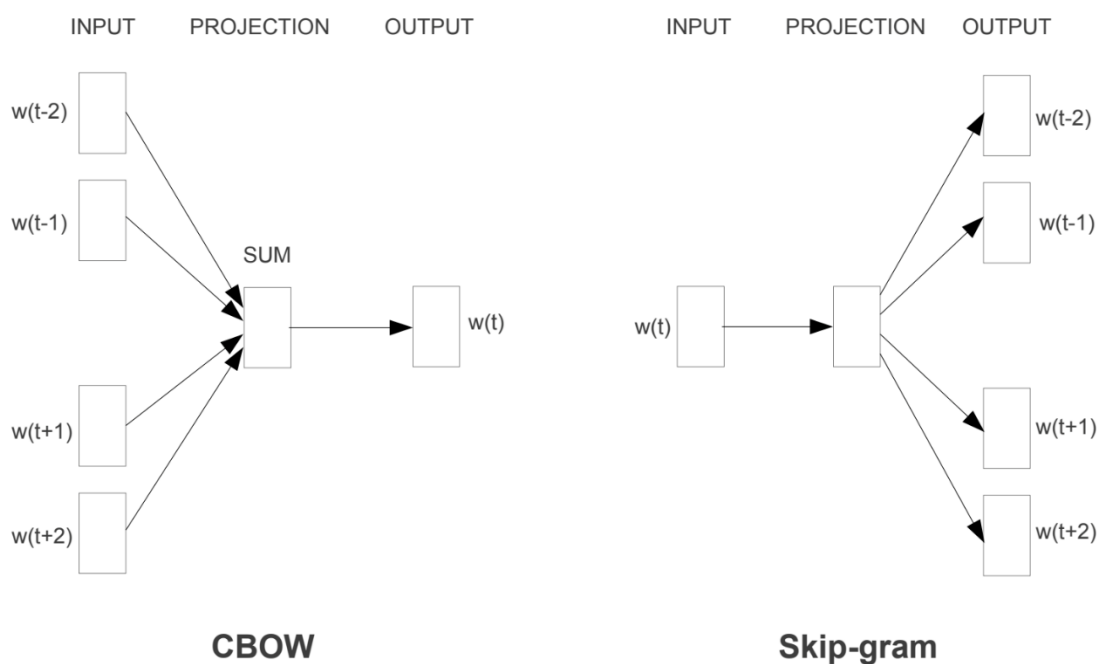


Figura 04: Esquema de los enfoques de Word2vec (CBOW y Skip-gram).

Según el propio Mikolov: Skip-gram funciona bien con una pequeña cantidad de datos de entrenamiento, representando bien incluso palabras o frases raras y CBOW es más rápido

de entrenar que Skip-gram y presenta una precisión ligeramente mejor para las palabras frecuentes (Sim, 2017).

Novedad y contribuciones

Junto al sistema de aprendizaje por refuerzo profundo se añadirá el uso de Word Embeddings como método para representar la información, con el objetivo de poder determinar si esta representación de la información ayuda o no y de que forma en tareas de PLN.

En concreto, el sistema llevará a cabo la tarea BioASQ de respuesta a preguntas, generando una respuesta en forma de resumen que conteste a preguntas en el ámbito biomédico mediante el uso de Word2vec para la representación de la información y se comparará con un sistema similar, pero que usa el método clásico de TFIDF.

De esta forma se sustituye el valor clásico de las palabras (TFIDF) basado en el número de veces que aparece la palabra en el conjunto de documentos por una representación vectorial de dicha palabra que refleja en mayor medida el significado que tiene dentro del contexto.

Partiendo de la base de la implementación realizada por Diego Mollá que resuelve esta tarea y que se encuentra disponible en GitHub (<https://github.com/dmollaaliiod/bioasq-rl>), se realizarán estudios y modificaciones para ver como nuevas tecnologías o cambios en la forma de actuar afectan al resultado.

Metodología

Tarea BIOASQ

La base sobre las que se ha desarrollado este proyecto han sido la tarea BioASQ (Bioasq.org, 2018), en su versión de 2017 (5ª edición).

BioASQ organiza retos en el campo de la indexación semántica biomédica y en la contestación de preguntas. De estos dos retos, el que se trata en este proyecto es el de la contestación de preguntas correspondiente a la tarea 5b: “*Biomedical semantic question answering*”. En esta tarea hay presentes 4 tipos de preguntas: “si/no”, “factoid”, “lista” y “resumen”.

Tipo	Número	Media de fragmentos disponibles
si/no	500	22,16
factoid	486	21,27
lista	413	18,34
resumen	400	19,19
global	1799	20,38

Figura 05: Tabla resumen de los datos disponibles.

Esta tarea utiliza conjuntos de datos de referencia que contienen preguntas de desarrollo y prueba, en inglés, junto con respuestas estándar generadas por un equipo de expertos biomédicos. Los participantes tienen que responder con conceptos relevantes, artículos, fragmentos y tripletas RDF, de recursos designados, así como respuestas exactas e ideales.

Para realizar dicha tarea se proporciona un archivo JSON estructurado con la información necesaria para realizar la tarea. Este fichero contiene una lista de elementos sobre los que se extrae información para contestar a la pregunta, la pregunta en cuestión y un conjunto de respuestas ideales (puede ser una respuesta o varias).

A continuación, se puede ver un ejemplo de una pregunta dentro de este fichero en formato JSON donde se ha acotado el número de “snippets” (fragmentos de texto) a 1 para reducir su longitud y que así sea más entendible y visible.

```
{
  "triples": [],
  "concepts": [
    "http://www.disease-ontology.org/api/metadata/DOID:10487",
    "http://www.nlm.nih.gov/cgi/mesh/2015/MB_cgi?field=uid&exact=Find+Exact+Term&term=D006627",
    "http://www.disease-ontology.org/api/metadata/DOID:11372",
    "http://www.nlm.nih.gov/cgi/mesh/2015/MB_cgi?field=uid&exact=Find+Exact+Term&term=D020412"
  ],
  "documents": [
    "http://www.ncbi.nlm.nih.gov/pubmed/15829955",
    "http://www.ncbi.nlm.nih.gov/pubmed/15617541",
    "http://www.ncbi.nlm.nih.gov/pubmed/12239580",
    "http://www.ncbi.nlm.nih.gov/pubmed/8896569",
    "http://www.ncbi.nlm.nih.gov/pubmed/6650562",
    "http://www.ncbi.nlm.nih.gov/pubmed/20598273",
    "http://www.ncbi.nlm.nih.gov/pubmed/21995290",
    "http://www.ncbi.nlm.nih.gov/pubmed/23001136"
  ],
  "snippets": [
    {
      "offsetInBeginSection": 131,
      "endSection": "abstract",
      "document":
        "http://www.ncbi.nlm.nih.gov/pubmed/15829955",
      "offsetInEndSection": 358,
      "text":
        "Hirschsprung disease (HSCR) is a multifactorial, non-mendelian disorder in which rare high-penetrance coding sequence mutations in the receptor tyrosine kinase RET contribute to risk in combination with mutations at other genes",
      "beginSection": "abstract"
    }
  ],
  "ideal_answer": [
    "Coding sequence mutations in RET, GDNF, EDNRB, EDN3, and SOX10 are involved in the development of Hirschsprung disease. The majority of these genes was shown to be related to Mendelian syndromic forms of Hirschsprung's disease, whereas the non-Mendelian inheritance of sporadic non-syndromic Hirschsprung disease proved to be complex; involvement of multiple loci was demonstrated in a multiplicative model."
  ],
  "id": "55031181e9bde69634000014",
}
```

```

"body":
  "Is Hirschsprung disease a mendelian or a multifactorial
  disorder?",
"type": "summary"
}

```

Para la tarea 5b, la información a utilizar son los “*snippets*” o fragmentos de donde se obtendrán la información para construir el resumen o respuesta, la “*ideal_answer*” que almacena la respuesta ideal a la pregunta y el “*body*” que contiene la pregunta en sí.

De modo que el sistema debería actuar cómo una persona que tuviera los conocimientos correspondientes a los “*snippets*” y contestar a la pregunta con la respuesta ideal tal y como se refleja en la figura 06.

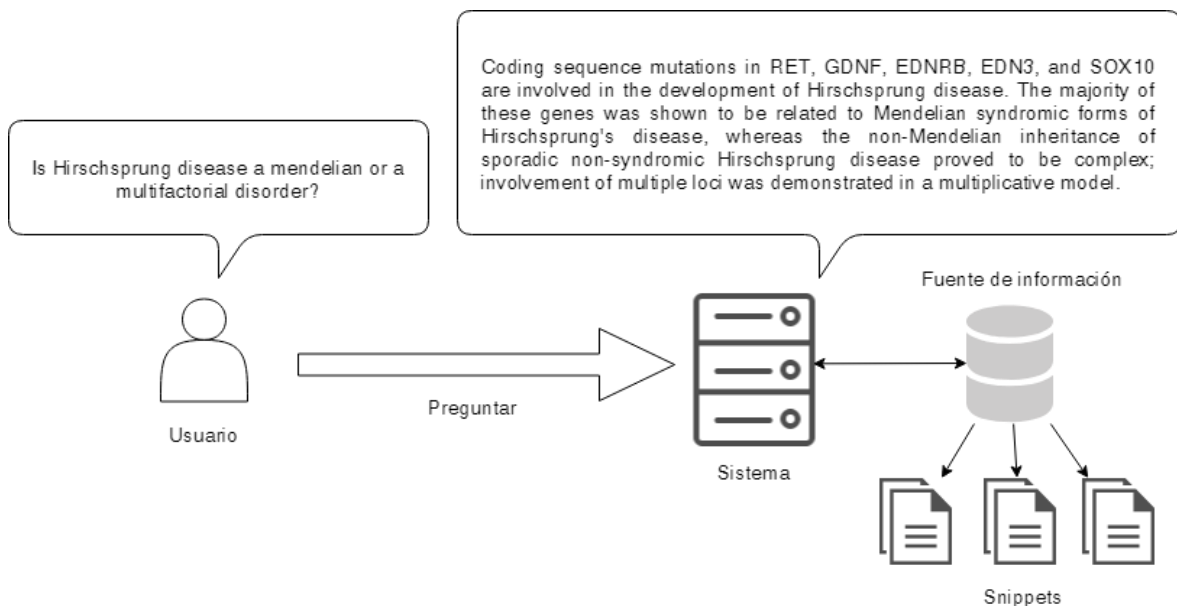


Figura 06: Esquema del funcionamiento del sistema.

Base de la investigación

En la implementación que mejores resultados obtuvo en la tarea 5b disponible en Github (<https://github.com/dmollaaliod/bioasq-rl>) y realizada por Diego Mollá (Mollá, 2017), se utiliza aprendizaje por refuerzo junto con una red neuronal donde:

- El agente necesita decidir si la frase *i* es parte del resumen o no.
- La recompensa se retrasa hasta que todas las posibles frases hayan sido procesadas.
- La recompensa del resumen es su puntuación en Rouge (determina cómo de bueno es el resumen comparándolo con el ideal), en concreto su versión Rouge-L (tiene en cuenta palabras con un solo significado común, ej: Nueva York) (Lin, 2004).

- El agente necesita aprender una política para la toma de decisiones.

Esta política se aprende de la siguiente forma:

- Se utiliza un conjunto del total de frases candidatas para aprender una política global.
- La política global determina la probabilidad de que la frase i proporcione la máxima recompensa.
- Se implementa por medio de una red neuronal con una capa oculta.
- La red se entrena mediante un descenso de gradiente.

Siguiendo esta lógica, utiliza el valor TFIDF como puntuación individual de cada palabra dentro del corpus como base para la red neuronal.

De esta forma y utilizando la librería de Python “TfidfVectorizer”, se calcula la puntuación de las frases que forman el corpus por medio de la puntuación TFIDF de las palabras que conforman dichas frases.

Como resultado se obtiene una matriz con las puntuaciones asociadas a cada palabra de tamaño fijo (el correspondiente a todas las diferentes palabras presentes en el corpus) y donde la mayoría de los elementos son 0 (palabras que no aparecen en la frase objetivo, pero que si se encuentran en el corpus).

De esta forma se construyen dos listas:

- Lista con la matriz correspondiente de todas las frases de entrenamiento transformada en un vector 2-dimensional (llamado representación densa).
- Lista con la matriz correspondiente a todas las frases candidatas para la pregunta transformada en un vector 2-dimensional.

Ambas listas se concatenan utilizando la función “numpy.hstack” de Python para poder pasarlas como el primer parámetro de la red neuronal.

El segundo parámetro de la red neuronal es la matriz de puntuaciones TFIDF correspondiente a la pregunta en forma de lista 2-dimensional.

Adicionalmente, en la etapa de entrenamiento de la política, se le proporcionará a la red el valor Rouge-L de las respuestas candidatas como parte del aprendizaje por refuerzo.

La red presente en el sistema es del tipo LSTM (Long short-term memory) perteneciente al tipo de redes neuronales recurrentes (RNN) por su idoneidad en tareas de PLN debido a su capacidad de recordar información durante largos periodos como comportamiento por defecto (ideal para secuencias de texto).

Enfoque propuesto

La variación propuesta a la implementación base explicada en el apartado anterior es el cambio del cálculo de las puntuaciones de las frases candidatas de TFIDF a Word Embeddings, donde cada palabra tendrá como puntuación el vector resultante de calcular su significado dentro del corpus. Esta variación por medio de Word Embedding es la implementación llevada a cabo cómo base para este trabajo de investigación.

Para la generación del modelo, se utiliza la librería “gensim” (librería Python de código abierto para PLN), en concreto su función para Word Embeddings basada en Word2vec denominada “word2vec” (Brownlee, 2017b). Aunque “word2vec” te permite cargar modelos predefinidos, el único gran modelo disponible es el de Google News (<https://github.com/mmhaltz/word2vec-GoogleNews-vectors>) y trata todo tipo de temáticas, no siendo valido a un dominio tan concreto (biomédico) y con un vocabulario muy específico.

Esta función corresponde a un constructor del modelo con una gran cantidad de parámetros, pero los más relevantes y que afectan significativamente a su funcionamiento y resultado son los siguientes:

- Size: (por defecto 100) Número de dimensiones del “embedding” (tamaño del vector que representa cada palabra).
- Window: (por defecto 5) Distancia máxima entre la palabra objetivo y las palabras que la rodean (contexto).
- Min_count: (por defecto 5) Mínima número de ocurrencias de una palabra en el corpus para tenerla en cuenta.

- Workers: (por defecto 3) Número de hilos utilizados durante el entrenamiento (rendimiento del constructor).
- Sg: (por defecto 0 o CBOW) El modelo utilizado para determinar el valor de las palabras. Variable entre CBOW (0) y Skip-gram (1).

Para este trabajo de fin de máster se utilizan los valores por defecto, a excepción del “Sg” que se modificará para realizar pruebas, de modo que se evaluarán ambos modelos sobre el mismo corpus.

Una vez ejecutado el constructor y obtenido el modelo sobre el que se trabajara para obtener las puntuaciones de las frases, se crean las listas de matrices con las puntuaciones asociadas a cada palabra correspondientes a dichas frases, obteniendo las mismas listas que en el caso anterior pero cuya puntuación corresponde al valor de “word2vec” de las palabras.

Al igual que en el caso anterior, los parámetros de la red son los mismos y también se transforman a vectores 2-dimensionales para que la red neuronal pueda trabajar con ellos.

Tecnologías utilizadas

Para llevar a cabo las implementaciones, se utiliza como lenguaje de programación Python3 en su versión 3.4.8 (<https://www.python.org/downloads/release/python-348/>) junto con un conjunto de librerías y herramientas específicas dentro del área del aprendizaje automático:

- NumPy y SciPy: Librerías fundamentales para machine learning que permiten trabajar con vectores N-dimensionales y computación de valores (<http://www.numpy.org> y <https://www.scipy.org>).
- NLTK: Librería para construir programas con procesamiento de lenguaje natural (<https://www.nltk.org>).
- Scikit-learn (sklearn): Librería para minería y análisis de datos (<http://scikit-learn.org>)
- Gensim: Librería para el Procesamiento de Lenguaje Natural (PLN) y Recuperación de Información (RI). En concreto utilizada por tener Word2vec (<https://pypi.org/project/gensim>).

- Tensorflow: Framework de código abierto para realizar cálculos numéricos mediante diagramas de flujo de datos (red neuronal) (<https://www.tensorflow.org>).
- Rouge: Librería Python con la implementación de la métrica Rouge (<https://github.com/pltrdy/rouge>).
- Matplotlib: Librería para el trazado 2D de valor y su representación en figuras de diferentes formas donde se extrae “matlib” (<https://matplotlib.org/>).

Experimentación y resultados

Pruebas preliminares

Con el fin de entender el funcionamiento de word2vec y de cómo se generan los vectores de puntuaciones de las palabras, se realizan pruebas basadas en el análisis de los componentes principales (PCA) mediante el uso de las librerías “*matlib*” y “*pca*” de Python. De esta forma se puede obtener una representación visual de dicha puntuación.

Con “*pca*” obtenemos una representación 2-dimensional del vector correspondiente a la frase que se representará mediante “*matlib*”. Así la información se puede visualizar en una gráfica representativa de las palabras que forman la frase dentro del modelo de word2vec global realizado con el corpus. En las siguientes figuras (Figura 07 y 08) podemos observar el resultado de la frase “*A sulfated glycoprotein was isolated from the culture media of Drosophila Kc cells and named papilin*” en los dos modelos de word2vec (CBOW y Skip-gram).

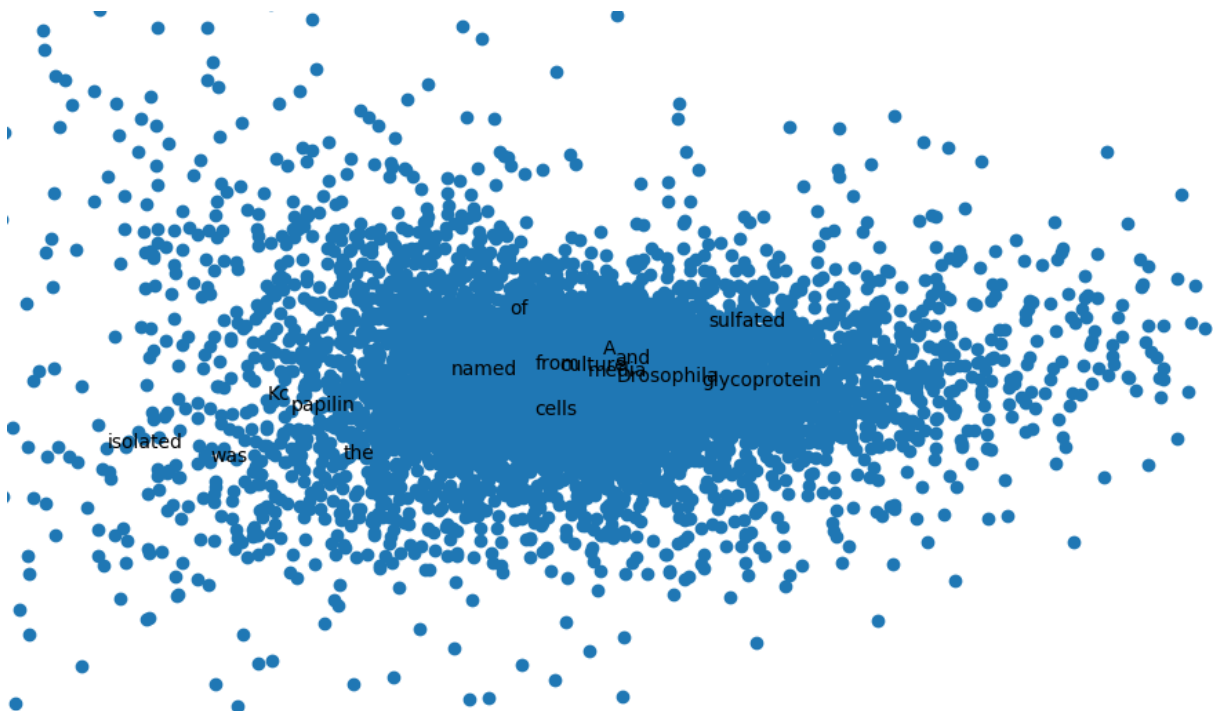


Figura 07: Representación de la frase de prueba en el modelo por CBOW.

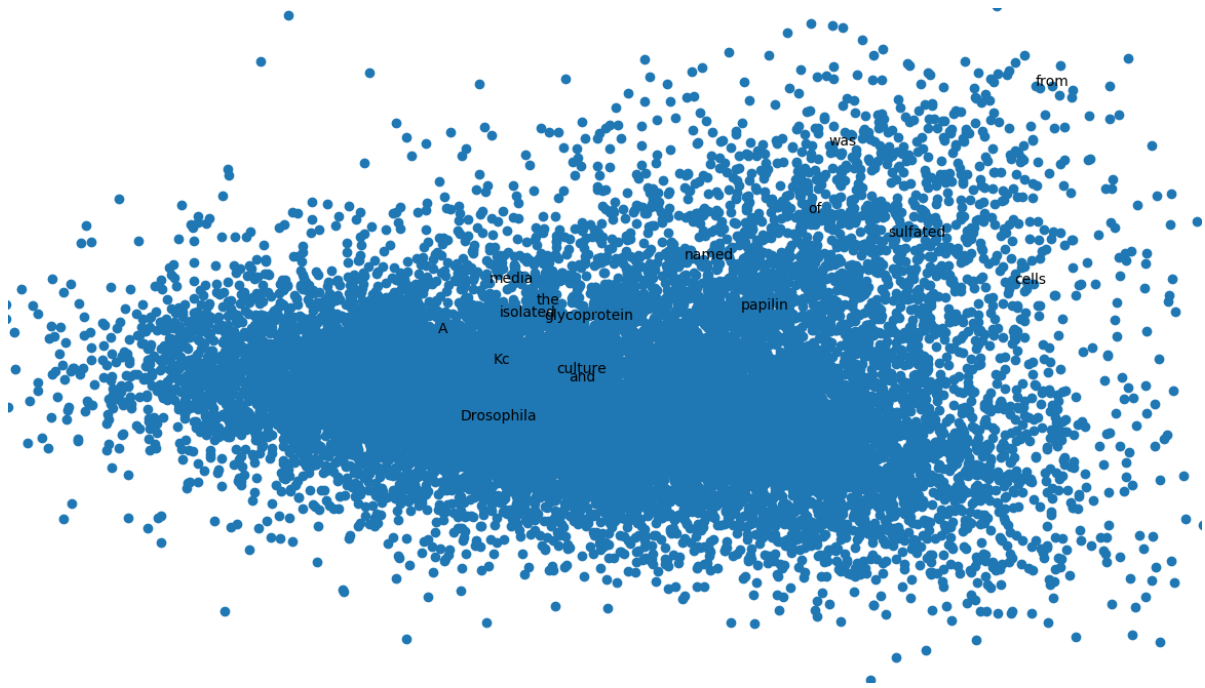


Figura 08: Representación de la frase de prueba en el modelo por Skip-gram.

En ambas representaciones de los modelos se destacan las palabras correspondientes a la frase objetivo dentro del conjunto. Además, se observa que en el caso de Skip-gram el conjunto total queda más acotado, lo que se corresponde con las afirmaciones mencionadas con anterioridad sobre este modelo (representa bien palabras raras y trabaja mejor con conjuntos de datos pequeños).

Pruebas realizadas

Para ejecutar el sistema, se utilizará un 80% del total como conjunto de datos de prueba y el 20% restante para evaluación, de modo que se irán obteniendo los valores Rouge-L correspondientes a comparar los resúmenes generados con la respuesta ideal.

A medida que avance el número de episodios, el sistema irá perfeccionando la política simplificada en decidir si una frase candidata forma parte del resumen o no, teniendo en cuenta el valor obtenido de comparar la respuesta generada con la idea. En las siguientes figuras (Figura 09 y 10) podemos observar el resultado generado para 2120 episodios.

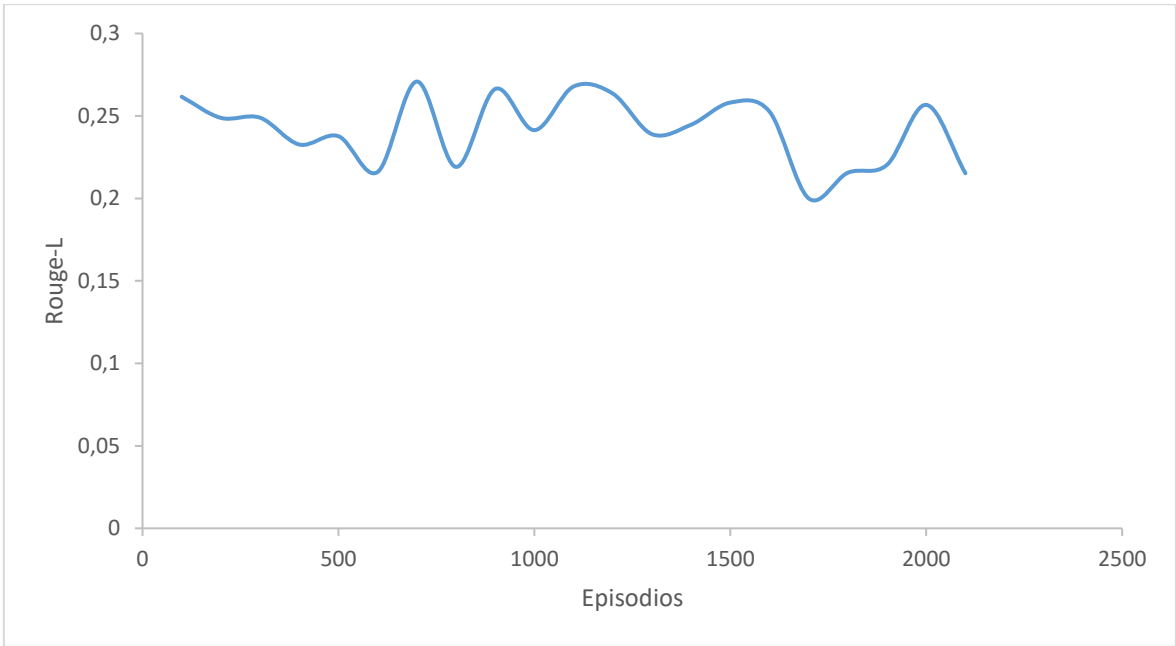


Figura 09: Resultados Rouge-L al utilizar CBOW.

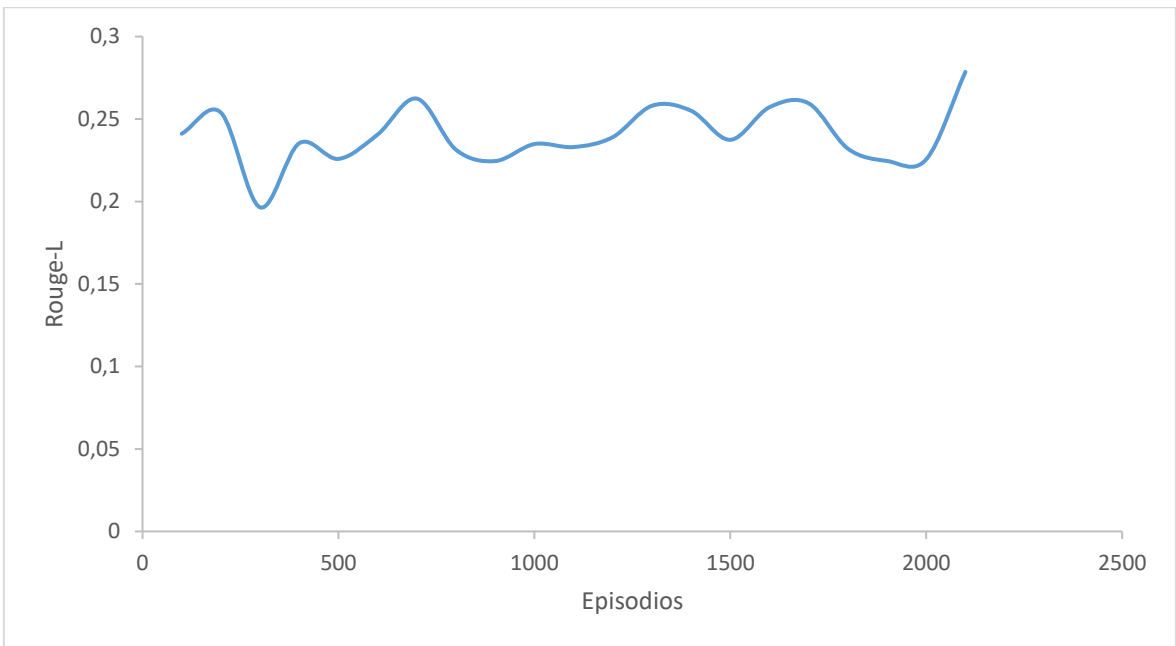


Figura 10: Resultados Rouge-L al utilizar Skip-gram.

Teniendo en cuenta los valores resultantes, a simple vista no se puede determinar su viabilidad. Pero si calculamos la recta de regresión resultante a la nube de puntos de cada modelo, podremos determinar de forma predictiva el comportamiento. Así pues, necesitaremos calcular la pendiente de dicha recta en función de los valores X e Y de nuestras gráficas (Número de episodios y puntuación Rouge-L respectivamente).

Los valores obtenidos para las pendientes de ambos modelos son: -0.0000034 en el caso de CBOW y de 0.000004 en el caso de Skip-gram. De estos valores podemos interpretar que el comportamiento del modelo CBOW es descendiente al presentar una pendiente negativa pero en el caso de Skip-gram, el comportamiento es ascendente por presentar una pendiente ascendente, lo que demuestra la idoneidad de este modelo para el conjunto de datos sobre el que se trabaja.

Evaluación y análisis de los resultados

En la implementación que obtuvo los mejores resultados en la tarea 5b disponible en Github(<https://github.com/dmollaaliad/bioasq-rl>) por medio de TFIDF, los resultados obtenidos por el sistema son (datos obtenidos del artículo (Mollá, 2017)):

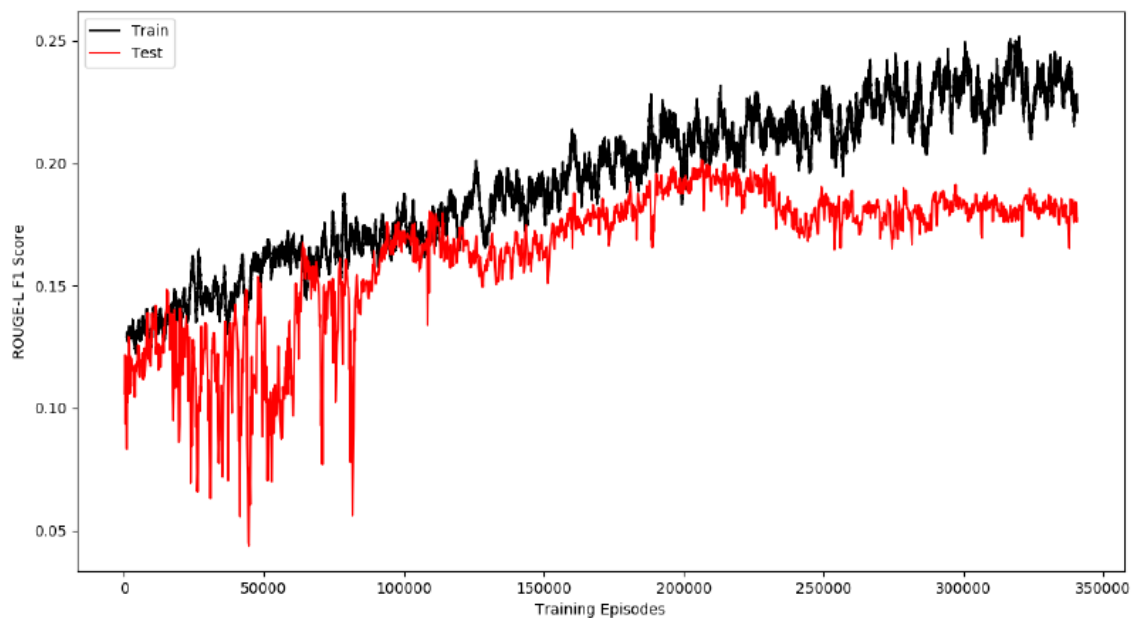


Figura 11: Resultados Rouge-L al utilizar TFIDF.

Claramente podemos observar como los valores Rouge-L en las etapas tempranas del sistema son significativamente menores a los obtenidos por la implementación mediante Word Embeddings, aunque dichos valores se acercan a los obtenidos pasados los 250.000 episodios y habría que llegar a ese número de episodios con el sistema de Word Embeddings para determinar de forma precisa si a la larga también es mejor.

Conclusiones

En el ámbito del PLN y sobre todo dentro de la importancia de las palabras en un contexto, la aproximación de TFIDF para medir dicha importancia resulta muy básica para ámbitos específicos como la medicina. A pesar de ello, sirve gratamente a su propósito como base con la que realizar implementaciones, en este caso de aprendizaje por refuerzo para la generación de resúmenes.

Sin embargo, técnicas más sofisticadas y depuradas como Word Embeddings dan mayor fuerza a las palabras y mayor significado dentro del contexto, pero requieren de un gran conjunto de datos para que dicha precisión supere a la aproximación TFIDF.

Dentro de las Word Embeddings, el uso del paradigma Skip-gram en nuestro caso, aporta un mayor valor a las palabras, acotando las puntuaciones y dando una puntuación más realista dentro del ámbito en el que se trabaja (las palabras tienen un significado más relacionado con la temática).

Mediante el uso de Word Embeddings se obtienen unos valores iniciales de Rouge-L mayores que los correspondientes de TFIDF y cuya recta de regresión indica que los valores futuros serán más elevados a los obtenidos, de modo que se puede prever que se obtendrán mejores resultados que los obtenidos por TFIDF.

Trabajo futuro

Como trabajo futuro a esta investigación, se plantea la búsqueda de un conjunto de datos predefinido en el ámbito biomédico con el suficiente tamaño como para que las puntuaciones del “embedding” sean lo más realistas posibles. Adicionalmente, existen archivos contenedores de modelos word2vec generales como el de Google News (disponible en: <https://github.com/mmhaltz/word2vec-GoogleNews-vectors>) que pueden utilizarse para realizar pruebas y comprobar su funcionalidad en ámbitos acotados.

También se plantea la investigación de otro paradigma de Word Embeddings como GloVe (Global Vector for Word Representation) el cual es más utilizado por los investigadores en PLN por sus resultados.

En vista a mejorar no solo la representación de la información utilizada para realizar el aprendizaje por refuerzo, también se plantea investigar otras formas de realizar dicho aprendizaje distintas al uso de Policy Gradients como Q Learning.

Finalmente, y en vistas de poner a prueba el futuro sistema, se plantea también la participación en la correspondiente competición BIOASQ poniendo en practica las investigaciones llevadas a cabo y las futuras con el fin de obtener el mejor resultado en un caso real.

Bibliografía y referencias

Moreno, Antonio 2017a. *Procesamiento del lenguaje natural ¿qué es?*, instituto de ingeniería del conocimiento. [online] Disponible en: <http://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>, Octubre 2017.

Moreno, Antonio 2017b. *Aplicaciones del Procesamiento del Lenguaje Natural*, instituto de ingeniería del conocimiento. [online] Disponible en: <http://www.iic.uam.es/inteligencia/aplicaciones-procesamiento-lenguaje-natural/>, Diciembre 2017.

Lloret, Elena y Palomar, Manuel 2012. *Text summarisation in progress: A literature review*. *Artificial Intelligence Review.*, 37(1):1–41.

Romero, J.J. 2007. *Inteligencia artificial y computación avanzada*, Fundación Alfredo Brañas, 2007. - 400 pág. - 22 cm. - Índice p. 5. - D.L.: C-2349-07. - ISBN.: 84-934497-0-9

Martín Mateos, F.J. y Ruiz Reina, J.L. 2013. *Procesamiento del lenguaje natural*, Dpto. Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla, materiales docentes, 2012-2013.

Cardoso, A. C. y Pérez, M. A. 2013. *Generación automática de resúmenes*, *Actas del 1er Congreso Nacional de Ingeniería Informática/ Sistemas de Información, CoNalISI 2013, Córdoba, 2013.*

ACERO, Ignacio 2001. *Generación automática de resúmenes personalizados*, *Procesamiento del Lenguaje Natural*. Nº 27 (sept. 2001), pp. 281-290.

Sempere, José M. 2013, *Aprendizaje por refuerzo*, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, materiales docentes, 2012-2013

Sancho, Fernando 2017a. *Aprendizaje por refuerzo: algoritmo Q Learning*. [online] Disponible en: <http://www.cs.us.es/~fsancho/?e=109> , Abril 2017

Palmer Pol, A. y Montaña Moreno, J.J. 1999, *¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones*, *Adicciones: Revista de sociodrogalcohol*, ISSN 0214-4840 , Vol. 11, Nº3, págs. 243/255, 1999

Sancho, Fernando 2017b. *Redes neuronales: una visión superficial*. [online] Disponible en: <http://www.cs.us.es/~fsancho/?e=72>, Abril 2017

Li, Yuxi 2017 *Deep Reinforcement Learning: An Overview*. arXiv:1701.07274, 2017.

D'Souza, Jocelyn 2018. *An Introduction to Bag-of-Words in NLP*. [online] Disponible en: <https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428>, Abril 2018

Naryani, Tushti 2017. *What are the differences between GloVe, word2vec and tf-idf?*, Quora. [online] Disponible en: <https://www.quora.com/What-are-the-differences-between-GloVe-word2vec-and-tf-idf>, Agosto 2017

Brownlee, Jason 2017a. *What Are Word Embeddings for Text?*, *Machine Learning Mastery*. [online] Disponible en: <https://machinelearningmastery.com/what-are-word-embeddings/>, Octubre 2017

Harris, Zellig (1954) *Distributional Structure*, *WORD*, 10:2-3, 146-162, DOI: 10.1080/00437956.1954.11659520

Sim, Yanchuan 2017. *What are the continuous bag of words and skip-gram architectures?* Quora. [online] Disponible en: <https://www.quora.com/What-are-the-continuous-bag-of-words-and-skip-gram-architectures>, Febrero 2017

BioASQ 2018. *A challenge on large-scale biomedical semantic indexing and question answering*, [online] Disponible en: <http://www.bioasq.org/>, 2018

Mollá, Diego 2017, *Towards the Use of Deep Reinforcement Learning with Global Policy For Query-based Extractive Summarisation*, Charla impartida, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Noviembre 2017

Lin, Chin-Yew 2004. *Rouge: A package for automatic evaluation of summaries*. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81

Brownlee, Jason 2017b. How to develop Word Embeddings in Python with Gensim, Machine Learning Mastery. [online] Disponible en: <https://machinelearningmastery.com/develop-word-embeddings-python-gensim/>, Octubre 2017