ANÁLISIS DE INFORMES MÉDICOS DE PACIENTES MEDIANTE TÉCNICAS DE NLP Y CONSULTA DE METATESAUROS

Beltrán Rodríguez-Mon Barrera

Autor

David Contreras Bárcena

Director

***Abstract*— La importancia del procesamiento de datos no estructurados dentro del mundo Big Data va en aumento, dado que existe un gran porcentaje de información recogida en formatos sin estructura, como textos, imágenes o audios. La propuesta de este proyecto es la elaboración de un sistema basado en técnicas de procesamiento de lenguaje natural (NLP) y la consulta de diccionarios y metatesauros biomédicos (UMLS) que se pueda utilizar de apoyo para los profesionales de la salud y del campo de la biomedicina en el tratamiento de la información masiva sobre recursos de carácter médico. El sistema desarrollado busca crear un flujo de procesamiento de información que extraiga las partes relevantes de informes clínicos de pacientes y presentar la información de forma que ayude en la toma de decisiones llevada a cabo en el contexto biomédico. En este proyecto se han elaborado tres módulos para la representación de la información y del conocimiento adquirido: clustering de documentos, un generador de resúmenes automáticos y un sistema de predicción en la creación automática de contenido en los informes basado en filtros colaborativos.**

Keywords — Big Data, NLP, NER, RegEx, MetaMap, TREC, UMLS, Automatic Text Summarization

I. Introducción

Actualmente se genera cada día una enorme cantidad de información proveniente de fuentes de datos muy diversas, trayendo consigo una multitud de formatos que hacen que la extracción de información y del correspondiente conocimiento sea distinta dependiendo de cada caso, yendo desde procesos simples y sencillos, como es el caso de una lectura de una tabla en un archivo *csv*, hasta procesos mucho más costosos como podría ser la interpretación de información más compleja tal como en el caso de un archivo de audio.

Existe actualmente una generación de datos no estructurados que sobrepasan en una gran proporción a aquella información sí estructurada, dado que éstos primeros tienen naturalezas mucho más amplias y diversas: desde textos como informes automáticos, correos electrónicos hasta elementos tales como las imágenes, los sonidos o las ondas. Esta inmensa cantidad de información, y la necesidad identificada por distintos sectores (empresas buscando una ventaja competitiva, u organismos públicos que desean contar con las mejores herramientas posibles) ha encaminado el desarrollo de muchas y diferentes formas de abordar este nuevo tipo de datos.

De esta forma, se ha desarrollado una clara nueva tendencia en cuanto al tratamiento de este tipo de información. Tanto para las empresas, como se explica en [1], como en otros campos una buena implementación de un sistema de tratamiento de datos no estructurados puede suponer un gran valor no alcanzable de otras formas.

Este proyecto tratará sobre la elaboración de una herramienta basada en técnicas de Procesamiento de Lenguaje Natural implementada sobre informes médicos de consultas de pacientes. Los textos de las consultas se filtrarán para eliminar las palabras sin información y se extraerán aquellos términos clave para el análisis posterior, pudiendo así generar fenotipos de los pacientes, de asociar las medicaciones con las dolencias de los pacientes y de ser capaces de poder identificar enfermedades a partir de los síntomas mostrados por el paciente.

II. Herramientas y Recursos

A. *Fuentes de Datos: TREC*

La fuente principal de datos de este proyecto son los 36.175 archivos de datos que contienen extractos de informes médicos obtenidos de *The Text Retrieval Conference (TREC)* en formato xml los cuales poseen la información de la dolencia principal del paciente (en una frase corta), el identificador de la consulta, códigos de diagnósticos admitidos y descartados, y el propio texto del informe.

La generación de estos archivos es, tal como se indica en la Figura 1, un compendio de las visitas:

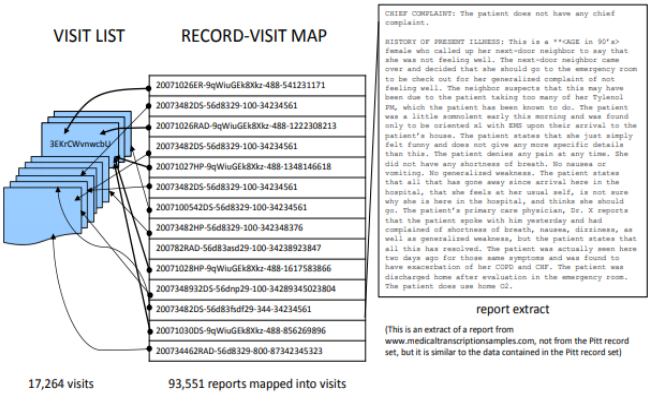

Figura 1. Recopilación de las visitas en extractos médicos [2]

Cada uno de los archivos de este proyecto se compone de los siguientes campos:

- ***Admit_diagnosis & Discharge_diagnosis***: Códigos ICD del informe; resumen en forma de códigos de las principales dolencias y problemas identificados a un paciente.

- ***Type & Subtype***: Tipo de informe mediante siglas, en concreto:
  - *Radiology Reports*
  - *History and Physicals*
  - *Consultation Reports*
  - *Emergency Department Reports*
  - *Progress Notes*
  - *Discharge Summaries*
  - *Operative Reports*
  - *Surgical Pathology Reports*
  - *Cardiology Reports*
- ***Chief_Complaint***: Texto libre y reducido que resume la dolencia principal del paciente.
- ***Report_text***: Recopilación en texto libre de las visitas de un paciente.

El campo de *report_text* supone la fuente principal de información para el proyecto y está compuesto por texto libre, suponiendo esto que su tamaño es variable y su estructuración (o forma en la que se ha generado) desconocida.

Esto supone que, para la elaboración de los modelos de *Machine Learning*, el pre-procesamiento mediante técnicas de NLP de este campo suponga una gran importancia. Al contener todos los análisis realizados por los expertos para cada uno de los pacientes un correcto tratamiento de esta información posibilitará la extracción de conocimiento para poder llevar a cabo todas las recomendaciones deseadas.

Para poder trabajar con estos datos se buscará estructurar lo máximo posible el texto de los informes siendo capaz de dividirlo en apartados tales como el historial previo de un paciente, su curso durante un ingreso o el diagnóstico realizado por el médico.

El procesamiento del texto supondrá una forma de estructurar los datos según su contenido para que los pasos posteriores se puedan llevar a cabo con la mayor precisión posible.

B. *UMLS*

Además, estos datos se complementarán con información de un metatesauro UMLS (*Unified Medical Language System*) [3] para así identificar de forma inequívoca los términos médicos utilizados en los informes.

C. *MetaMap*

Esta herramienta supone la integración del conocimiento del dominio de la medicina dentro del proyecto, y se ha implementado dentro del *pipeline* encapsulada en una API a la que, a la hora de procesar los documentos, se le llamará para que lleve a cabo un proceso de reconocimiento de entidades dentro del texto [4].

La principal utilidad de *MetaMap* es la de integración del metatesauro UMLS de una forma sencilla y eficaz dentro de la herramienta. Este programa encapsula dentro de sí todo el diccionario de términos con sus correspondientes códigos, es capaz de analizar automáticamente un texto y dividirlo en palabras o grupos asignándoles un identificador y explicando la entidad a la que hace referencia cada término (relacionando los términos de forma semántica).

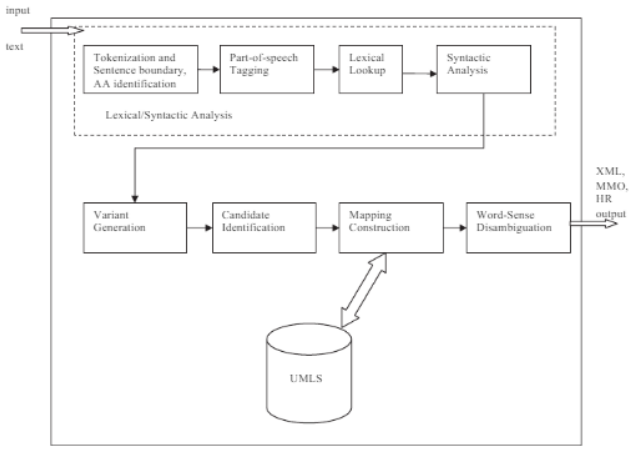

Figura 2. Arquitectura de *MetaMap*

III. Propuesta del Sistema

Durante todo este proceso se utilizarán diversas técnicas de NLP, ayudándose de librerías para el pre-procesado del texto (PoS Tagging) y modelos de *Machine Learning* utilizados para generar vectores representativos del texto a partir del contenido de este, además del uso de técnicas genéricas para el procesado de datos con el objetivo de poder realizar *clusters* de los distintos informes agrupándolos según la enfermedad del paciente o sus molestias previas antes de acudir a la consulta.

A. *Ciclo de vida del proyecto*

El desarrollo de este proyecto se ha dividido en tres bloques funcionales básicos (*Figura 3*).

Primero de todo, tomando como fuente de los datos los informes médicos, se realiza una limpieza de los datos mediante técnicas propias de *Text Mining* y *NLP*.

Tras esto, se lleva a cabo la integración de la aplicación MetaMap para integrar el conocimiento del dominio de la salud y la biomedicina dentro del proyecto.

Los últimos módulos desarrollados son los relativos al procesamiento del texto con el objetivo de servir como sistemas de apoyo a los profesionales del ámbito descrito a la hora del tratamiento del histórico de los datos.

La arquitectura a alto nivel del ciclo de vida del proyecto queda definida en la *Figura 3*.

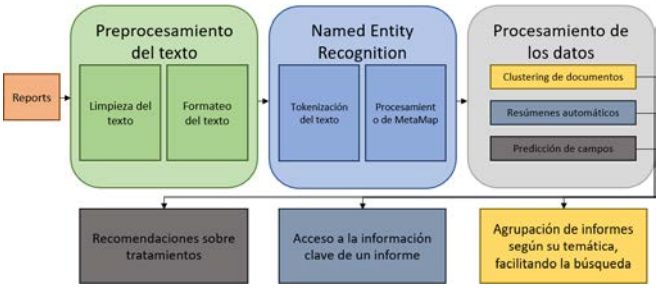

Figura 3. Esquema del sistema desarrollado

B. *Herramientas utilizadas*

Este proyecto se ha desarrollado en Python funcionando como una aplicación que encapsule toda la arquitectura mostrada en la *Figura 3*.

Esta aplicación estará alojada dentro de un *Cluster Big Data* para tener acceso a todo el histórico de datos necesario para su funcionamiento, y será accesible a través de una interfaz.

Además, la herramienta MetaMap, que necesita de un compilador de *JAVA*, también estará alojada dentro de este *Cluster* y es accesible a través de una API.

En los capítulos posteriores se describirán los módulos que desarrolladas en las distintas etapas del desarrollo.

IV. PrePROCESAMIENTO DEL TEXTO

El primer bloque del proyecto es el módulo de procesamiento de los datos de texto. El objetivo de éste es conseguir obtener una estructuración del texto lo más alta posible, de forma que los apartados siguientes puedan ser tratados facilitando lo máximo posible la entrega de una información estructurada en varios apartados.

El campo '*report_text*' utilizado, aunque no posea un formato claro, sí que tiene algunas características de las que nos podemos aprovechar a la hora de realizar el procesamiento del texto.

Primero de todo, todos los archivos de texto van encabezados por un identificador del informe, entre corchetes, que no posee ningún tipo de información valiosa para realizar un Procesamiento de Lenguaje Natural posterior.

Además, se aprecia que en el texto se va dividiendo el informe en bloques de información utilizando palabras clave en mayúsculas, lo cual será de provecho para estructurar cada uno de los informes en una estructura de apartados para el análisis posterior.

A. *Lectura de los informes médicos (XML)*

En primer lugar, se realiza la lectura automática de los archivos *xml* que contienen la información de los informes mediante la librería 'DOM'. Tras la lectura del archivo se debe realizar la captura del campo de texto para procesarlo a continuación.

A partir de aquí cada informe queda contenido dentro de una variable, que procesaremos más adelante para acabar con una estructura generada a partir de ésta.

B. *Limpieza del texto*

El objetivo de este módulo es eliminar las estructuras de texto no deseadas para que el formateo posterior del texto se realice utilizando texto útil y no caracteres indeseados.

El primer paso llevado a cabo es la regularización del texto utilizando un diccionario de contracciones del inglés para deshacerlas, de forma que los analizadores sintácticos y léxicos posteriores no tengan problemas a la hora de identificar estos términos.

Tras esto, se descartarán partes del texto sin utilidad, como son los metadatos generados en la creación de los informes que se incrustaron dentro del campo de texto libre, como son la fecha de emisión del informe completo o el responsable del mismo.

La funcionalidad principal empleada para el filtrado de metadatos incrustados dentro del informe ha sido la creación de un archivo de texto con palabras clave de forma que, si una línea comienza por alguna de estas palabras, ésta sea descartada directamente del texto.

Tras esto, para el filtrado de los elementos no deseados del se utiliza principalmente *RegEx*, apoyando también este procesamiento con el uso de sustituciones de literales (*.replace()* en *Python*).

Los casos concretos para los que se ha utilizado RegEx en este proyecto son los siguientes:

El primer paso es la desencapsulación de la información personal ofuscada para quedarnos únicamente con valores numéricos útiles.

Tras esto se eliminan los rastros de datos omitidos en la generación del informe.

Posteriormente queda capturada la información entre paréntesis para facilitar el procesamiento posterior de NLP.

En el caso de la información entre el resto de delimitadores se puede descartar totalmente, dado que no tienen utilidad.

Tras los procesos anteriores, se eliminan los caracteres sin utilidad dentro de un texto que representa el Lenguaje Natural en inglés.

El siguiente paso es eliminar caracteres utilizados para identificar listas (pero manteniendo el contenido de estas).

Por último, se eliminan los espacios y los saltos de línea múltiples creados al generar el informe.

Este proceso queda ejemplificado en los siguientes bloques, definidos por el patrón de colores mostrado.

Versión antes del filtrado:

```
DATE OF SERVICE:  **DATE[Feb 02 06]

HISTORY OF PRESENT ILLNESS:
The patient is a **AGE[in 20s]-year-old gentleman
who was  sticking a pin in his mouth when his
roommate accidently hit it...

IMPRESSION:
1. UNREMARKABLE CT ANGIOGRAM OF THE HEAD.
2. UNREMARKABLE CT ANGIOGRAM OF THE NECK.

EMERGENCY DEPARTMENT COURSE:  ... He is not having
any pain at this time and is resting comfortably.
I spoke with Dr. [_____]*** and
discussed these findings with him including the
...

(\n)

My signature below is attestation that I have
interpreted this/these examination(s) and agree
with the findings as noted above.
```

Versión después del filtrado:

```
HISTORY OF PRESENT ILLNESS:
The patient is a 20-year-old gentleman who was
sticking a pin in his mouth when his roommate
accidently hit it...

IMPRESSION:
UNREMARKABLE CT ANGIOGRAM OF THE HEAD.
UNREMARKABLE CT ANGIOGRAM OF THE NECK.

EMERGENCY DEPARTMENT COURSE:
... He is  not having any pain at this time and is
```

```
resting comfortably.  I spoke with  Dr. and
discussed these findings with him including the
...

My signature below is attestation that I have
interpreted this/these examination and agree with
the findings as noted above.
```

C. *Formateo del texto*

En este paso, mediante la identificación de frases concretas dentro del texto, se procede a la división de éste en sus distintas partes para poder realizar un análisis posterior más acorde al área de trabajo.

Para llevar a cabo esta parte lo que se ha realizado es crear unos archivos de configuración en formato *.txt* similares al utilizado anteriormente, en los que se almacenarán todas las cadenas de caracteres que representen títulos de un apartado en concreto. La ventaja de utilizar estos archivos es que, si en la ejecución de la herramienta creada un documento falla, podemos capturar fácilmente qué frase o frases representan un apartado, y añadirlos al final del fichero para ajustar su forma de funcionamiento.

En la ejecución de este componente cada cabecera identificada en el texto queda sustituida por diversas etiquetas encabezadas por el nombre del archivo en el que se encontró la cadena (sin la extensión *.txt*), delimitando de esta forma cada uno de los apartados del texto encontrados dentro de un informe y facilitando una división posterior del texto en los diversos apartados identificados.

Tras la correcta inserción de las etiquetas en el texto éste se divide y toda la información queda integrada dentro de una instancia de la clase *Report*, cuya estructura queda definida en la *Figura 4*, para que la información resulte más fácilmente manipulable en los apartados posteriores.

REPORT
ID del report
Alergias
Síntomas de entrada
Diagnóstico
Evolución en el hospital
Historial clínico
Análisis de laboratorio
Medicamentos
Operaciones realizadas
Signos del paciente
Recomendaciones de los médicos

Figura 1. Estructura de información estructurada de los informes

V. INTEGRACIÓN DE METAMAP

La segunda parte de este proyecto queda constituida por la integración de la aplicación *MetaMap* dentro del pre-procesamiento del texto con el objetivo principal de integrar todo el conocimiento de expertos aportado por el metatesauro UMLS.

Los resultados devueltos por *MetaMap* se utilizará para complementar la información estructurada en el *Apartado IV – PreProcesamiento del Texto*. Cada palabra o grupos de palabras del texto se acompaña de información sobre el identificador dentro del metatesauro, una explicación generada por éste, y las etiquetas según los grupos semánticos definidos dentro de la aplicación.

El objetivo principal de *MetaMap* es llevar a cabo el proceso de *Pos Tagging* y *Named-Entity Recognition* de los conceptos referenciados en el texto, prerrequisito primordial para ciertas aplicaciones relacionadas con la recuperación de información, *Text Mining*, categorización, resúmenes y otras tareas del procesamiento del lenguaje natural.

A. *Estrategias de mapeo de MetaMap*

Para cada expresión de entrada al sistema, la aplicación sigue los siguientes pasos:

1. Analizar el texto en sintagmas, llevando a cabo los pasos posteriores para cada una de las divisiones generadas.
2. Generar variaciones para cada frase nominal (consistentes en una o más frases con abreviaciones, acrónimos, sinónimos …).
3. Formar el conjunto de candidatos, conteniendo cada una de las variaciones.
4. Para cada candidato calcular su fuerza mediante funciones.
5. Seleccionar y combinar aquellos candidatos que mejor se ajusten al mapeo de la frase original, generando así ***Meta Mappings***

Los *Meta Mappings* son los elementos finales que definen las diferentes variaciones de conceptos asociadas a una cadena de entrada [5].

B. *Tokenización del texto*

Para la creación de las representaciones numéricas dentro de la herramienta se ha de llevar a cabo un proceso de reducción de los términos para una correcta interpretación de éstos.

El proceso de *tokenización* del texto separa cada elemento (oración del texto) en unidades de menor tamaño, que pueden ser palabras, caracteres o *substrings*. Este modelado de los datos ayuda a una representación fidedigna del texto adaptando el análisis posterior al caso de uso concreto en el que nos encontramos.

Para este proyecto se han realizado dos pruebas de *tokenización* distintas. Primero se optó por llevar a cabo un proceso de *lemmatization* (análisis sintáctico de las oraciones para llevar las palabras a su raíz) independiente a la aplicación de *MetaMap* utilizando la librería *Spacy* [6].

Este proceso extrae cada frase del informe y la analiza sintácticamente extrayendo la raíz real de cada una de las palabras (la que se puede encontrar en los diccionarios), y ofrece mejores resultados que los procesos de *stemming*, dado que estos últimos no aseguran que el resultado sea una palabra correctamente escrita (a costa de un mayor coste de computación).

Por otro lado, también se llevó a cabo un procesamiento del texto omitiendo la parte de *tokenización* para simplemente

integrar la salida del filtrado y formateo del texto hacia la aplicación de *MetaMap*.

Este último método ha demostrado tener mejores resultados que el primero dado que su eficiente integración dentro de la aplicación beneficia a la división del texto de entrada en los términos que mejor funcionen en la aplicación, independientemente de no eliminar con anterioridad a este paso los términos sin valor ni significado dentro del dominio de trabajo (como las *stopwords*).

C. *Procesamiento de MetaMap*

Tras la limpieza del texto por parte de la aplicación, se obtiene una salida en formato de líneas de texto sin terminación, esto es, un fichero en el que cada *Meta Mapping* de los distintos sintagmas queda definido en una línea del fichero, tal como se muestra al final de este apartado.

Por otro lado, todas las posibles salidas del programa son:

- ***ID*** – Identificador único del texto.
- ***MMI***
- ***Score*** – MetaMap Indexing (MMI) Cuanto mayor sea este valor, significa que el concepto UMLS es de mayor importancia según los algoritmos MMI.
- ***UMLS Concept Preferred Name*** – UMLS preferred name para cada uno de los conceptos UMLS del texto.
- ***UMLS Concept Unique Identifier (CUI)*** – CUI para los conceptos UMLS identificados.
- ***Semantic Type List*** – Lista de abreviaciones semánticas de los conceptos UMLS identificados.
- ***Trigger Information*** – Tupla de 6 elementos separados por comas que muestran los disparadores MMI para identificar el concepto UMLS.
  - ***UMLS Concept*** (Preferred or Synonym Text)
  - ***loc*** – Posición en el texto.
    - ***ti*** – Título
    - ***ab*** – Abstract
    - ***tx*** – Texto Libre
  - ***locPos*** – Número del enunciado dentro de la ubicación
  - ***text*** – Texto mapeado con el concepto UMLS.
  - ***Part of Speech*** – Determinado por el *MedPost Tagger* o el *Lexicon Lookup*
- ***Negation Flag*** – 1 si el texto se considera una negación para *MetaMap*.
- ***Positional Information*** – Información posicional separada por ';'.
- ***Treecode(s)*** – Lista separada por punto y coma de cualquier código de árbol *MeSH* asociado con el concepto UMLS.

En este proyecto, dado la limitación del conocimiento experto y los requerimientos necesarios para la parte de análisis posterior, solamente se han utilizado los siguientes campos, resultado de la salida por defecto de la aplicación:

- ***Concept Score***
- ***CUI***
- ***UMLS string matched***
- ***Concept's Preferred Name***
- ***Concept's Semantic Type(s)***

De esta salida cabe destacar la utilidad que resulta el campo del *Semantic Type*, utilizado para realizar el *Named Entity Recognition* (NER), utilizando así el conocimiento adquirido del dominio del caso.

```
Phrase: right sided subclavian line in place
again.

Meta Mapping (719):
612   C0205090:right sided (Right) [Spatial
Concept]
581   C0589488:Subclavian (Subclavicular
approach) [Spatial Concept]
748   C0205132:Line (Linear) [Spatial
Concept]
581   C0442504:Place [Spatial Concept]

Meta Mapping (719):
612   C0205090:right sided (Right) [Spatial
Concept]
581   C0589488:Subclavian (Subclavicular
approach) [Spatial Concept]
748   C0205132:Line (Linear) [Spatial
Concept]
581   C1533810:Place (Placement action)
[Health Care Activity]

...
```

Tras todo este procesamiento, aplicando el conocimiento adquirido gracias a *MetaMap* y el metatesauro UMLS, se procede con las fases de análisis de la información recabada para generar modelos de recomendación para los expertos dentro de este dominio de trabajo.

VI. Clustering de documentos

La primera de las funcionalidades integradas dentro de este proyecto es la elaboración de un sistema de *clustering* basado en el contenido almacenado en cada uno de los informes.

El objetivo de este módulo del proyecto es la creación de un sistema de agrupación de documentos por contenidos según los diferentes bloques explicados anteriormente en el *Apartado IV – PreProcesamiento del Texto.*

Este módulo apoyará a los profesionales en el análisis y búsqueda en el histórico de informes para poder descartar de forma automática aquellos que no tengan relevancia en el caso de uso, y posibilitando el trabajo posterior con únicamente un subgrupo del set de datos.

Para llevar a cabo este proceso, el primero de los pasos es la creación de las representaciones matemáticas dentro del sistema; este concepto se denomina en el ámbito del NLP como ***Word Embeddings***.

Los *Word Embeddings* son básicamente una forma de representación de palabras que une la comprensión humana del lenguaje con la de una máquina. Estas representaciones de texto se llevan a un espacio de n dimensiones donde las palabras que tienen el mismo significado tienen una representación vectorial próxima, lo que significa que dos palabras similares están representadas por vectores muy similares que son cercanos dentro de un espacio vectorial.

Esta característica es esencial para resolver la mayoría de los problemas de procesamiento del lenguaje natural relativos a este concepto.

Por lo tanto, cuando se utilizan *Word Embeddings*, todas las palabras individuales se representan como vectores de valores reales en un espacio predefinido. Cada palabra se asigna a un vector y los valores del vector se aprenden de una manera que se asemeja a una red neuronal.

Existen varias formas de generar esta representación vectorial de los textos: Mediante cálculos puramente estadísticos, mediante la utilización de modelos predictivos o basados en casos de uso concretos (ya sean redes neuronales, grafos…) …

Además, a la hora de configurar los métodos de *Word Embeddings* a utilizar, un parámetro clave es el de la **dimensión de los vectores.** En este proyecto todos los métodos utilizados utilizan una **dimensión de vectores de 200** debido a que los tamaños de los documentos a tratar son mayores al tamaño típico a la hora de realizar *Embeddings* de documentos**.**

En este proyecto se han utilizado dos métodos de creación de *Word Embeddings* distintos:

A. *TF-IDF*

TF-IDF es una medida estadística que evalúa la relevancia de una palabra para un documento comparándola con su aparición en toda una colección de documentos. Esto se hace multiplicando dos métricas: cuántas veces aparece una palabra en un documento y la frecuencia inversa del documento de la palabra en un conjunto de documentos.

$$w_{i,j} = tf_{i,j} \times \log\left({}^{N}/_{df_i}\right)$$

W = Peso de *i* en *j*

tf = Número de ocurrencias de *i* en *j*

df = Número de documentos que contienen *i*

N = Número total de documentos

Su forma de funcionamiento es sencilla, la aparición de una palabra dentro de un único documento hace que el peso suba, pero si esta misma palabra aparece en el resto de los documentos, baja. De esta forma, podemos computar qué elementos son característicos y diferenciadores para un documento en concreto, representando así un vector de características que recoja la información más importante de cada texto.

Además, se ha integrado un sistema de obtención del *topic* de cada *cluster* mediante el uso de técnicas de *LDA* (*Latent Dirichlet Allocation*) para poder definir de forma automática un tema orientativo de cada grupo, aunque la funcionalidad principal será la de buscar el *cluster* de un informe de interés y analizar a partir de ahí el resto de los informes.

B. *Doc2Vec*

Por otro lado, Doc2Vec es un modelo de predicción que se basa en aprender cómo proyectar un documento (matemáticamente) dentro de un espacio vectorial latente de n dimensiones. Existen dos métodos fundamentales de funcionamiento: el basado en *skip-grams* y el basado en *Bag of Words*; en este proyecto se ha utilizado el segundo método.

Doc2Vec está basado en el modelo **Word2Vec** [7], utilizado para representar en vectores palabras que aparecen en documentos. Lo que hace este modelo es aprender a predecir el valor de una palabra basándose en el contexto

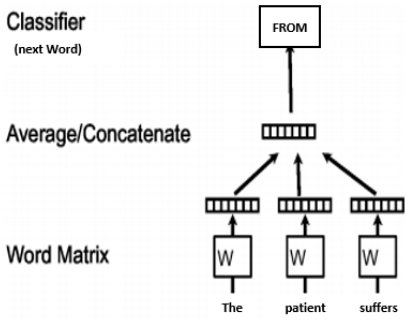

Figura 5. Ejemplo del procesamiento de Word2Vec

En Doc2Vec se añade una entrada más al modelo, el *Document ID*, y el objetivo del modelo será el de, además de predecir las representaciones de cada palabra, también predecir el valor de este identificador, que corresponderá a la representación del documento.

C. *Clustering con K-Means*

Tras obtener las representaciones vectoriales de los documentos, éstas son utilizadas para llevar a cabo una agrupación de los documentos utilizando *K-Means*.

De forma resumida, la forma de agrupación de *K-Means* es aquella en la que asegure que la distancia entre los elementos de un mismo *cluster* es mínima, mientras que aquellos elementos del resto de *clusters* estén lo más alejados que sea posible.

En este proyecto se ha integrado un sistema de cálculo automático del número óptimo de *clusters* implementando un cálculo del *Silhouette Coefficient* de los grupos, esto es, una medida de cuán similar es un objeto a su propio grupo (cohesión) en comparación con otros grupos (separación).

Las imágenes mostradas corresponden a una representación de los vectores en dos dimensiones sobre las dos componentes principales, por lo que se debe de considerar que la representación es orientativa de la distribución de los informes, pero faltan una gran cantidad de dimensiones para conseguir la representación perfecta de los grupos.

A continuación, se muestran las representaciones utilizando los dos métodos definidos anteriormente para clasificar los informes según los *Síntomas de entrada* de los pacientes**.**

En el caso de **TF-IDF**, obtenemos la representación mostrada en la Figura 6,

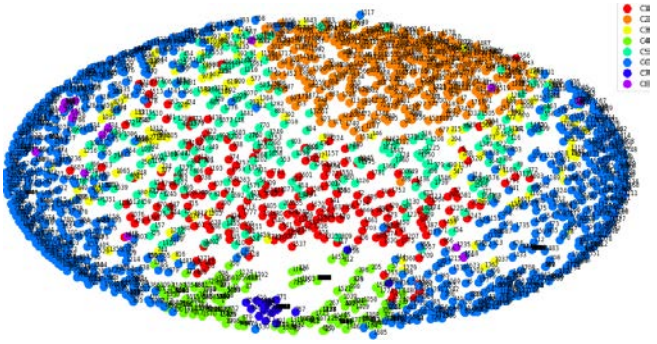

Figura 6. Representación del Clustering utilizando TF-IDF

Por otro lado, los resultados obtenidos utilizando el método de **Doc2Vec,** para la misma configuración de *K-Means*, son:

Figura 7. Representación del Clustering utilizando Doc2Vec

En este caso se aprecia que el codo se encuentra para únicamente 2 *clusters*.

D. *Comparativa de los resultados*

Por un lado, para la salida obtenida utilizando **TF-IDF** se obtiene, para el *cluster #6*, el siguiente resultado:

```
Topic:["pain" "Abdominal" "nausea" "vomiting"
"back")]
```

Siendo el contenido dentro de varios de los informes:

```
Report4.xml:
Abdominal pain. Abdominal pain and change in
mental status.

Report1111.xml:
Abdominal pain.

Report15340.xml:
Abdominal pain and vomiting.
```

Mientras que en el caso de Doc2Vec, observando el primero de los *clusters*:

```
Topic: "pain" "normal" "creatinine" "Chest"
"count"
```

Se ve que los conceptos resumidos en el *topic* son menos clarificadores. Por otro lado, analizando los documentos de este *cluster*.

```
Report1.xml:
Tooth pain.

Report108.xml:
I am coping with depression. BUN 25,
creatinine 1. 5. Chest x-ray from the 29th of
showed an enlarged heart. No pulmonaryedema.
I personally reviewed the examinations and
agree with thatinterpretation.

Report10282.xml:
Right hearing loss times three days.
```

Se aprecia que en este caso no existe ninguna relación entre los *Síntomas de entrada* de los pacientes, lo que invita a descartar el método de **Doc2Vec** para un uso como el de este proyecto.

VII. RESÚMENES AUTOMÁTICOS

Text Summarization recoge el conjunto de técnicas empleadas por una máquina para condensar un documento o un conjunto de documentos en breves párrafos o líneas utilizando métodos matemáticos.

El resultado de esta parte del proyecto busca servir de apoyo de forma que condense la información más relevante de los apartados que componen a un informe y que, de esta forma, sirva para ahorrar el tiempo de los profesionales a la hora de tener que buscar información dentro los históricos médicos.

Para llevar a cabo este proceso existen dos aproximaciones principales, ambas probadas durante el desarrollo de este proyecto, que quedan definidas a continuación:

A. *Extractive Text Summarization*

Este primer proceso se basa en la identificación de las partes más importantes mediante un sistema de *ranking* de las distintas divisiones generadas en el mismo (ya sean párrafos, líneas…). El resultado final de esta técnica es un resumen compuesto por las frases más importantes identificadas, de forma que el modelo no ha de ser capaz de crear de cero una salida en lenguaje natural.

Los distintos modelos probados han sido:

- ***TextRank*** [8]. Este algoritmo se basa en 2 pasos, que son:
  - Identificar oraciones relevantes: Construcción de un grafo donde los vértices representan cada oración en un documento y los cruces entre las oraciones se basan en la superposición de contenido.
  - Identificar palabras clave relevantes: Construcción de una red de palabras según términos consecutivos. Se establece un enlace entre dos palabras si se suceden, el enlace adquiere un peso mayor si estas 2 palabras aparecen con mayor frecuencia una al lado de la otra en el texto.
- ***LexRank*** [9]. Una oración que es similar a muchas otras oraciones del texto tiene una alta probabilidad de ser importante.
- ***LSA (Latent semantic analysis)***. Análisis de relaciones entre un conjunto de documentos y los términos que contienen mediante la producción de un conjunto de conceptos relacionados con los documentos y términos, asumiendo que las palabras con un significado cercano aparecerán en fragmentos de texto similares.
- ***Luhn*** [10]**.** Enfoque basado en TF-IDF. Es útil cuando las palabras muy poco frecuentes y las palabras muy frecuentes (*stopwords*…) no son significativas.
- ***KL.Sum*** [11]. Selecciona oraciones basándose en la similitud de distribución de palabras con el texto original. Su objetivo es reducir los criterios de divergencia KL. Utiliza un enfoque de optimización

greedy y sigue agregando oraciones hasta que disminuye la divergencia KL.

- ***BERTSUM*** [12]**.** Adaptación del modelo BERT para utilizarse en procesos de generación de resúmenes automáticos. El modelo BERT se utiliza dentro de este contexto como forma de identificar dentro de un documento el peso que tiene cada una de las frases. Este peso se computa en base al contexto del texto (tanto anterior como posterior, característica fundamental en BERT). La salida de este modelo son valoraciones para cada una de las frases del documento según el valor que aporten dentro del mismo, escogiendo las n frases con mayor valoración.

*B. **Abstractive Text Summarization***

Por otro lado, esta otra técnica se basa en la reproducción del contenido más importante de una nueva forma tras la interpretación y el examen del texto utilizando técnicas avanzadas de lenguaje natural para generar un nuevo texto más corto que transmite la información más crítica del original.

Este enfoque implica la creación de un resumen basado en *Deep Learning*, generando nuevas frases y términos diferentes a las del documento real, buscando representar la idea identificada tras analizar el documento y suponiendo una técnica mucho más compleja que el otro enfoque.

En esta técnica se utilizan modelos *seq-to-seq* de cadenas de texto que se entrenan para el caso de uso específico de la elaboración de resúmenes.

Los modelos probados correspondientes a esta técnica han sido:

- ***T5*** [13]. Transformer elaborado por *Google* basado en un modelo codificador-decodificador que enfoca todos los casos de uso a un caso *text-to-text* y entrenado utilizando un gran corpus de textos.
- ***BART*** [14]. Modelo *seq-2-seq* basado en un codificador bidireccional (como BERT) y un decodificador *left-to-right* (como GPT). Su entrenamiento se lleva a cabo ordenando de forma aleatoria todas las frases donde partes del texto se encapsulan tras *tokens* de enmascaramiento. Modelo eficiente en la generación de texto.
- ***GPT-2 (Generative Pre-trained Transformer 2)*** [15]. Modelo entrenado con el objetivo de predecir la siguiente palabra de una oración, dadas todas las palabras anteriores dentro de un texto. La diversidad del conjunto de datos hace que este simple objetivo contenga demostraciones naturales de muchas tareas en diversos dominios.
- ***XLM*** [16]. Modelo especializado en el aprendizaje de relaciones entre términos de distintos idiomas y en procesos de clasificación de términos multi-idioma.

*C. **Modelo definitivo***

Tras las pruebas realizadas sobre multitud de documentos se ha concluido que la técnica que mejores resultados proporciona es la de *Extractive Text Summarization*, debido principalmente a la no dependencia de un modelo entrenado como fuente principal de generación de los resúmenes. El área de trabajo tan concreta y especializada en la que se elabora este proyecto hace que, sin un entrenamiento específico de modelos de gran complejidad con una fuente de datos propia de este ámbito, estas técnicas no aporten valor al proyecto.

Por otro lado, dentro del método de *Extractive Text Summarization*, se ha decidido escoger en concreto el método de BERTSUM que se integra modelos de *Machine Learning* dentro de esta técnica, debido a que, aunque utilice un modelo por detrás, éste no tiene las limitaciones por las cuales se descartan las anteriores, dado que el modelo se utiliza para ponderar las frases y no para crearlas desde cero.

Un ejemplo de resultado es el siguiente:

Para el caso de tener un texto de entrada como el siguiente

```
 Additionally, the patient denies having any
fevers, chills, headache, neck pain, abdominal
pain, diarrhea, or dysuria. Remaining review of
systems is otherwise negative

 This is a well-appearing male sitting in bed in
no acute distress

 Pupils are equal and round bilaterally.
Extraocular movements are intact. Oropharynx is
without erythema or exudate. No lymphadenopathy or
JVD. Regular rate and rhythm. Clear to
auscultation bilaterally without wheezing, rales,
or rhonchi. Soft, nontender, and nondistended
without guarding or rebound. Without clubbing,
cyanosis, or edema. Alert and oriented x3. No
focal deficits are noted

 The patient remained stable in the emergency
department. He was given an inch of nitro paste
and 4 baby aspirin. He is not having any pain at
this time and is resting comfortably

 I spoke withand discussed these findings with him
including the CT scan of the chest. I also spoke
with Dr the attending radiologist who read the CT
scan. He states that this type of volvulus does
not require urgent surgical consultation
especially if the patient has no findings of
abdominal pain or vomiting. However, he noted that
the volvulus may need to be corrected in the
future if any problems arise

 Dr asked that we consult Dr and order a stress
test for the morning
```

Se aprecian 3 temas principales dentro del texto:

- Una descripción de las condiciones del paciente
- Los estudios realizados por el personal médico
- Comentarios realizados a los distintos doctores (cuyos nombres están anonimizados y por ende no aparecen)

El resultado generado por el modelo SUMBERT es:

```
Additionally, the patient denies having any
fevers, chills, headache, neck pain, abdominal
pain, diarrhea, or dysuria.

This is a well-appearing male sitting in bed in no
acute distress.
```

```
The patient remained stable in the emergency
department.

I spoke withand discussed these findings with him
including the CT scan of the chest.
```

Se aprecia que se tocan todos los temas mencionados anteriormente, utilizando las frases que el modelo ha identificado como más representativas de todo el texto.

Además, cabe destacar que el número de frases utilizadas se basa en un método de *clustering* utilizado por el modelo, donde hay tantas frases como *clusters*, y que además este número de grupos se calcula de forma automática calculando el codo obtenido mediante la aplicación del *Silhouette Coefficient.*

VIII. Predicción de campos

El último componente desarrollado en este proyecto ha sido un módulo de predicción de campos a partir de todo el procesado previo realizado, y con una fuerte influencia del conocimiento adquirido gracias al uso de *MetaMap*.

Principalmente, este módulo del proyecto se basa en la búsqueda de similitudes entre apartados de los documentos para poder deducir ciertos contenidos de otros que se desconocen.

Los resultados que se presentarán más adelante sobre este módulo se han realizado buscando predecir los *Medicamentos* que se le podrían recomendar a un médico como aptas para un paciente a partir de los *Síntomas de entrada* presentados por el mismo, pero esta parte es extrapolable a la predicción de otros campos como, por ejemplo, los *Análisis de laboratorio* que recomendar sobre las muestras de un paciente basándose en el *Historial clínico* del mismo.

Para este parte de predicción se ha decidido realizar un modelo basado en filtros colaborativos, concepto muy utilizado a la hora de creación de sistemas de recomendación y que son muy típicos en sets de datos como por ejemplo los de valoraciones de usuarios a contenidos.

Los filtros colaborativos basan la predicción del valor de un campo a partir del resto de los valores almacenados, buscando similitudes con otros *users* (informes, en nuestro caso) de la matriz. Si dos elementos poseen valores muy similares para ciertos casos, el sistema recomendará que un informe, en un valor X que no posee, adquiera el valor del otro informe similar.

En el caso de este proyecto, los datos que compondrán la matriz serán un *one-hot encoding* marcando las medicinas que el médico le ha recetado a los pacientes, siendo el vocabulario total utilizado todas las diferentes medicinas que aparecen en los informes.

A. *Cold Start*

El principal problema que se debe encarar a la hora de programar un sistema como este es el caso en el que un informe no tenga ninguna medicina, lo que supone que su matriz estará llena de valores *void* (ni positivo ni negativo), por lo que le será imposible al recomendador predecir la medicina que más le convenga.

La solución implementada en el proyecto para este caso es la creación de un sistema de inicialización de los campos basado en la búsqueda de similitudes entre documentos mediante la utilización de la *Distancia del Coseno* para que, de esta forma, se puedan utilizar los valores contenidos dentro del documento más parecido al de caso de uso para capturar algunos de sus valores que sean válidos para iniciar el recomendador.

Para la generación de los *Word Embeddings* utilizados en esta parte se ha implementado el módulo TF-IDF utilizado anteriormente, dado que para esta necesidad ha demostrado un mejor funcionamiento que un sistema de generación de vectores más complejo.

Además, la técnica desarrollada en esta parte del proyecto sirve como herramienta única de recomendación mediante el cruce de varios campos, debido a que permite al usuario conocer automáticamente qué informes poseen un contenido lo más similar posible al informe que se analiza por el profesional. De esta forma exponiendo un caso en concreto, el sistema podría recomendar unos *Análisis de Laboratorio* según los *Síntomas de entrada* de un paciente.

B. Filtros colaborativos con KNN

Para poder realizar el sistema de recomendación, primero de todo se ha de configurar la matriz mediante el *one-hot encoding* mencionado anteriormente. La matriz está configurada marcando con valores ‘1’ las medicinas que aparecen en un informe, y un ‘-1’ las medicinas que se conoce que no son válidas, manteniendo con valor ‘0’ aquellos valores desconocidos.

	Medicina1	**Medicina2**	…	**MedicinaN**
Report1	1	-1	…	-1
…	…	…	…	…
ReportM	0	1	…	0

Tabla 1. Estructura de la matriz del filtro colaborativo

De este modo, el sistema buscará predecir qué valores con ‘0’ en la matriz son las más ‘cercanas’ a tener un valor de ‘1’.

Este sistema de recomendación se basa en la utilización del método de clasificación supervisada de *K-NN (k-nearest neighbors)* para identificar qué informes son los más cercanos al que se busca completar, dado que el sistema no ha de identificar únicamente el primero que tenga a ‘1’ el valor o los valores del informe en cuestión, sino que debe de computar los mejores valores a partir de muchos documentos.

En este caso mostrado a continuación se buscará recomendar sobre qué medicinas podrían ser útiles para un paciente según sus **molestias:**

Sobre un informe concreto el sistema ha generado la siguiente salida:

```
The list of the Medicines listed in report1:
Fentanyl, Valium, Neurontin, Dilaudid

The list of the Recommended Medicines:
1: Ranitidine
2: Steroids
3: Atorvastatin
4: Sulbactam
```

IX. Conclusiones y Trabajo futuro

El trabajo con una fuente de datos no etiquetada y que pertenece a un área de estudio tan complejo como es la medicina supone la necesidad de crear herramientas que en ningún caso sirvan como tomadoras directas de las decisiones. Es por esto por lo que el enfoque adquirido para la elaboración de este proyecto haya sido la de crear sistemas de apoyo de los profesionales.

Además, a la hora de llevar a cabo la programación de todo el proyecto, siempre se ha mantenido una traza del flujo llevado a cabo por los informes, mediante el uso de las clases, para que se pueda comprobar siempre el texto original y que en ningún caso se omita información directamente al responsable de tomar las decisiones.

Las herramientas de clasificación de documentos proporcionan una gran utilidad limitando la información que ha de ser analizada, y en este módulo del proyecto el mayor peso recae sobre la técnica de *Word Embedding* elegida, dado que es la parte encargada de representar el texto de la forma más fidedigna posible.

Por otro lado, el generador de resúmenes automáticos supone la omisión de información dentro de un informe y, aunque el modelo pondere las frases más importantes como relevantes, dentro de un contexto médico el contenido suele estar limitado a temas relevantes.

Por último, aunque todo sea funcional y ofrezca los resultados esperados, la mayor potencia en el sistema de recomendación final desarrollado es la capacidad de predecir un campo del que no se disponga información a partir del resto, más que el uso de la matriz colaborativa para aconsejar sobre decisiones a partir del resto de decisiones tomadas.

Como trabajo futuro se podrían definir dos visiones complementarias.

A corto plazo se deberían de implementar otros sistemas de *Word Embedding* para complementar al de *TF-IDF* mediante el uso de modelos de *Machine Learning* de mayor complejidad que el desarrollado con Doc2Vec.

Además, en el sistema de recomendación sería aconsejable integrar más sistemas de comparativa de la similitud de los informes además del ya hecho mediante la *Distancia del Coseno*, potenciando de esta forma la recomendación sobre decisiones futuras a partir del histórico de datos del que se dispone.

Pensando a largo plazo también se podría analizar el uso de un mayor número de salidas de la aplicación de MetaMap, dado que el foco principal de este proyecto es la integración del conocimiento del metatesauro UMLS, y de esta forma se podría explotar todo el potencial de esta herramienta.

Además, conseguir datos etiquetados para poder entrenar de una forma más eficiente los modelos utilizados (*BERTSUM*, los utilizados en *Word Embedding…)* supondría una mejor implementación estos dentro de la aplicación.

En este proyecto se ha visto el potencial de integrar herramientas tan modernas como las de NLP en un ámbito como el de la medicina, pero también ha quedado claro la necesaria intervención de profesionales con conocimiento de ambas áreas (programáticas y de la salud) para que el tratamiento de los datos sea lo más cuidadoso posible.

X. Referencias

[1] M. V. I. K. Milan Kubina, Use of Big Data for Competitive Advantage of Company, 2015. M. Gracia (Nov, 2020). IoT – Internet Of Things. Available:

[2] E. M. Voorhees, «The TREC Medical Records Track,» de National Institute of Standards and Technology.

[3] «National Library of Medicine,» Available: https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html.

[4] National Institute of Health, «National Library of Medicine,» Available: https://www.nlm.nih.gov/research/umls/implementation_resources/metamap.html.

[5] I. A. Martínez, Análisis y optimización del recurso UMLS en la recuperación de la información biomédica mediante métricas de similitud semántica, Madrid, 2016.

[6] Spacy Available: https://spacy.io/.

[7] K. C. G. C. J. D. Tomas Mikolov, «Efficient Estimation of Word Representations in Vector Space,» 2013.

[8] P. T. Rada Mihalcea, TextRank: Bringing Order into Texts, Texas.

[9] D. R. R. Güneş Erkan, LexRank: Graph-based Lexical Centrality as Salience in Text Summarization, Michigan

[10] Luhn, H.P. (1958). The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 2(2), pp. 159-165

[11] Aria H. and Lucy V. (2009). Exploring content models for multi-document summarization. In Proceedings of Human Language Technologies, Annual Conference of the Noth American Chapter of the Association for Computational Linguistics

[12] Y. Liu, Fine-tune BERT for Extractive Summarization, Edinburh, 2019

[13] Google AI, «Google Blog,» 24 February 2020. Available: https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html.

[14] Y. L. N. G. M. G. A. M. O. L. V. S. L. Z. Mike Lewis, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, Facebook AI.

[15] J. W. R. C. D. L. D. A. Alec Radford, Language Models are Unsupervised Multitask Learners.

[16] A. C. Guillaume Lample, Cross-lingual Language Model Pretraining, Facebook AI