Trabajo Fin de Máster

Análisis de textos médicos mediante NLP

Autor:
Jesús Enrique Cartas Rascón

Profesor:
Rocío Romero Zaliz

Resumen

En el ámbito de la medicina se almacena una gran cantidad de información relevante: desde valores numéricos correspondientes a signos vitales hasta texto plano que realiza un especialista para completar un informe. Muchas veces los datos guardados en el historial médico de un paciente, que no tiene una estructura determinada, son ignorados. Este proyecto propone recuperar texto médico sin formato para extraer nuevo conociemiento que pueda utilizarse para complementar la información estructurada y mejorar en la clasificación y tratamiento de los pacientes.

Índice general

1. Introducción

En este capítulo introduciremos los principales problemas existentes en el contexto de minería de datos en el texto médico y pronpodremos una solución que se desarrollará a lo largo del documento.

1.1. Texto sin formato

Toda atención médica dispone de documentos que recogen toda la información relacionada con el paciente, enfermedad, y un seguimiento de ambos, así como los recursos a utilizar. En estos documentos suele haber una sección en la que el o la profesional en cuestión describe, en lo que podríamos denominar *texto sin formato* todas estos factores. Debido a la carencia de formato, es difícil trabajar con dichas secciones, por lo que se suelen ignorar.

La idea es centrar nuestra atención en esas secciones de texto, con objeto de obtener la mayor cantidad de información posible y anexarla, ahora con formato, al documento del que provienen, enriqueciendo el informe y habilitando nuevas claves de búsqueda, así como mejorando el indexado de los documentos.

1.2. Falta de datos

Uno de los principales problemas a los que nos enfrentamos es la falta de *datasets* o conjuntos de datos en los que estos documentos estén presentes.

Se buscarán y agregarán tantas fuentes de datos como sea posible, se unificarán y se creará una herramienta que aproveche todos los datos disponibles públicamente para generar datos nuevos.

1.3. Objetivos

Dado este marco, describiremos en esta sección los objetivos de nuestro trabajo.

En primer lugar, se agregarán todas las fuentes de información públicas que nos provean con datos de comentarios médicos listos para su minería y análisis.

Utilizando todos estos datos, se hará una evaluación de las herramientas que ya existen en el estado del arte. Haremos una revisión de cómo se utilizan y del rendimiento de dichas herramientas.

Sin embargo, para evaluar dichas herramientas, no utilizaremos los datos encontrados, sino que efectuaremos un flujo de trabajo alternativo. Utilizando técnicas de aprendizaje automático y generativo, crearemos un modelo que sea capaz de generar tantos comentarios médicos como sea necesario. La idea es suplir la carencia de datos con un modelo generativo, de forma que no se tenga que lidiar con aspectos de privacidad o licencia, ya que todos los comentarios serían generados de forma sintética.

Si bien los comentarios son sintéticos, deben ser lo suficientemente convincentes como para que la evaluación de las herramientas sea fiel y rigurosa. Esto ofrece una herramienta para los desarrolladores de las herramientas que habilita a un mejor y más fructífero desarrollo, ya que se disponde de una cantidad, *idealmente infinita* de comentarios.

2. Fundamentos de la minería de texto

En este capítulo discutiremos algunas de las técnicas y términos más importantes a la hora de hablar de minería de texto, así como minería de datos en general, con objeto de que todas las consideraciones realizadas posteriormente queden claras.

En *Text Mining Applied to Electronic Medical Records: A Literature Review* [1] se hace una revisión de los diferentes aspectos a tener en cuenta durante el procesamiento de textos médicos. Nos apoyaremos en gran medida en la estructura, contenidos y referencias de este artículo, que resume muy bien todo lo que necesitamos saber para resolver nuestro problema.

2.1. Minería de datos

La minería de datos es una rama de la informática que se dedica a encontrar tendencias y patrones en grandes volúmenes de información. Estas tendencias y patrones crean *conocimiento* a partir de los datos, es decir: información estructurada desde los datos no estructurados. Esta información es muy valiosa y contribuye en las decisiones que se vayan a tomar o a monitorizar algunos aspectos que sean de vital importancia para el interesado.

La minería de datos puede dividirse en un número de técnicas que funcionan de forma diferente en función del tipo de datos que tengamos y la información que busquemos.

1. **Asociación**: esta técnica se centra en encontrar relaciones entre las distintas variables de nuestros datos, con objeto de encontrar muestras que sean estadísticamente dependientes. Una de las técnicas más utilizadas son las reglas de asociación, cuya salida tras el cálculo son un conjunto de reglas con antecedentes y consecuentes, muy fácilmente interpretables por cualquier persona, familiarizada o no con la ciencia de datos. [2]

2. **Clasificación**: el proceso de clasificación trata de asignar una categoría a un conjunto de elementos que tengan algún aspecto en común. La clasificación en la minería

de datos es una de las técnicas más utilizadas, ya que la naturaleza de gran parte de los datos responden bien a este método. [3]

3. **Agrupamiento**: también denominado *clustering* trata de agrupar muestras que tengan características similares. A diferencia de la clasificación, aquí no tenemos una etiqueta o categoría a la que asignar las muestras, sino que las agrupamos *a ciegas*, simplemente basándonos en alguna métrica para evaluar la distancia que haya entre un determinado par de muestras. [4]

4. **Predicción**: la predicción nos ayuda a encontrar tendencias entre variables, generalmente en datos con una componente temporal fuerte. [5] Es común poder predecir si un paciente sufrirá una determinada enfermedad conociendo su historial médico, por ejemplo.

5. **Identificación de patrones secuenciales**: Al igual que la predicción, se trabaja sobre datos con una componente temporal marcada. En este caso, se buscan patrones, es decir, conjuntos o cadenas de muestras que aparecen de forma frecuente en un orden concreto.

2.2. Minería de texto

En esta sección, discutiremos los diferentes aspectos a tener en cuenta en la minería de textos en concreto, tras haber abordado el concepto de minería de datos en un ámbito más general.

2.2.1. Términos

Definiremos algunos de los términos más utilizados en esta disciplina, guiándonos principalmente por el trabajo de Kamran Kowsari, *Text Classification Algorithms: A Survey* [6].

Tokens

El término más esencial en minería de textos es *token*. Un token es la mínima unidad en la que dividiremos un cuerpo de texto a la hora de analizarlo. Este elemento suele corresponderse con una palabra, que en el contexto de la mayoría de los idiomas corresponde con un conjunto de letras separado por espacios anterior y posteriormente. Esto da lugar a la creación de *Tokenizers*, algoritmos que toman un cuerpo de texto como una

cadena de caracteres muy larga, y devuelven un vector de palabras. Estos *tokenizers* no han de tomar el espacio en blanco necesariamente ni exclusivamente como criterio divisor, aunque suele ser lo más común. Algunos de los *tokenizers* más famosos son:

- **Tokenizers de palabras**

  - **Standard Tokenizer**: El Standard Tokenizer divide el texto en términos siguiendo los límites de las palabras según están definidos en el algoritmo *Unicode Text Segmentation*. Funciona bien en general.
  - **Letter Tokenizer**: divide el texto en términos cada vez que encuentra un carácter que no es una letra.
  - **Whitespace Tokenizer**: Toma como criterio divisor el espacio en blanco.
  - **Language Tokenizer**: Otros tipos de tokenizers adaptados a diferentes idiomas, como el inglés, que es el idioma más estudiado con diferencia, pero también otros idiomas con caracteres y reglas diferentes a aquellos basados en reglas occidentales, como el tailandés, o el chino.

- **Tokenizers de palabras parciales**

  - **N-Gram Tokenizer**: Este tokenizador incluye un parámetro adicional. Primero divide el texto con alguna de las reglas mencionadas anteriormente, y posteriormente, divide cada término del vector resultante en una ventana deslizante de n elementos, de ahí *N-Gram.* Por ejemplo: *quick fox* devolvería [qu, ui, ic, ck], [fo, ox], dado un $n = 2$. Estos tokenizers también pueden utilizarse a nivel de párrafo, por lo que se devolverían pares de palabras, algo que puede ser muy útil para el análisis de *dichos* o expresiones.

- **Tokenizers de texto estructurado**

  - **Pattern Tokenizer**: este tokenizer utiliza el patrón provisto como parámetro para la división de texto, utilizando expresiones regulares.
  - **Simple Pattern Tokenizer**: este tokenizer utiliza el patrón provisto como parámetro para la división de texto, utilizando expresiones optimizadas para el patrón dado, lo que hace que funcione generalmente más rápido pero también será más específico.

Palabras vacías

Corrección ortográfica

Stemming y Lematización

Bolsas de palabras

Frecuencias: TF, IDF

Bibliografía

[1] Luis Pereira, Rui Rijo, Catarina Silva, and Ricardo Martinho. Text mining applied to electronic medical records: A literature review. *International Journal of E-Health and Medical Communications (IJEHMC)*, 6:1–18, 07 2015.

[2] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, 1991.

[3] R. Kumar and Rajesh Verma. Classification algorithms for data mining: A survey. *International Journal of Innovations in Engineering and Technology (IJIET)*, 1:7–14, 2012.

[4] Anil K. Jain, M. N. Murty, and P. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, 1999.

[5] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques, third edition. 2012.

[6] Kamran Kowsari, K. Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and D. Brown. Text classification algorithms: A survey. *Inf.*, 10:150, 2019.