UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Trabajo de Fin de Máster en
Ingeniería y Ciencia de Datos

Detección Temprana de Riesgos de Salud a Partir de Minería de Textos en Redes Sociales

Autor:

Samuel Moñux Salvador

Dirigido por:

Lourdes Araujo Serna
Raquel Martínez Unanue

Curso: 2021-2022: Convocatoria Ordinaria

Agradecimientos

Quisiera dedicar este trabajo a mi familia: a mi padres, Ángel e Isabel, a mis hijos, Marcos y Paula, y a mi mujer, Vanesa. Este Máster ha requerido más tiempo y esfuerzo del que había imaginado y sin su ayuda y apoyo incondicionales jamás hubiese podido acabarlo.

También quiero que expresar mi agradecimiento a las Directoras del TFM, Lourdes y Raquel. Sus respuestas han sido siempre inmediatas, exhaustivas y en todo momento se han mostrado pacientes con mis errores.

Tampoco puedo olvidar a mi amigo Antonio Juano, sin cuya inspiración y consejo nunca me hubiese decidido a afrontar este reto.

A todos ellos, gracias.

Zaragoza, Junio de 2022

Resumen

La anorexia y los desordenes de la alimentación relacionados son un problema de salud pública con unos altos costes en términos de sufrimiento y de gasto sanitario. Esta condición es especialmente prevalente entre las mujeres jóvenes y adolescentes de países desarrollados. Este sector demográfico tiene una fuerte presencia en redes sociales, lo que unido al hecho de ser diagnosticable por rasgos conductuales, hace que la tarea de su detección sea una buena candidata para la aplicación de herramientas de rastreo en redes sociales que estén basadas en técnicas de Aprendizaje Automático.

Esta es la base bajo la que se desarrollaron las ediciones de 2018 y 2019 del laboratorio *CLEF eRisk,* en el que una serie de equipos de distintas organizaciones de diferentes nacionalidades compitieron en el desarrollo del algoritmo más preciso y rápido (en términos de su capacidad de detección) sobre un conjunto de datos proporcionado por los organizadores.

Los objetivos de este proyecto están relacionados con estas competiciones y con la tarea sobre la que se desarrollaron y comprenden:

- La selección de los enfoques más prometedores y la elaboración de un estudio del "estado del arte" en aquel momento.
- El diseño e implementación de un sistema similar a los reseñados, inspirado en las técnicas y metodologías empleadas por los mejores equipos pero que incorpore también características novedosas.
- La evaluación de la solución desarrollada, en comparación con los algoritmos mejor puntuados pero también en términos de su "interpretabilidad". Este objetivo, sin ser uno de los enunciados explícitamente en la tarea del laboratorio, resulta crucial para cualquier herramienta que aspire a asistir en la toma de decisiones a los profesionales sanitarios de cualquier campo. Estos profesionales necesitan entender la lógica sobre la que se sustenta la predicción para confiar en ella, por lo que debe ser tenida en cuenta desde el principio.

El sistema desarrollado demuestra ser competitivo con las mejores soluciones, situándose "virtualmente" entre la décima posición (de 51) en la métrica considerada más útil para evaluar el rendimiento del sistema ($F1$ ponderada por la rapidez de detección$)$. Además, permite justificar las decisiones identificando los comentarios de los usuarios que más peso han tenido en la predicción, así cómo "describir" la temática de esos comentarios mediante palabras clave (que pueden estar o no en el texto pero que semánticamente están relacionadas con el contenido del mismo).

Abstract

Anorexia and its related eating disorders is a public health problem which has high costs in terms of human suffering and healthcare spending. It is also specially prevalent among young females of developed nations. The high level of engagement of this demographic in social media and the fact that it can be diagnosed by behavioral traits makes it a good candidate for its early detection by Machine Learning-powered scanning tools.

This is the basis assumption under which the *CLEF eRisk* lab was set up in its 2018 and 2019 editions. For the time it lasted an array of teams from different organizations and countries competed to develop the most accurate and quickest (in terms speed of detection) algorithm, which was evaluated against a curated *dataset*.

This project goals are related to the lab and the underlying task. These are:

- Select the most promising approaches and conduct a study about the "state of the art" at that moment.
- Design and develop a Machine Learning-based system over the same *dataset*, inspired in the techniques and methodologies of the best performers but which also incorporates some novel ideas.
- Asses the results of the proposed system, in comparison with the highest scored algorithms but also in terms of its interpretability. This self-imposed goal, which wasn't explicitly stated in the lab, is crucial for any tool which aims to assist healthcare professionals of any field. Those professionals will need to understand the rationale of the prediction in order to trust it, so it needs to be engineered upfront.

The performance achieved by the proposal is shown to be competitive with the best solutions, reaching a "virtual" tenth place in the competition (of 51) if ranked by the most useful metric to asses the performance of the system (F1 weighted by speed). Furthermore, it is able to identify the most important comments and describe their topic by using keywords (which may be absent from the text but are semantically related to it).

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción general y objetivos

1.1. Motivación

Los trastornos de la alimentación son afecciones graves de salud mental que pueden producir importantes problemas en las personas que los padecen e incluso la muerte en los casos más extremos. No existen datos fiables a nivel mundial que determinen de la prevalencia general con la que se producen debido a dos problemas fundamentales:

- La complejidad para identificar y registrar a los pacientes que las sufren.
- La variabilidad intercultural entre las distintas sociedades humanas en lo que respecta a la alimentación y lo que es considerado normal y anormal.

En el caso de España se estima que la prevalencia se encuentra en el rango de $4,1-6,4\,\%$ en mujeres de 12 a 21 años. En los hombres ronda el $0,3\,\%$ (Sociedad Española Médicos Generales y de Familia (2018)).

El *Manual Diagnóstico y Estadístico de Trastornos Mentales* o *DSM* (American Psychiatric Association (2013)) en su versión 5[1] dedica uno de sus capítulos a definir y establecer criterios de diagnóstico para cada uno de estos trastornos. Excluyendo a la pica (ingestión de sustancias no alimenticias) y a la rumiación (masticación de alimentos regurgitados), el resto de trastornos descritos tienen características comunes y de acuerdo a este mismo manual es frecuente que se produzcan diagnósticos cruzados. Los trastornos son:

- La anorexia nerviosa.
- La bulimia nerviosa.
- El trastorno por atracones *(binge-eating).*

[1]Este texto es referenciado comúnmente por las letras DSM seguidas por la versión, en este caso DSM-5.

La anorexia es probablemente la enfermedad más conocida de todas, y se caracteriza por una pérdida de peso a niveles considerados no saludables (índice de masa corporal inferior a $18,5kg/m^2$), acompañados por un intenso temor a engordar o a ganar peso. La percepción del propio cuerpo está alterada y los pacientes que la sufren o no son conscientes de su estado de delgadez o no consideran que sea un riesgo para su salud.

Dentro de la anorexia existen 2 subtipos:

- De tipo restrictivo, en la que la pérdida de peso se produce por restricciones en la ingesta de alimentos y/o intenso ejercicio físico.
- De tipo "atracón y purga" en la que se producen episodios de ingesta compulsiva seguidos por "purgas" realizadas consistentes en la autoinducción al vómito, o en el uso no terapéutico de laxantes, diuréticos, enemas o medicamentos.

La bulimia es similar a este segundo tipo, aunque los individuos que la padecen se mantienen en un rango de índice de masa corporal considerado como normal.

En cuanto al trastorno por atracones, de nuevo comparte características con los anteriores. En este caso, se producen también episodios de ingesta compulsiva, si bien no se incurre (al menos de forma tan extrema) en comportamientos compensatorios. No es infrecuente observar casos de obesidad en estos individuos.

Los problemas de salud derivados de estos comportamientos resultan en hospitalizaciones recurrentes, secuelas que afectan durante toda la vida y en los casos más extremos, en el fallecimiento de los pacientes. Este fallecimiento no siempre se produce por la desnutrición y sus problemas derivados. En un número elevado de casos es consecuencia del suicidio del paciente.

Dada la gravedad del problema, resulta muy útil el desarrollo de programas que puedan mejorar el diagnóstico y detección temprana de estas patologías, de forma que puedan ser tratadas lo antes posible.

Ésta es la tesis bajo la que se desarrolla la conferencia anual CLEF[2] eRisk, que explora la factibilidad de la detección temprana de diversos riesgos (incluidos problemas de salud) por 3 vías: la propuesta de metodologías de detección, la creación de fórmulas para medir la efectividad de las intervenciones y aplicaciones de tipo práctico.

En las ediciones de 2018 y 2019 versaron sobre la posibilidad de aplicar este tipo de técnicas al diagnóstico temprano de la anorexia (así como de otras dolencias del área de la salud mental como la depresión). El modelo para hacerlo fue el de un laboratorio de tipo competitivo, en el que varios grupos de distintos países intentaban identificar a los individuos que habían sido diagnosticados con anorexia.

El presente trabajo fin de máster pretende ser una continuación del trabajo expuesto en aquellas ediciones cuyos objetivos incluyen:

[2]Conference and Labs of the Evaluation Forum (CLEF) es un organismo cuya misión principal es la promoción de la I+D+i de sistemas de información basados en tecnologías multimodales y multilenguaje. http://www.clef-initiative.eu/

- Analizar los resultados, planteamientos y conclusiones obtenidos, obteniendo así una visión del "estado del arte en el problema".
- Desarrollar, a partir del **dataset de referencia,** una prueba de concepto para la resolución del problema que ponga en perspectiva la dificultad del problema así como la facilidad de reproducción de las técnicas aplicadas durante la competición.
- La evaluación de la solución desarrollada, en comparación con los algoritmos mejor puntuados pero también en términos de su "interpretabilidad".

Para ello se utilizarán las competencias adquiridas dentro del Máster en Ingeniería y Ciencia de Datos, y en especial, las que atañen a la Minería de Textos y el Aprendizaje Automático.

1.2. Diagnóstico conductual de la anorexia

En el apartado anterior se han presentado algunos de los rasgos básicos de la anorexia, que para afrontar la tarea desde un vista puramente tecnológico podrían ser suficientes. Al fin y al cabo, las técnicas a aplicar se fundamentan en la obtención automática de patrones y no en el conocimiento de dominio, que era el acercamiento preferido para la resolución de este tipo de problemas en los Sistemas Expertos de los 90 y que se actualmente se consideran superados.

No obstante, uno de los problemas del Aprendizaje Automático en general y del Procesamiento del Lenguaje Natural en particular, es lo que se conoce como el efecto *Clever Hans* (Heinzerling (2019)). Este fenómeno está inspirado en el caso del caballo del mismo nombre que era supuestamente capaz de resolver problemas aritméticos. En realidad el animal lo que detectaba era cambios de actitud de su cuidador según la respuesta mostrada fuese la correcta o no.

Trasladado al caso que nos ocupa, dado el sesgo de género del problema (9 de cada diez pacientes son mujeres), sería plausible que un simple[3] "detector de feminidad" pudiese llegar a obtener resultados respetables, al menos en lo que respecta a la exactitud, en conjuntos de datos que estuviesen sesgados (extraídos de una red social mayoritariamente frecuentada por hombres, por ejemplo).

Para evitar este problema, puede resultar útil el tratar de responder a la pregunta ¿cómo un ser humano intentaría reconocer a una persona que sufre anorexia a través de lo que escribe en una red social?

Sin la colaboración de un experto en Salud Mental o Psiquiatría, la opción con más posibilidades de éxito es la de acudir a la literatura científica para intentar identificar:

- Los temas e intereses que pueden ser habituales en las personas que sufren estos trastornos (por ejemplo, conversaciones sobre laxantes, etc...)[4]

[3] La detección del género en redes sociales es también un problema complejo (ver por ejemplo Vashisth and Meehan (2020)) si bien no es, por sí sólo, suficiente para resolver el problema planteado por razones evidentes.

[4] El dataset objeto de estudio contiene estrictamente casos de anorexia exclusivamente, si bien en un contexto real

- Rasgos de personalidad o emociones que puedan ser identificables en un texto escrito por un lector humano (como narcisismo, tristeza, etc...)

Para ello podemos acudir de nuevo al *DSM-5*, en el capítulo que trata estos problemas. De su lectura podemos extraer los siguientes datos de utilidad que ayudan a caracterizar a estas personas:

- El ya referido sesgo entre la prevalencia en hombres y mujeres.
- Temor a engordar y obsesión por la figura y en especial por la acumulación de grasa en determinadas zonas (caderas, muslos, etc...). Este temor se revela en comportamientos tales como:
  - Comprobación frecuente del aspecto corporal en el espejo o fotografías.
  - Comprobación del peso corporal hasta varias veces al día.
- Uso de laxantes, enemas, etc...
- Ejercicio físico realizado de forma excesiva, no saludable.
- Irritabilidad y/o comportamiento depresivo.
- Desinterés por las relaciones sociales y por el sexo.

En cuanto a los rasgos de personalidad que caracterizan a estas personas, en Duffy et al. (2019) se hace referencia a varios de ellos:

- Un nivel elevado de *Evitación del Daño*, lo que se caracteriza por una tendencia a inhibir comportamientos para evitar un castigo. En este caso se puede interpretar como la inhibición de tomar alimentos ante el castigo que supondría el "sentirse gordo".
- Elevado *Perfeccionismo* entendido como el mantenimiento de altos estándares personales así cómo de la preocupación por no cometer errores.
- Una baja *Autodirección*, que es el grado con el que una persona se ve a sí misma como integrada e independiente. Como hipótesis, ésto haría a estas personas más susceptibles a las presiones sociales, como la que sería consecuencia de una "cultura de la delgadez".

Podemos acudir también a estudios realizados en España como Díaz-Marsá et al. (2000), que también se apoya en el *Modelo de Cloninger*[5] y que arroja conclusiones parecidas:

parece difícil conseguir un diagnóstico tan preciso cuando hasta los profesionales en ocasiones tienen dificultades en hacerlo. A este nivel de análisis, se va a tratar de los rasgos presentes en individuos con cualquiera de estos trastornos.

[5]El Modelo de Cloninger propone 7 factores fundamentales que actuarían como componentes de la personalidad de un individuo, entre los que se encuentran los referidos Pefeccionismo, Evitación del Daño, etc...

> CI harm avoidance was significantly higher...and self-directedness was significantly lower ...in patients, compared with controls....To summarize, eating disorders were characterized by high neuroticism and low self-directedness. Furthermore, anorexia nervosa patients appear to be characterized by a high persistence, while bulimic patients (BN and bpAN)[6] were characterized by high harm avoidance.

Extraemos de esta lectura como otro rasgo psicológico relevante a tener en cuenta el *Neuroticismo*, que se manifiesta en sensaciones negativas permanentes como la tristeza o la ansiedad.

1.2.1. Prueba de concepto

A partir de las intuiciones adquiridas en el apartado anterior, resulta de interés comprobar hasta qué punto son relevantes y cómo de factible es para una persona sin formación en Salud Mental identificar a estas personas, disponiendo únicamente la mínima instrucción que proporcionan la lectura de las fuentes anteriormente referenciadas (el DSM-5 y los artículos).

La manera ideal para sacar conclusiones con cierta capacidad de generalización resultaría de aplicar un experimento con un alto número de participantes, obtenidos incluso de alguna plataforma de *crowdsourcing* como *MTurk*[7]. En este caso, no se dispone de los recursos para realizarlo y se aparta del objetivo principal del TFM.

Como alternativa se realizará como parte de este TFM una tarea similar a la que se ofrecería a los voluntarios. Las conclusiones servirán únicamente como intuiciones, sin el valor de las que podrían obtenerse con un experimento con varios participantes.

El ejercicio consiste en la obtención de 10 sujetos del *dataset* de forma aleatoria, de forma que:

- 3 individuos padezcan anorexia
- 3 individuos no padezcan anorexia
- 4 obtenidos aleatoriamente con 50 % de probabilidad (que es distinta a la elección aleatoria de 4 observaciones, dado que hay una ratio entre ambas clases de 9 a 1).

De esta manera el número de casos de cada tipo no es conocido de antemano y no condiciona las últimas predicciones, pero se asegura que aparecen ejemplos suficientes (al menos 3) de cada caso que identificar. Los textos se mostrarán de forma secuencial y en cualquier momento el participante podrá elegir entre calificar al sujeto como:

- Con anorexia (confirmar).
- Sin anorexia (descartar).

[6] BN=Bulimia Nervosa, bpAN=binge-purging Anorexia Nervosa, rAN=restricting Anorexia Nervosa

[7] https://www.mturk.com/ Es una plataforma en la que se pueden someter microtareas para que sean resueltas por seres humanos (porque no son todavía resolubles por ML) a cambio de una contraprestación económica. Uno de los casos de uso más comunes es el de etiquetado de *datasets* para su posterior automatización.

- Ver siguiente texto (no decidido).

En caso de llegarse al último elemento será obligatorio el emitir un resultado.

Este "experimento" se ha ejecutado como prueba de concepto de la manera descrita. El resultado para cada uno de ellos junto con la justificación de la etiqueta emitida para cada uno de los usuarios se describe a continuación. Los usuarios se identifican con un *hash* generado por el *script* de aleatorización.

Sujeto 1 (2c4246e1fd)

Gran parte de los primeros textos son *posts*[8] que están vacíos de contenido preguntando cosas simples como:

> Guys, I colorized this mugshot of Elvis. What do you think Reddit?

Por el estilo parece que es un usuario masculino y joven. Habla en ocasiones de que tiene novia. A partir del texto 130 se aprecia que el nivel de contenido ha subido y los comentarios son más largos.

En 170 de 1303 se emite un resultado: **NEGATIVO**.

Sujeto 2 (4f01e9da6c)

Persona joven (habla de su *SAT* reciente que es un examen estandarizado americano previo a la admisión en una Universidad). Se puede identificar cierto perfeccionismo y frustración con sus notas.

En 19 menciona su género (varón). En 22 que se muerde las uñas (comportamiento que se puede considerar como compulsivo). En 49 parece que aparte de hablar sobre esos comportamientos, su interés principal son los videojuegos así como preguntas de Historia.

En 56 dice que los métodos para disimular las uñas mordidas son para mujeres y aplicárselos le harían parecer afeminado.

En 85 parece que está algo depresivo.

En 95 de 1263 se emite el resultado **NEGATIVO**, dado que no parece tener problemas más allá de los propios de un adolescente.

Sujeto 3 (5e78d10dd0)

En 4 está hablando de comida y hace referencia al *subreddit*[9] denominado *bodyacceptance*.

En 7 habla de un incidente con una mujer con sobrepeso a la que se refiere de forma despectiva. En el comentario 25 reconoce tener problemas médicos. En 27 habla de su miedo a comer.

En 27 de 44 se emite el resultado **POSITIVO.**

[8] La red social Reddit de la que se ha extraído el dataset tiene dos tipos de contribuciones que pueden enviar los usuarios. Los *posts* o historias que se envían para iniciar discusiones (y que muchas veces son simples imágenes o memes sin contenido textual) y los comentarios que se hacen en respuesta al "tema" propuesto por el post.

[9] Son los subforos en los que se divide *Reddit*

Sujeto 4 (4612802d00)

Empieza hablando de medicamentos (*seroquel*, *lacuda*) para problemas de comportamiento. En 4 habla de una pelea con el novio con mucha ira. En 5 habla de "recaída".

En 5 de 416 se emite el resultado de **POSITIVO**.

Sujeto 5 (c1e9a86220)

Casi todos los *posts* son sobre pintura, sin que aparezcan otros temas en absoluto.

En 82 se descarta que pueda tener ideas depresivas u otro tipo de problemas.

En 82 de 393 se emite el resultado de **NEGATIVO**.

Sujeto 6 (d8c77291a1)

Se identifica inmediatamente como varón. Intereses de armas y videojuegos principalmente.

En 27 de 1224 se emite el resultado de **NEGATIVO.**

Sujeto 7 (e9b002d1ad)

Hay muchos comentarios que son respuestas descontextualizadas y por tanto no se puede saber de qué tema son. Sí hay referencias a Harry Potter y parece que es un usuario joven. Por el estilo más propio de una chica que de un chico probablemente.

En 147 de 378 no hay conversaciones que puedan sugerir problemas de autoestima o con la comida de ningún tipo, así que se emite un resultado de **NEGATIVO**.

Sujeto 8 (e0861832fa)

Son todo comentarios sobre política americana de forma consistente sin que aparezcan otro tipo de temas.

En 105 de 1122 se emite el resultado de **NEGATIVO**.

Sujeto 9 (f677ab2a34)

Se identifica como chica y habla en varios comentarios de su novio. Vive en Dubai y parece ser musulmana.

Pregunta en 111 dónde comprar carne de camello. Parece una pregunta muy extraña para una persona con problemas con la comida por lo que en 111 de 259 se emite el resultado de **NEGATIVO**.

Sujeto 10 (fc1d3e006b)

Sólo hay 32 comentario sobre temas muy diversos, así que en 32 de 32 se emite el resultado de **NEGATIVO**.

#	Username	Hash	Predicción	Valor verdadero
1	TRsubject195[10]	2c4246e1fd	0	0
2	TRsubject1637	4f01e9da6c	**0**	**1**
3	TRsubject1913	5e78d10dd0	1	1
4	TRsubject5127	4612802d00	1	1
5	TRsubject3614	c1e9a86220	0	0
6	TRsubject9229	d8c77291a1	**0**	**1**
7	TRsubject1604	e9b002d1ad	0	0
8	TRsubject3339	e0861832fa	0	0
9	TRsubject3132	f677ab2a34	**0**	**1**
10	TRsubject1496	fc1d3e006b	0	0

Cuadro 1.1: Resultados evaluación manual

1.2.2. Resultados

En este punto resulta evidente que se han cometido errores en la clasificación porque sólo se han emitido dos resultados como positivos, pero debería haber al menos 3 de ellos. Los resultados se resumen a continuación:

La precisión es alta (aquellos que se identifican como positivos lo son), pero la exhaustividad es realmente baja ($\frac{2}{5}$). Se ha intentado dar respuestas rápidas pero con cierta motivación de por qué se emite el resultado. Repasando los resultados observamos:

- El usuario 2 realmente tiene todo tipo de discusiones no relacionadas con la anorexia hasta que en 344 empieza ha preocuparse por el ejercicio y la comida. En ese punto los comentarios tienden a ser más oscuros y depresivos aunque sigue habiendo sitio para todo tipo de temas.

- Con el usuario 6 sucede algo similar, hasta 300+ no empieza a hablar de ejercicio y/o comida, aunque en ~200 menciona que ha sido diagnosticado con anorexia nerviosa. Pero en realidad sus comentarios parecen más asociados a una persona obsesionada con el ejercicio y la masa muscular que con trastornos de la alimentación. Parece que ha pasado la enfermedad y está en una etapa diferente, aunque menciona sus problemas varias veces.

- En cuanto al usuario 9, lo cierto es que en este caso la evaluación ha podido ser precipitada. Había ya algunos comentarios sobre comida, pero no eran nada depresivos, hasta que en 195 indica que tiene anorexia y que se está recuperando. Por otro lado, hay cierta similitud con el caso anterior, parece que el ejercicio y el levantamiento de pesas han sustituido hasta cierto punto la obsesión previa con la comida.

Las conclusiones del ejercicio, aunque pueda ser precipitada la extrapolación a partir de tan pocos resultados son:

[10]Los nombres de los usuarios se han prefijado con los valores TR(training)-VA(validation)-TE(test) dado que no eran únicos entre las distintas ediciones de CLEF

- Las personas con anorexia no necesariamente tienen por qué estar continuamente centradas en su problema y pueden tratar todo tipo de temas en el mismo período. Parece evidente, pero hay que tener en cuenta que una gran parte de los comentarios sean "neutrales" con respecto a la anorexia y simplemente deban ser ignorados por su escaso valor.

- El *dataset* parece contener personas que ya están en un período de recuperación de la enfermedad. De cara al resultado "competitivo", no es demasiado relevante, pero en un contexto real estas personas no serían objeto de las intervenciones y deberían ser identificadas (tal vez con una tercera etiqueta) y descartadas del cribado.

- Se observan también admisiones muy claras de la enfermedad y por lo que parece no se han eliminado del dataset. Ésto es peligroso a la hora de utilizar métodos poco interpretables ya que se pueden estar elaborando modelos que detecten admisiones explícitas de haber pasado la enfermedad y estar bajo tratamiento. La utilización en un contexto de detección de riesgo temprano de estos modelos no aportaría mucho valor a pesar de presentar tal vez buen rendimiento.

- La detección "temprana" en algunos casos puede ser imposible y los algoritmos que penalizan decisiones tardías pueden dejar escapar casos que se revelan más adelante.

1.3. Objetivos

Como parte de este TFM, y a propósito de lo expuesto en este capítulo introductorio, los objetivos que se plantean son:

- Revisar las aproximaciones realizadas en el contexto de la conferencia CLEF eRisk e intentar resumir cual es el "estado del arte" en cuanto al problema.

- Describir el dataset proporcionado y comentar las características que lo definen.

- Crear un modelo que, a partir de lo aprendido, "compita" en diferido en CLEF eRisk 2019, esperando que tenga resultados competitivos (con la admisión explícita de la ventaja que supone hacerlo sin las restricciones de tiempo y pudiendo aprender de los resultados obtenidos). Pero además, se pondrá especial atención en hacer que el modelo sea **interpretable**, de forma que se evite el efecto *Clever Hans* en la medida de lo posible.

- Atender no sólo a la capacidad predictiva del modelo, si no también a las posibles conclusiones (sean novedosas o meramente confirmatorias) sobre los comportamientos e intereses de las personas con anorexia y otros trastornos de la alimentación.

1.4. Estructura del resto de la memoria

La memoria, tras esta parte inicial introductoria, se estructura en 3 capítulos adicionales y un anexo.

En el capítulo segundo se centra en la descripción del "estado del arte". En cada una de sus secciones.

- Se explican las condiciones sobre las que se celebraron los laboratorios de CLEF eRisk 2019 y el procedimiento de elaboración del *dataset* que se empleó, junto con un breve análisis del mismo sin entrar en las cuestiones relacionadas con el procesamiento del lenguaje natural.
- Se definen las métricas creadas *ad-hoc* para este tipo de problemas que deben tener en cuenta la prontitud en emitir un diagnóstico.
- Se analizan los artículos publicados por los competidores con mejores resultados.

El capítulo tercero trata sobre una propuesta de implementación de un Sistema de Detección Temprana de Anorexia que siga las reglas del laboratorio en CLEF eRisk 2019. Se hace hincapié en las motivaciones que llevan a tomar cada una de las decisiones de implementación, tanto de las elegidas como de las descartadas. Se incluyen también descripciones a alto nivel de los algoritmos subyacentes sin cuya comprensión resulta difívil entender el encaje de las distintas piezas.

Por último, se evalúa su rendimiento y se hace una comparativa del mismo frente al resto de soluciones analizadas.

En el último capítulo, por un lado se intentan extraer conclusiones derivadas de la realización del TFM, prestando especial atención a las relacionadas con la interpretabilidad, y por otro se plantean posibles líneas de mejora que no se han terminado de explorar por limitaciones de tiempo.

En cuanto al anexo, consiste en una descripción breve de las herramientas utilizadas en la propuesta de implementación, así como algunos comentarios de carácter práctico relacionados con su elección y/o utilización.

Capítulo 2

Detección Anorexia en CLEF eRisk

Como se ha definido en el capítulo anterior, uno de los objetivos del TFM es el de revisar los avances realizados en el contexto de los dos laboratorios realizados en las ediciones sucesivas de 2018 y 2019. Tomando los acercamientos de estos años como punto de partida se propondrá e implementará un sistema que aborde la misma tarea en el capítulo siguiente.

El formato en el que se desarrollaron los laboratorios es el siguiente:

- En la edición de 2018, los datos se dividieron en 10 trozos o *chunks*, que contenían aproximadamente el mismo porcentaje de comentarios de todos los usuarios y que estaban ordenados cronológicamente. Cada semana se publicaba uno de estos *chunks* para cada usuario y cada equipo participante debía elegir entre:
  - Emitir un resultado (positivo o negativo) para el usuario
  - No emitirlo y esperar al *chunk* de la semana siguiente
- En 2019 se eligió un modelo distinto, en el que la decisión pasa a ser individualizada para cada comentario de cada usuario. En este caso se puede emitir un resultado tras el primero, tras el segundo, etc... hasta un máximo de 2000 que es límite impuesto por la API de *Reddit* y por tanto el máximo que puede contener el *dataset*.

  Además del resultado (que es vinculante y no permite recibir los comentarios siguientes de ese individuo), debía comunicarse una puntuación en el rango $[0, 1]$ que indicaría el nivel de riesgo asignado por el sistema a cada uno de los individuos. Esta puntuación no tenía que ser estrictamente una probabilidad (aunque es una de las opciones), sino que permitiría conocer de forma comparativa el nivel de riesgo de padecer anorexia que asigna al sistema a cada uno de los individuos por medio de la elaboración de un *ranking* (e.g: el usuario con id=2008 es el de más riesgo, el id=32 el siguiente, etc...).

El resumen con los resultados de cada edición se recoge en Losada et al. (2019, 2018) .

2.1. Análisis del corpus 2018

El análisis se va a limitar al de la edición de 2018, dado que hacerlo sobre el 2019 sesgaría los resultados del método de clasificación implementado en el capítulo siguiente. En 2019, se consideró que los datos de 2018 (entrenamiento y los de competición) eran un conjunto de entrenamiento suficiente para esa edición.

2.1.1. Generación

La generación del corpus empleado sigue el procedimiento descrito en Losada and Crestani (2016), con ligeras variaciones dado que el dataset en ese caso estaba centrado en detectar usuarios con depresión (otra de las tareas competitivas del laboratorio) y no con anorexia. El mecanismo descrito en dicho artículo sigue los siguientes pasos:

1. **Selección de la fuente de datos**. En este punto, se analizan varias redes sociales que podrían ser eficaces para construir un *dataset* de estas características. En este caso se elige *reddit* porque no tiene restricciones de número de caracteres y por tanto permite comentarios más elaborados, además de ser una red con multitud de usuarios y subforos (*subreddits*), por lo que es posible obtener datos utilizables tanto por su cantidad como por su calidad y variedad.

2. **Identificación de usuarios con la condición a evaluar (muestras positivas)**. Se utilizan reglas de detección de auto-diagnóstico (por ejemplo "He padecido anorexia") que permitan con cierta seguridad identificar sujetos que cumplan la condición.

3. **Identificación del grupo de control (muestras negativas).** Éstos se dividen en:

    - usuarios elegidos aleatoriamente de todo *reddit*
    - usuarios activos en subforos relacionados con la enfermedad en cuestión, pero que no la sufren (profesionales, voluntarios, familiares, etc...). Se pueden considerar "señuelos" que los mejores algoritmos de clasificación serán capaces de no clasificar como falsos positivos.

Destacar que este modo de selección tiene cierto sesgo, puesto que los usuarios identificados en el paso 2 o son conscientes de su problema (veíamos en el primer capítulo que estos pacientes no siempre admiten su problema en determinados estadios de la enfermedad) o ya han sido diagnosticados y tratados por lo que la intervención temprana no es necesaria.

No obstante, obtener un *dataset* que no tuviese este problema requeriría identificar a pacientes en la vida real (no bajo seudónimo) lo que dificultaría que alcanzase un tamaño suficiente debido a la regulaciones de protección de datos aplicables a este tipo de estudios. Hacerlo así además no evitaría la introducción de otro tipo de sesgos como por ejemplo el de selección.

De cada usuario se seleccionan hasta 1000 *posts* y 1000 comentarios, lo máximo que permite la API, que se agregan en ficheros XML independientes junto con algunos otros metadatos, como la fecha y hora en que se han producido.

2.1.2. Descripción

El siguiente cuadro muestra la distribución por etiquetas y conjunto de datos (entrenamiento o test) de la edición de 2018. Como se ha puesto de manifiesto anteriormente, la proporción entre clases tiene una relación $9 : 1$, que debe mantenerse constante en las muestras para que resulten representativas.

	Entrenamiento	*Test*	**Total**
Negativo	279	132	**411**
Positivo	41	20	**61**
Totales	**320**	**152**	**472**

Cuadro 2.1: Datos proporcionados en 2018. Distribución de usuarios por etiquetas y conjunto de datos.

Se observa bastante regularidad entre ambas clases en las características no puramente textuales, lo que obliga a afrontar el problema de clasificación atendiendo al análisis semántico de los comentarios. Como ejemplo, vemos que la distribución del número de comentarios por usuario es bastante similar:

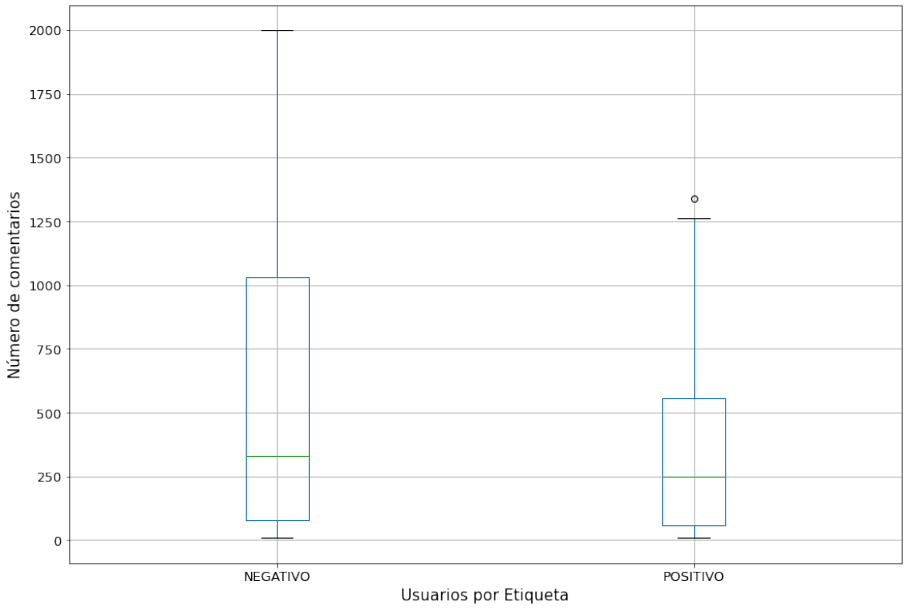

Figura 2.1: Distribución del número de comentarios en el dataset por etiqueta

O que la distribución horaria del momento en el que se realizan los comentarios[1], tampoco

[1]Esta métrica se ve afectada además por la falta de datos de zona horaria del usuario en cuestión. Aunque el

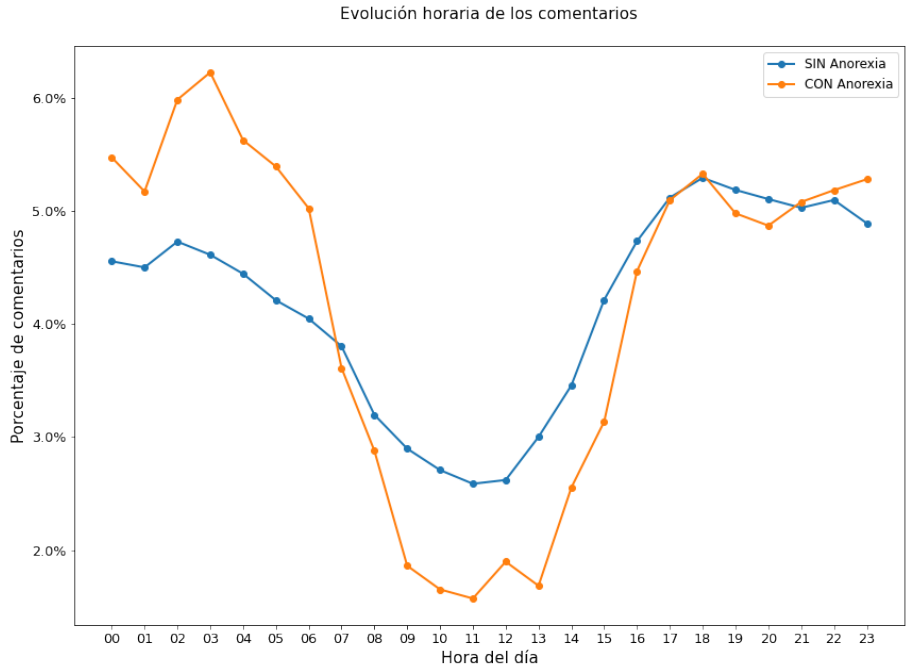

Figura 2.2: Distribución porcentaje de comentarios según la hora del día en el que se producen.

muestra tendencias demasiado diferentes. La mayor variabilidad de las tendencias de la clase positiva (variaciones más abruptas) es debido a que existen muchos menos datos en esta clase.

En resumen, para la resolución del problema es necesario recurrir a técnicas de Procesamiento del Lenguaje Natural, sin que *a priori* parezcan relevantes otras características presentes en el conjunto de datos.

2.2. Métricas de evaluación

Como resulta previsible, se utilizan las métricas típicas para evaluar el rendimiento de los algoritmos clasificación:

- La precisión o el número de aciertos entre los resultados identificados como positivos $\frac{TP}{TP+FP}$
- La exhaustividad (*recall*) o los aciertos entre las instancias positivas del *dataset*: $\frac{TP}{TP+FN}$
- La *F1* o media armónica calculada sobre entre las dos magnitudes anteriores. Actúa como "compromiso" entre ambas dado que resultaría trivial maximizarlas individualmente: $2 \times \frac{precision \times recall}{precision + recall}$

Sin embargo, el hecho de que el problema de fondo consista en la detección **temprana**, obliga a introducir nuevas medidas relacionadas con el momento en el que se da la alerta que complementen

usuario de Reddit es típicamente norteamericano, en este país

a estas métricas. A menudo será preferible un algoritmo más rápido a riesgo de ser menos preciso si permite realizar intervenciones en estadios menos desarrollados de la enfermedad.

2.2.1. Early Risk Detection Error (ERDE)

Esta métrica pretende incorporar el número de mensajes procesados hasta el instante de la predicción[2] al cálculo del error incurrido. De esta manera, una identificación tardía de la enfermedad puede ser no muy diferente a un error por falso negativo, dado que el paciente no va a poder ser atendido a tiempo como para que la intervención sea significativa.

Los parámetros que es preciso definir para su cálculo son:

- Penalización por falso positivo c_{fp}
- Penalización por falso negativo c_{fn}
- Penalización por auténtico positivo. Es un término calculado dependiente del momento de detección (k) y una constante $c_{tp} \times lc_o(k)$
- **No hay penalización en los aciertos negativos** (no hay daño en emitir un resultado tardío negativo).

Las constantes c_{fp} y c_{fn} se deben ajustar al problema en cuestión. En este caso, se determina un coste de 1 (el máximo error) en un falso negativo dado que genera el máximo riesgo y un coste del falso positivo de 0.1296 que corresponde a la proporción de positivos en el dataset.

El término más complejo de calcular es el del acierto en positivos. Se considera que emitir un diagnóstico acertado en el último momento es tan peligroso como no llegar a hacerlo nunca ($c_{tp} = 1$) por lo que sólo queda asignar un valor a $lc_0(k)$ que es una función que emite valores en el rango $[0, 1]$.

La fórmula elegida para este término de penalización es:

$$lc_0(k) = 1 - \frac{1}{1 + e^{k-o}} \tag{2.1}$$

Según esta fórmula, si $k - o$ es positivo, el cociente tenderá (exponencialmente) a 0 y por tanto el resultado será prácticamente 1, alcanzando la máxima penalización. En cambio si $k - o$ es negativo ocurrirá lo contrario.

Por tanto k no es utilizado en solitario, sino que se complementa la métrica con un valor o, que modela el mínimo número de comentarios que en promedio hay que procesar para poder emitir un resultado. De no hacerlo así, el rango efectivo sería $[0{,}5, 1]$ y con un crecimiento exponencial

[2]Nótese que la frecuencia de los comentarios de los distintos usuarios puede variar en gran medida por lo que en un entorno realista la métrica a valorar sería el tiempo transcurrido hasta la detección. Sin embargo, y a pesar de que se recoge en el dataset el momento de cada comentario, sería muy difícil poder comparar los algoritmos de forma que se asume una frecuencia de producción de textos similar.

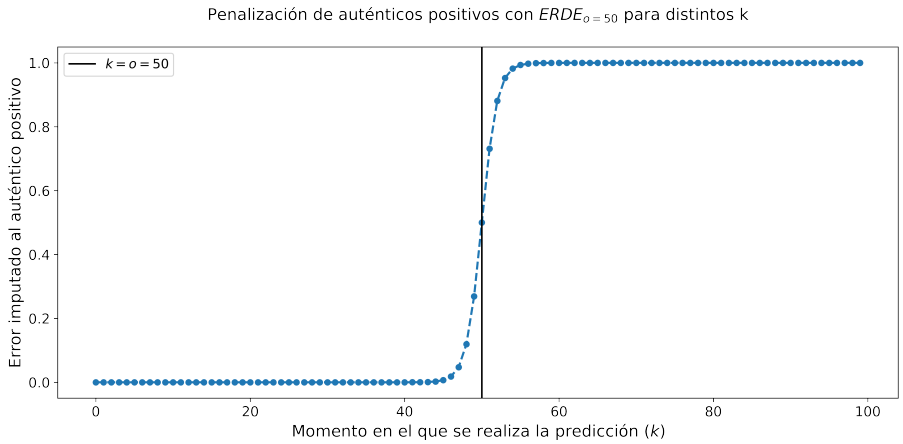

Figura 2.3: Coste de Latencia(lc) para $o = 50$ en $ERDE$

instantáneo la métrica carecería de interés. El comportamiento del coste de latencia se puede ver en 2.3.

Esta métrica no está exenta de problemas, los cuales son analizados en la edición de 2019, destacando:

- La penalización asciende rápidamente a 1 en cuanto se cruza el umbral.
- Aún identificando a un auténtico positivo en el momento $k = 1$ no es posible tener un error de exactamente 0.
- Si k representa *chunks* en lugar de *posts* individuales que contienen un porcentaje del total de comentarios de un usuario (el modo de funcionamiento de la edición de 2018) el resultado es poco representativo ya que depende de la frecuencia de publicación del usuario.
- $ERDE$ no es interpretable directamente. Se expresa como porcentaje, pero no se puede deducir exactamente cuánto mejor es un valor que otro, más allá del hecho de que valores menores son preferibles a mayores.

2.2.2. $F_{latency}$

$F_{latency}$es una métrica alternativa a $ERDE$ diseñada también con el objetivo específico de introducir la velocidad con la que un sistema es capaz de emitir resultados correctos en la clase positiva, pero que no tenga las deficiencias identificadas en el final del apartado anterior.

Se basa en el cálculo de F_1, el cual es ponderado por una estimación de la velocidad del algoritmo:

$$F_{latency} = F_1 \times speed \tag{2.2}$$

Este parámetro de "velocidad", se calcula a partir del rendimiento sobre los auténticos positivos detectados, de acuerdo a la siguiente definición:

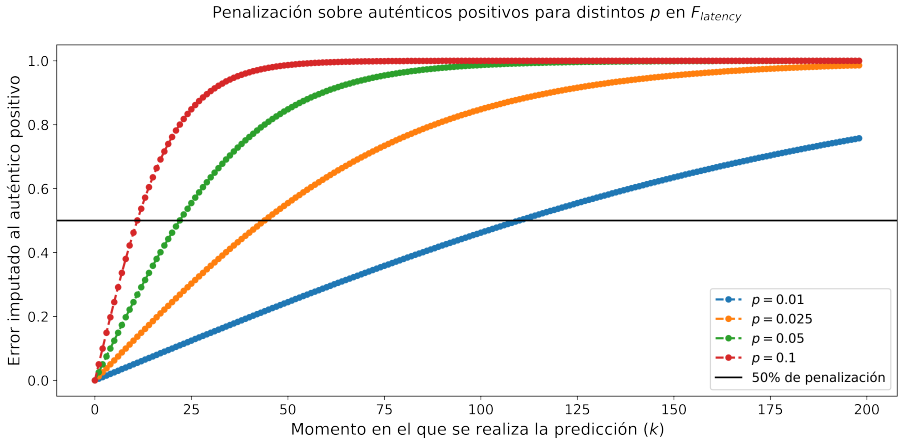

Figura 2.4: Penalización en $F_{latency}$

$$speed = 1 - median\{penalty(k_u) : u \in U, d_u = g_u = 1\} \tag{2.3}$$

O lo que es lo mismo: 1 menos la penalización *mediana* de los auténticos positivos detectados por el sistema. A su vez, esa penalización se calcula:

$$penalty(k_u) = -1 + \frac{2}{1 + e^{-p \cdot (k_u - 1)}} \tag{2.4}$$

Donde k_u es el "instante" en el que se ha emitido un resultado para el usuario u y p es un parámetro que indica la velocidad del crecimiento de p. Esta penalización es mucho más gradual que el coste de latencia de $ERDE$.

En la figura 2.4 se puede ver el efecto de la elección del parámetro p. En el caso de CLEF 2019, se eligió un valor de p tal que el acierto en la mediana de comentarios supusiera un $0,5$ de penalización (línea negra en el gráfico).

2.2.3. P@10 y NDCG

Otra forma de evaluar los algoritmos consiste en utilizar métricas basadas en un *ranking* de vulnerabilidad a la anorexia calculada para cada usuario. En un sistema dedicado a identificar y asistir a personas en peligro de anorexia en el que los recursos disponibles fuesen limitados, resultaría útil ordenarlas de acuerdo a la vulnerabilidad asignada por el sistema.

De esta manera, las personas en mayor riesgo, siempre según el algoritmo, serían candidatas a ser evaluadas antes que las de menor. Se maximizaría así la efectividad de las intervenciones realizadas por profesionales sanitarios.

Para este propósito se utilizan estas 2 métricas del área de la *Recuperación de la Información (IR)* evaluadas en distintos momentos k:

- *P@10* o "Precisión en 10" que consiste en evaluar la precisión únicamente sobre los 10 resulta-

dos más relevantes. De esta manera si de los 10 resultados con mayor probabilidad de padecer anorexia 8 lo son, su valor será de $0,8$.

- *NDCG@10* o *Normalized Discounted Cumulative Gain at 10.* Esta métrica requiere de la definición algunos conceptos previos.

  Cumulative Gain que se define como la suma de la relevancia de los resultados obtenida en una lista de resultados (los 10 primeros de la lista ordenada en este caso):

$$CG_p = \sum_{i=1}^{p} rel_i \tag{2.5}$$

  La relevancia *rel* no se define en este caso concreto e implicaría que existe una ordenación óptima de la gravedad de los casos elegidos, si bien no se hace referencia a ella en ningún otro lugar. Por este motivo se deduce que se utiliza una escala binaria (positivo=1, negativo=0), si bien no está explicitado en Losada et al. (2019) ni en Losada et al. (2018).

  Este parámetro no tiene en cuenta la ordenación de los resultados(los más relevantes deberían estar antes), por lo que se modifica de la siguiente manera para obtener el *Discounted Cumulative Gain*:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{2.6}$$

  Este resultado ya permite comparar dos listas de resultados, pero sigue sin ser una medida del todo interpretable, por lo que se añade una última transformación, que consiste en obtener el ratio entre el DCG_pobtenido y el DGC_p **ideal** que representaría a la lista con la máxima relevancia:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{2.7}$$

2.3. Resumen de las aproximaciones al problema en 2019

En esta sección, se hace un análisis de varias aproximaciones presentadas en la edición de 2019, haciendo énfasis en aquellas que mejores resultados han obtenido. Destacar que cada equipo podía competir con hasta 5 algoritmos distintos.

Los resultados de la edición de 2019 se pueden ver en el cuadro 2.2. Cada una de las columnas que se muestran en la tabla corresponden a las siguientes magnitudes:

- Nombre del equipo.
- Ejecución (cada equipo podía presentar hasta 5 algoritmos distintos identificados por números de ejecución numerados del 0 al 4).
- Precisión (P), Exhaustividad (R) y F1.

- Resultados de la métrica "ERDE" descrita en la sección 2.2.1 para los valores $o = 5$ *($ERDE_5$)* y $o = 50$ *($ERDE_{50}$)*.
- Latencia (*latencyTP*) o número *mediano* de ejecuciones en las que se emitió un resultado de verdadero positivo.
- Velocidad (*speed*) según se define en la fórmula 2.3.
- F1 ponderada (*latency-weighted F1*) según se define en la fórmula 2.2.

De todos éstos equipos competidores se ha procedido a hacer una selección de aquellos que, además de tener unos resultados aceptables, destacaban por algún motivo específico, que se detalla a continuación:

- **UppsalaNLP.** Esta opción compara diversas técnicas de representación vectorial para los textos (Random Indexing, GloVe y ELMo), lo que permite generalizar su capacidad para capturar las características semánticas apropiadas para este problema.
- **BiTeM.** Consiste en un método "simple" que a pesar de ello es competitivo. Una *SVM* con una buena elección de hiperparámetros y de términos considerados relevantes permite compensar la pérdida de matices producida por la agregación de los textos de los usuarios en un único documento.
- **lirmm.** Aquí destaca el uso del *Transfer Learning*; a partir de modelos preentrenados es posible obtener buenos resultados sin aumentar la complejidad del modelo ni hacerlo excesivamente *ad-hoc*.
- **CLaC**. Este equipo es el que ha obtenido el mejor resultado de todos a partir de una solución 100 % basada en técnicas de *Deep Learning*. Además de su rendimiento destaca por no aplicar apenas técnicas de *feature engeneering*, lo que demuestra la potencia de este tipo de soluciones. Es además la que computacionalmente requiere más recursos por el alto número de parámetros de los modelos.
- **UDE**. Al igual que en el caso de **BiTeM**, este resultado es destacable debido a que una aproximación simple, tanto que no parecería más que una prueba de contexto, es la que mejor resultado consigue de las 4 presentadas por el equipo y es competitiva con algoritmos muy complejos. También concatena todos los textos de un usuario para su procesamiento.
- **UNSL.** En este caso durante la construcción del modelo se hace especial énfasis en la interpretabilidad y explicabilidad del mismo. Siendo que la mayoría de ellos dejan este aspecto a un lado y se centran únicamente en conseguir el mejor rendimiento posible, esta intención resulta destacable.

team	run	P	R	F1	ERDE5	ERDE50	latencyTP	speed	latency-weighted F1
UppsalaNLP	0	.32	.44	.37	5.83 %	5.77 %	1	1	.37
UppsalaNLP	1	.36	.39	.37	6.13 %	6.07 %	1	1	.37
UppsalaNLP	2	.34	.42	.38	5.88 %	5.81 %	1	1	.38
UppsalaNLP	3	.39	.30	.34	6.68 %	6.63 %	1	1	.34
UppsalaNLP	4	.40	.42	.41	5.73 %	5.66 %	1	1	.41
BioInfo@UAVR	0	.32	.44	.37	5.84 %	5.77 %	1	1	.37
BiTeM	0	.42	.07	.12	8.58 %	8.42 %	1	1	.12
BiTeM	1	.44	.70	.54	5.89 %	3.40 %	3	.99	.54
BiTeM	2	.73	.11	.19	8.42 %	8.01 %	3	.99	.19
BiTeM	3	1	.01	.03	8.84 %	8.83 %	1	1	.03
lirmm	**0**	**.74**	**.63**	**.68**	**9.13 %**	**5.14 %**	**21**	**.92**	**.63**
lirmm	1	.77	.60	.68	9.10 %	5.51 %	21	.92	.62
lirmm	2	.66	.70	.68	9.24 %	5.81 %	31	.88	.60
lirmm	3	.74	.42	.54	9.08 %	6.62 %	31	.88	.48
lirmm	4	.57	.75	.65	9.41 %	7.32 %	2023	3e-7	2e-7
CLaC	0	.45	.74	.56	6.72 %	3.93 %	7	.98	.54
CLaC	**1**	**.61**	**.82**	**.70**	**5.73 %**	**3.13 %**	**4**	**.99**	**.69**
CLaC	**2**	**.60**	**.81**	**.69**	**6.02 %**	**3.13 %**	**6**	**.98**	**.68**
CLaC	**3**	**.63**	**.76**	**.69**	**6.27 %**	**3.55 %**	**7**	**.98**	**.68**
CLaC	**4**	**.64**	**.79**	**.71**	**6.25 %**	**3.43 %**	**7**	**.98**	**.69**
SINAI	0	.12	.97	.21	10.58 %	6.59 %	5	.98	.21
SINAI	1	.11	.99	.20	10.80 %	6.76 %	5	.98	.20
SINAI	2	.18	.95	.30	9.04 %	4.89 %	8	.97	.30
HULAT	0	.11	.30	.17	10.84 %	8.14 %	16.5	.94	.16
HULAT	1	.11	.30	.17	10.84 %	8.14 %	16.5	.94	.16
HULAT	2	.11	.30	.17	10.84 %	8.14 %	16.5	.94	.16
HULAT	3	.11	.30	.17	10.84 %	8.14 %	16.5	.94	.16
HULAT	4	.11	.30	.17	10.84 %	8.14 %	16.5	.94	.16
UDE	0	.51	.74	.61	8.48 %	3.87 %	11	.96	.58
UDE	1	.44	.73	.55	7.48 %	3.94 %	9	.97	.53
UDE	2	.13	.68	.22	12.52 %	8.21 %	35	.87	.19
SSN-NLP	0	.32	.16	.22	8.24 %	7.76 %	2	1	.22
SSN-NLP	1	.30	.22	.25	7.90 %	7.41 %	1	1	.25
SSN-NLP	2	.47	.22	.30	7.80 %	7.19 %	2	1	.30
SSN-NLP	3	.48	.26	.34	7.61 %	6.86 %	2	1	.33
SSN-NLP	4	.32	.15	.21	8.08 %	7.86 %	1	1	.21
Fazl	0	.09	1	.16	17.11 %	13.91 %	97	.64	.11
Fazl	1	.09	1	.16	17.11 %	13.79 %	88	.67	.11
Fazl	2	.09	1	.16	17.11 %	11.22 %	34	.87	.14
UNSL	0	.42	.78	.55	5.54 %	3.92 %	2	1	.55
UNSL	1	.43	.75	.55	5.68 %	4.10 %	2	1	.55
UNSL	2	.36	.86	.51	5.56 %	3.34 %	2	1	.50
UNSL	3	.35	.85	.50	5.59 %	3.49 %	2	1	.49
UNSL	4	.31	.92	.47	6.14 %	2.97 %	3	.99	.46
LTL-INAOE	0	.45	.75	.57	7.78 %	4.23 %	11	.96	.54
LTL-INAOE	1	.47	.75	.58	7.74 %	4.20 %	11	.96	.55
INAOE-CIMAT	**0**	**.56**	**.78**	**.66**	**9.30 %**	**3.98 %**	**15**	**.95**	**.62**
INAOE-CIMAT	2	.58	.77	.66	9.28 %	9.16 %	65	.76	.50
INAOE-CIMAT	**3**	**.67**	**.68**	**.68**	**9.17 %**	**4.75 %**	**20**	**.93**	**.63**
INAOE-CIMAT	4	.69	.63	.66	9.13 %	5.08 %	20	.93	.61

Cuadro 2.2: Resultados edición 2019 (eliminadas las ejecuciones que no llegaron a generar resultados válidos).

- **LTL-INAOE.** El algoritmo de clasificación se basa en una serie de métricas desarrolladas específicamente para problemas de identificación de la autoría de textos, centrado en el análisis de frases redactadas en primera persona. Teniendo en cuenta que en la elaboración del *dataset* se han introducido personas que sin tener anorexia sí participan en foros relacionados con ésta (por su condición de profesionales o familiares de pacientes) y que por ése motivo no hablarán de sus experiencias en primera persona, este enfoque tiene la capacidad de evitar la clasificación incorrecta de estos "señuelos" como falsos positivos.
- **INAOE-CIMAT.** El trabajo de este equipo destaca por la utilización de una recurso léxico externo que puntúa las palabras más frecuentes en inglés en 10 dimensiones que representa su grado de asociación con 8 emociones características (Ira, Alegría, Miedo, etc...) así como su nivel general de positividad y negatividad. También resulta de interés el hecho de que este recurso léxico es de utilidad a la hora de interpretar los resultados.

2.3.1. UppsalaNLP Elena Fano1 and Nivre1 (2019)

El acercamiento propuesto en este artículo distingue dos fases en la resolución del problema: la generación de la representación numérica de los elementos léxicos y la clasificación realizada a partir de estas representaciones.

Para el primer paso, se eligieron 3 representaciones vectoriales distintas de forma que se pudiesen comparar sus resultados. Éstas eran:

- Random Indexing.
- GloVe.
- ELMo.

En cuanto a la fase de clasificación, se orquestó una solución basada en 2 subclasificadores:

- Un **clasificador de comentarios** basado en una *Red Neuronal Recurrente (RNN)* implementada con *Celdas de Memoria Largas-Cortas (LSTM)*. Este tipo de arquitectura de Red Neuronal, donde la salida de la capa oculta de un lapso de tiempo se convierte en entrada de uno posterior, está adaptado al procesamiento de secuencias, como lo son en este caso los comentarios ordenados temporalmente de cada usuario. Este clasificador emite la probabilidad de que un texto determinado pertenezca a la categoría de "en riesgo de anorexia".
- Un **clasificador de usuarios** que recibe como entradas:
  - La salida del clasificador de comentarios.
  - La media y desviación estándar de las puntuaciones vistas hasta el momento.
  - El promedio del 20 % de textos con mayor puntuación.

- La diferencia entre el promedio de los textos del 20 % superior e inferior.

Para este clasificador, se ha probado con dos arquitecturas, una que utiliza la *regresión logística* y un *perceptrón multicapa*. Los distintos algoritmos que compiten en CLef resultan de combinar distintas representaciones vectoriales y distintos clasificadores.

No obstante, esta solución por sí sola no está del todo adaptada a las necesidades de la competición, porque se valora emitir un resultado lo antes posible. Ésto hace obligatorio el desarrollo de una **estrategia de parada**, en la que el sistema emite una decisión sin observar todos los datos. En este caso, y dado que el resultado del segundo clasificador se puede interpretar como una probabilidad, bastaba con elegir un *umbral de corte* a partir del cual se emite el resultado positivo y se para la evaluación (las predicciones negativas no sufren de ninguna penalización por lo que se pueden seguir evaluando hasta el final).

2.3.2. BiTeM Naderi et al. (2019)

Este equipo utilizó varios enfoques diferentes para cada uno de los clasificadores que hicieron competir:

1. El primero de ellos es un simple *Bag Of Words*, donde los comentarios de un usuario son agregados en un único documento y se construye una representación del mismo utilizando la técnica conocida como *Term Frecuency-Inverse Document Frecuency (tf-idf).* Esta técnica es una de las más conocidas para representar documentos completos de un corpus y extraer métricas de similaridad entre los mismos. Para el paso de clasificación se utilizó una *Support Vector Machine (SVM)*, a partir de la representación vectorial.

2. En este acercamiento, se partió de la identificación de los 200 n-gramas (para $n = 1, 2, 3$) más representativos de *subreddits* que podrían estar relacionados con la categoría positiva (los relacionados con anorexia) y negativa (otros *subreddits* como */r/funny* o */r/movies*). Con esta información, cada comentario quedó "clasificado" como positivo si contenía más n-gramas positivos y negativo si ocurría lo contrario.

   A partir de esta información, se identificaron los 200 n-gramas más representativos según el criterio de *información mutua* y cada documento-usuario agregado quedaba caracterizado por la presencia o no de estos n-gramas. Esta información era después utilizada en clasificador logístico.

3. Para este clasificador, se utilizó una *Red Neuronal Convolucional (CNN),* a la que se le proporciona una secuencia de palabras (hasta 300) representada por la forma vectorial resultante de aplicar el algoritmo *word2vec* con 200 dimensiones. Cada comentario queda por tanto representado por una matriz 300×200 que es entregada a la *CNN* para su clasificación.

4. Este último método es una combinación o *ensemble* de los resultados de los anteriores clasificadores.

No se ha observado una mención expresa a los criterios de parada utilizados para emitir una respuesta temprana. Tampoco se ha podido determinar en alguna de las aproximaciones cómo se obtiene el resultado agregado a partir de los diferentes comentarios individuales.

2.3.3. lirmm Ragheb et al. (2019)

En este caso, el equipo identificó estos 3 pasos fundamentales en la resolución del problema:

1. **Módulo de Vectorización de Textos**. Encargado de la generación de la representación vectorial a partir de los documentos del *corpu*s.

2. **Módulo de Evaluación del Estado de Ánimo**. En este paso se asigna a cada documento una puntuación similar a una probabilidad. La salida del mismo es una *serie temporal* que representa la variación del estado de ánimo de un usuario a lo largo del tiempo.

3. **Módulo de Modelado Temporal**. Este paso procesa la serie temporal anterior para obtener el resultado final del problema de clasificación.

Los dos primeros pasos se unifican en una solución basada en la arquitectura *Universal Language Model Fine-tuning for Text Classification (ULMFiT)* Howard and Ruder (2018). Esta arquitectura pretende emular lo conseguido en cuanto a *Transferencia del Conocimiento* en el ámbito de la *Visión por Computador*.

La *Transferencia del Conocimiento* se basa en la hipótesis de que determinados problemas resolubles mediante Redes Neuronales, en este caso de procesamiento del lenguaje pero típicamente de reconocimiento de imágenes, comparten en su raíz subproblemas similares (e.g: reconocimiento de aristas, identificación de objetos, etc...). Esta similitud permite que de la resolución de uno de los problemas, para el que existan grandes cantidades de datos de entrenamiento, se obtengan submodelos que puedan ser utilizados como componentes en el resto.

Estos modelos preentrenados ahorran una enorme cantidad de procesamiento y son distribuidos en forma de capas de *Redes Neuronales Profundas*. Éstas capas "congeladas" o "semicongeladas" (no necesitan entrenarse completamente al haberlo sido ya) sirven para obtener las características de bajo nivel que las capas superiores *ad-hoc* utilizarán para resolver la tarea de clasificación objetivo. El proyecto *ImageNet*[3] es el arquetipo en el que se inspira *ULMFiT.*

En *ULMFit*, se distinguen 3 fases:

1. Una primera en la que se entrena el modelo con gran cantidad de datos textuales. El *dataset* empleado para la demostración de sus posibilidades en el artículo original es el *Wikitext-103*

[3] *https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures*

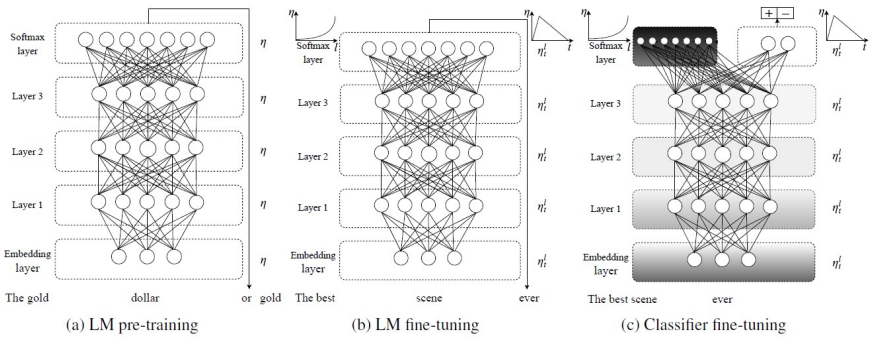

Figura 2.5: Las 3 fases en *ULMFit*, preentrenamiento del *Modelo de Lenguaje*, afinado con datos de la tarea y entrenamiento final del problema de clasificación.

que consiste en 28.000 artículos de la *Wikipedia* con un total de 103 millones de palabras. La arquitectura de la *Red Neuronal Profunda* se basa en varias capas *LSTM* que serán capaces de capturar relaciones temporales entre las palabras.

2. Posteriormente se procede a realizar un entrenamiento del *Modelo de Lenguaje* con los datos de la tarea objetivo de forma que se capturen sus características únicas. Este entrenamiento se realiza utilizando una técnica de *afinado discriminativo*, que utiliza distintas tasas de aprendizaje η para las distintas capas. La tasa de aprendizaje de las iniciales es menor que la de las superiores para evitar inestabilidades. Se ha demostrado que hacerlo así reduce el tiempo de entrenamiento.

3. Por último, se añaden capas adicionales que realizarán la tarea de clasificación con las etiquetas definitivas. En esta fase, además del *afinado discriminativo* se realiza un descongelamiento gradual de las capas inferiores de forma que no se produzca el "olvido" de lo aprendido en las fases anteriores.

Este procedimiento abarca las dos primeras fases identificadas por el equipo para la resolución del problema, obteniendo una serie temporal de predicciones para cada uno de los comentarios emitidos.

En el último paso, el correspondiente al Módulo de Modelado Temporal, se prueban 3 técnicas distintas:

- Un *Perceptrón Multi Capa (MLP).* Éste es el tipo de arquitectura de *Red Neuronal* clásica, en el que varias capas de *perceptrones* ("neuronas artificiales") se concatenan de manera que cada uno de ellos queda conectado con todos los de los niveles anteriores y posteriores. Estos *perceptrones* calculan una combinación lineal, ponderada por los pesos $w_{ij}^{(k)}$[4], a partir de las salidas la capa anterior. Este resultado es modificado por una función denominada "de activación", típicamente la llamada *sigmoidal* o *logística,* que a su vez será una de las entradas de la capa siguiente.

[4]El *perceptrón* j de la capa k está conectado a los i *perceptrones* de la capa anterior, con pesos $w_{ij}^{(k)}$.

El cálculo de los pesos que determinarán el comportamiento de la red se hace utilizando el algoritmo de propagación inversa o *backpropagation*.

- Un *Bosque Aleatorio (RF)*. Este tipo de algoritmo es una iteración sobre los conocidos como *Árboles de Decisión*. Esta técnica, que puede aplicarse tanto a problemas de regresión como de clasificación, funciona creando una sucesión de comparaciones "menor qué" sobre las variables de entrada. Cada una de estas comparaciones crea 2 ramas distintas por lo que el resultado acaba siendo un *árbol binario no balanceado*. Una vez calculado, partiendo del nodo raíz y de una nueva observación se puede generar una predicción.

  Cada una de estas comparaciones se elige de forma que maximice el acierto de la predicción en el conjunto de entrenamiento, pero ésto hace que sea un método muy sensible al *overfitting*. Los *Bosques Aleatorios* reducen ese problema generando varios *Árboles de Decisión* en los que en cada iteración sólo se contemplan un subconjunto de las variables (determinado de forma aleatoria, lo que da el nombre a la técnica). En el momento de la predicción estos árboles "votan" para determinar el resultado que será la salida del algoritmo.

- La técnica estadística conocida como *Inferencia Variacional Bayesiana*. Se utiliza en este caso para calcular la probabilidad de la variable inobservable "el individuo padece de anorexia" expresada como U_k a partir de las variables observables. La variable observable en este caso es la puntuación obtenida por la clasificación anterior en los M_k escritos de un mismo usuario k. Esta puntuación sobre un texto determinado se representa como W_i^k.

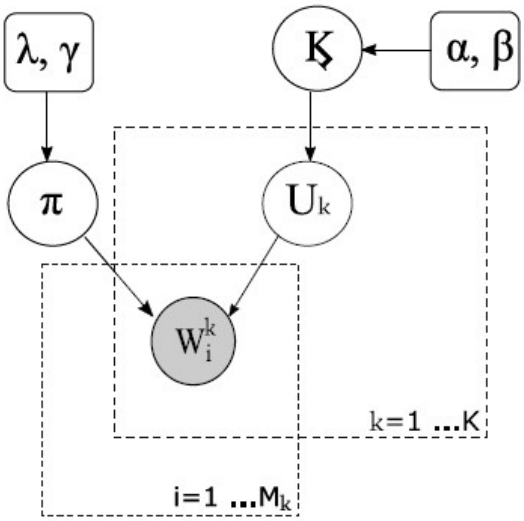

Figura 2.6: Modelo gráfico de la *Inferencia Variacional Bayesiana*. π es la probabilidad con la que un usuario k emite un escrito clasificado como "positivo". U_k es la variable inobservable que nos interesa (el usuario k padece anorexia) y W_i^k representa la clasificación del comentario i del usuario k. $\{\lambda, \gamma, \kappa, \alpha, \beta\}$ son los hiperparámetros de las distribuciones aleatorias, donde W y U son *Bernouilli* y π y κ son *Beta*.

2.3.4. CLaC Elham Mohammadi and Kosseim (2019)

La solución propuesta por este equipo se basa totalmente en una arquitectura que combina varios elementos de *Deep Learning*. El esquema general se puede ver en la figura 2.7. Cada una de las capas que actúan como componentes son:

1. **Capa de Entrada**. Cada comentario es tokenizado y enviado al generador de la representación vectorial de cada palabra. Coincide con otros equipos en probar más de un tipo de generadores de *embeddings*, en este caso GloVe y ELMo.

2. **Capas Ocultas**. Encargada de procesar los tokens vectorizados. Se utilizan hasta 4 variantes: una *Red Neuronal Convolucional (CNN)*, una *Red Neuronal Recurrente Bidireccional (BiRNN)*, una *Memoria Bidireccional de Corto-Largo Plazo (BiLSTM)* y una *Unidad Recurrente Bidireccional con Puertas (BiGRU)*. Todas ellas procesan los tokens de manera secuencial de principio a fin y en el sentido contrario. De esta manera se tienen en cuenta todos los tokens posteriores y anteriores para cada token procesado.

3. **Capa de Atención Nivel de Comentario/Capa *Pooling***. En los modelos que utilizan *CNN* como capa oculta, consiste en una una capa de *Max-Pooling*. El efecto de esta capa es el de reducir el tamaño de la entrada y por tanto de los parámetros procurando mantener las características de la misma. Es un componente típico usado en combinación con capas convolucionales en problemas de Visión por Computador.

   Para el resto de tipos de capas ocultas, se realiza un promedio ponderado de las salidas de la capa anterior para cada texto, de modo que se genera un vector salida P tal que:

   $$P = \sum_{t=1}^{n} y_t \omega_t$$

   Donde y_t es la salida de la capa anterior en el momento t y ω_t es el peso asignado a la salida en ese momento.

4. **Capa de Atención a nivel de Usuario.** En esta capa se genera de forma similar a la anterior una representación a nivel de usuario, a partir de la agregación de los valores de los comentarios.

5. **Capa de Clasificación.** La capa final consiste en una capa totalmente conectada que proyecta sobre un vector de tamaño 2 (correspondiente a las clases negativa y positiva). Una capa de activación *Softmax* da el resultado final.

6. ***Ensemble***. El resultado de la última capa podría usarse exclusivamente para caracterizar a cada uno de los usuarios, pero en este caso, se utiliza tan sólo como característica en la entrada a un clasificador *SVM*. Este modelo se alimenta con otras entradas como son las salidas del la capa de Atención a nivel de usuario.

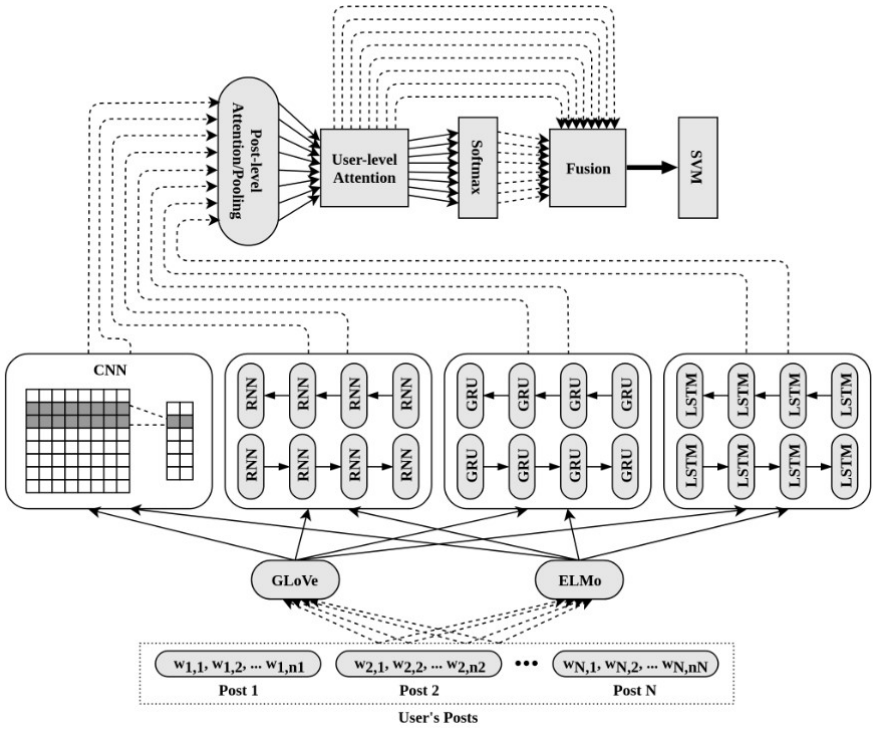

Figura 2.7: Arquitectura propuesta por el equipo CLaC

2.3.5. UDE Razan Masood (2019)

Este equipo entrenó 5 algoritmos distintos, algunos de ellos totalmente independientes:

1. **SVM.** Este intento se trata de un único clasificador basado en *Máquinas de Vector Soporte (SVM)*. El único procesamiento que se hace es el propio de la *selección de características* aplicada sobre los textos. En este caso consiste en elegir los 500 términos más significativos sobre todo el *corpus*. Posteriormente, los textos de un usuario concreto son concatenados, y se genera una representación vectorial a partir de esta concatenación que tiene en cuenta la frecuencia de cada uno de los términos selecionados. Una vez hecho ésto se realiza una búsqueda sobre los hiperparámetros de *SVM* para seleccionar aquellos que son más efectivos empleando estas representaciones vectoriales como entrada del clasificador.

2. **SVM tras filtrado.** Es una iteración del anterior que se basa en la idea de que tan sólo algunos *posts* son relevantes al estado del individuo y el resto no son realmente diferentes de los que haría cualquier otra persona (situación que ya se ha identificado en el primer capítulo en la inspección de los elementos seleccionados aleatoriamente). Estos textos "indiferentes" se eliminarían y no constituirían parte de la entrada entregada al *SVM*.

3. **Red Neuronal LSTM.** Al igual que otros acercamientos que se han visto hasta el momento, se realiza la elección natural para la clasificación de secuencias temporales: una red Neuronal

con celdas *LSTM*. En este caso, cada texto se vectoriza con la salida resultado de *doc2vec* de tamaño 200 concatenado con un vector frecuencia sobre los 70 términos más utilizados.

4. **Modelo de Atención Global.** *Atención* es un mecanismo usado en las *Redes Neuronales* de tipo *Codificador-Decodificador*, y que resuelve el problema característico de este tipo de sistemas consistente en que tienden a "olvidar" en su vector contexto las primeras entradas cuando las secuencias son muy largas. En este acercamiento, se mantienen los estados ocultos intermedios para generar el vector contexto. Dado que se aplica esta técnica a toda la entrada (todos los textos de un usuario) a esta variante del método se le conoce como *Atención Global*.

5. **Modelo de Atención Local.** Similar al caso anterior pero en este caso se presta atención a parte de la entrada y no a la secuencia completa.

2.3.6. UNSL Burdisso et al. (2019b)

En esta ocasión se ha optado por la utilización de una librería, SS3,[5] desarrollada por miembros de este equipo en el seno de la investigación del análisis de sentimiento en redes sociales.

En el artículo original en el que se describe el componente SS3 Burdisso et al. (2019a), se identifican 3 requerimientos clave:

- La clasificación debe ser **incremental.**
- Debe haber soporte para la clasificación **temprana.**
- La solución debe ser **explicable.**

Los algoritmos existentes en su formas habituales no son capaces de cumplir estos requerimientos, por lo que se desarrolla este componente ad-hoc llamado SS3 que significa *Sequential S3 (Smoothness, Significance and Sanction)*.

Este procedimiento se basa en la existencia de una función llamada gv. Esta función aplicada sobre una palabra, asigna una puntuación de confianza en su pertenencia a un conjunto de clases predefinidas c_i. La concatenación de todos estos valores es un vector:

$$gv('apple', travel) = 0$$

$$gv('apple', technology) = 0{,}8$$

$$gv('apple', food) = 0{,}4$$

[5] https://github.com/sergioburdisso/pyss3

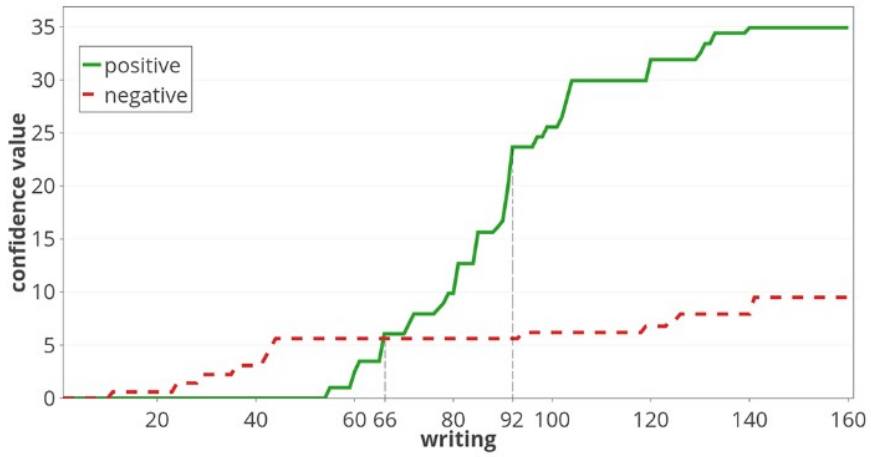

Figura 2.8: Progresión de la confianza positiva y negativa para un sujeto determinado. En el escrito 66 se produce el punto de inflexión donde la probabilidad positiva supera a la negativa.

$$gv('apple') = (0,0{,}8,0{,}4)$$

Estos vectores individuales para cada palabra se combinan de forma jerárquica utilizando un *operador resumen*. Por ejemplo, se calcularán los de las frases a partir de las palabras, de los párrafos a partir de las frases, etc... realizando una suma acumulativa de los vectores.

Por la naturaleza de estos *operadores resumen* (suma, multiplicación, etc..), los resultados se pueden combinar de forma incremental, lo que facilita generar reglas de detección temprana que se pueden hacer corresponder a determinados puntos de inflexión. Un ejemplo de criterio sería *"un sujeto se clasifica en el grupo positivo en el momento en el que se considera que es más probable que lo contrario"*.

El problema se reduce por tanto a la creación de esta función gv. Para encontrar esta función primero es necesario definir las funciones lv o *local value,* sg o *signficance value* y sn o *sanction value*.

Empezando por lv esta se puede expresar como:

$$lv_\sigma(w,c) = (\frac{P(\omega|c)}{P(\omega_{max}|c)})^\sigma \tag{2.8}$$

Donde el hiperparámetro σ se encuentra en el rango $[0,1]$ y sirve para controlar la *suavidad* de la función.

Por otro lado $sg(w,c)$ es una función que debe cumplir:

- Produce un valor cercano a 1 cuando la categoría $lv(w,c)$ es significativamente superior al resto de categorías

- Produce un valor cercano a 0 para las categorías en las que este valor es similar.

Por ejemplo, para las *stop-words* que están presentes por igual en todos los grupos el valor debería ser cercano a 0.

Por último el cometido de $sn(w, c)$ es el de decrementar el valor de w en relación al número de categoría para el que ω es significativa. De esta manera sólo estará próxima a 1 si existe una única categoría en la que este valor sea significativo.

Con todos estos datos se puede definir gv:

$$gv(\omega, c) = lv(w, c) \times sg(w, c) \times sn(\omega, c)$$

Con estas definiciones, la aplicación al problema actual se limita a calcular estos valores para cada secuencia de texto del usuario y a elegir una estrategia de parada determinada.

Un resultado de aplicar este enfoque es que las palabras que más han contribuido al valor final se pueden extraer con facilidad (y las frases, comentarios, etc...), con lo que el modelo no se comporta como una caja negra sino que es posible justificar sus decisiones.

2.3.7. LTL-INAOE Ortega-Mendoza et al. (2019)

Este caso se basa en la premisa de que son las frases redactadas en primera persona las que revelan el estado mental de una persona y por tanto sobre las que se debe actuar para realizar un buen trabajo en la clasificación. Esta técnica se conoce con el nombre DPP-EXPEI debido al nombre de las dos métricas que intervienen:

- *Discriminative Personal Purity (DPP).* Se compone de dos factores: un factor descriptivo que definido como el máximo valor de la función PP_k, que captura el nivel de ocurrencia de un término en frases personales, y un factor discriminativo basado en la función dif que representa la diferencia en el número de documentos que contienen el término $t_i$en la clase positiva.

$$DPP(t_i) = max_{k=1}^{|C|}\{PP_k(t_i)\} \times dif(t_i) \tag{2.9}$$

  A su vez $PP_k(t_i)$ se calcula:

$$PP_k(t_i) = log_2(2 + \frac{1}{2}\sum \frac{PEI(t_i, d_i) + 1}{NEI(t_i, d_i) + 1}) \tag{2.10}$$

  donde PEI o *Personal Expression Index* es una métrica que combina la frecuencia de un término en frases personales respecto a su ocurrencia en frases no personales y NEI es el opuesto, expresando el nivel de asociación de cada término a la información no personal.

  Como resultado de este paso, se obtienen aquellos términos que resultan más relevantes y lo que se usarán para la clasificación, descartando el resto. Cada usuario quedará caracterizado

por la ocurrencia de estos términos sobre el documento resultante de la concatenación de todos sus comentarios.

- *Exponential Reward of Personal Information (EXPEI).* Una vez seleccionados los términos, estos son ponderados de acuerdo al grado de contribución al documento en cuestión. Para ésto, a partir de la frecuencia de un término t_i en un documento d_j, el resultado obtenido es:

$$w_{ij} = (\sqrt{TF(t_i,d_j)})^{1-PEI(t_i,d_j)} \tag{2.11}$$

El resultado es un vector de tamaño 1000 que identifica a cada usuario y sobre el que se puede aplicar cualquier método de clasificación ordinario. En este caso se ha optado por *SVM*.

2.3.8. INAOE-CIMAT Aragón et al. (2019)

Se utiliza una técnica bautizada con el nombre de *Bag of Sub-Emotions (BoSE)*. Ésta parte de un *dataset* Mohammad and Turney (2013)que contiene, para cada palabra, una puntuación que caracteriza su carga emocional para cada uno de estos ocho sentimientos:

- *Anger* (Enfado)
- *Anticipation* (Anticipación)
- *Disgust* (Asco)
- *Fear* (Miedo)
- *Joy* (Alegría)
- *Sadness* (Tristeza)
- *Surprise* (Sorpresa)
- *Trust* (Confianza)

Además de dos características adicionales que son el nivel de Negatividad y Positividad.

Sobre las 10 dimensiones de las palabras del dataset se aplica un algoritmo de clustering que permite identificar *Sub-Emociones*. Cada uno de los documentos en cuestión (en este caso la agregación de todos los comentarios) queda caracterizado por una tabla de frecuencias de las *Sub-Emociones* que aparecen.

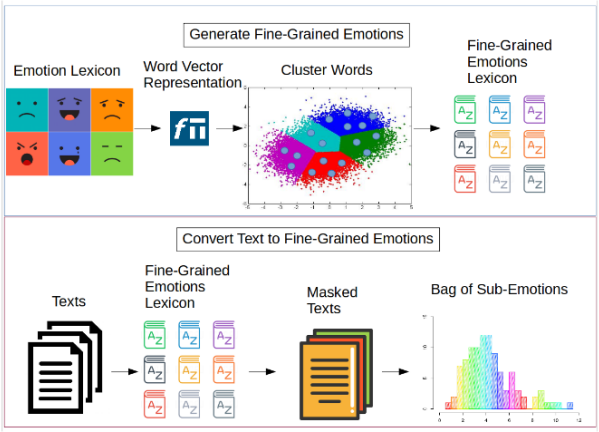

Figura 2.9: Técnica Bag of Sub-Emotions

Sobre esta representación vectorial resultante se puede aplicar cualquier método de clasificación clásico, siendo elegido en este caso *SVM*.

Capítulo 3

Propuesta de un Sistema de Detección Temprana de Anorexia

Revisadas las distintas aproximaciones que compitieron en CLEF2019, queda evaluar el nivel de complejidad que requiere el generar una estrategia comparable que hubiese podido ser competitiva en dicha edición. Resulta imposible replicar las condiciones exactas de dicho laboratorio debido a que los participantes recibieron los datos de forma escrupulosamente secuencial, con restricciones de tiempo y sin acceso a las implementaciones de sus competidores (aunque sí a las de la edición de 2018).

No obstante, sí que se ha intentado reproducir con cierta fidelidad el escenario, evaluando los resultados sobre el dataset final una única vez[1]. Adicionalmente, recordamos que en 1.3 se hacía hincapié en que el método fuese interpretable, y, de ser posible, se pudiese relacionar con alguno de los rasgos conductuales descritos como típicos de la Anorexia. De ser así aumentaría la confianza en lo que respecta a la generabilidad de los resultados.

El planteamiento general del sistema contemplaría los siguientes pasos:

1. La elección de una representación vectorial de los textos de forma que puedan ser procesados por los distintos algoritmos.

2. La asignación de una métrica o *score* a cada texto individual. Se ha visto en el capítulo anterior que no es imprescindible hacerlo así y que varias de las implementaciones acumulan varios *posts* en textos conjuntos.

   No obstante, aunque los resultados obtenidos pueden llegar a ser satisfactorios, resulta antiintuitivo de acuerdo a lo observado en el "diagnóstico" individual realizado en el primer capítulo. En éste se concluía que los *posts* de los usuarios podían tener una diversidad de temáticas elevada, tanto de aquellos sujetos que tenían anorexia como de aquellos que no, y por tanto agregarlos generaría amalgamas de texto sin continuidad semántica entre sus distintas partes.

[1] Con la salvedad de una ejecución sobre el conjunto de datos de 2019 durante la fase de prueba de concepto, lo que fue advertido como un error metodológico por parte de la dirección del TFM.

Este paso constituye en sí mismo un problema de tipo no-supervisado o semisupervisado, dado que no es posible de forma realista el etiquetado de los *posts*. En Masood et al. (2020), se parte de un etiquetado multidimensional en distintas temáticas (desorden alimenticio, medicación, dieta, etc...) y obtienen buenos resultados, pero es una estrategia no viable en este contexto.

3. La combinación de los *scores* de cada uno de los textos dentro de una serie temporal, que permita tomar una decisión que tenga en cuenta toda la evidencia obtenida hasta el momento.

4. La generación de una estrategia de parada que permita la generación de una decisión temprana y por tanto no se vea penalizada en exceso en las métricas de evaluación que ponderan según el momento temporal de la emisión del resultado.

En cuanto a la estrategia, se ha considerado pertinente realizar una prueba de concepto inicial que validase el esquema general de funcionamiento y las premisas de las que parte. En dicha prueba se esperaba confirmar que el acercamiento general al problema presentaba los suficientes indicios de eficacia. Posteriormente se procedería a la generación de un sistema más sofisticado pero basado en los mismos conceptos generales que pudiera alcanzar mejores resultados.

Los datos utilizados en esta fase son los de la edición de 2018 (una agregación de los que se proporcionaron en esa edición tanto para entrenamiento como para la competición), separando de forma aleatoria un 80 % para el entrenamiento y un 20 % para la evaluación.

3.1. Prueba de concepto

Según los pasos enumerados en el apartado anterior, se plantea una solución exploratoria con los siguientes elementos:

3.1.1. Representación vectorial

Para este paso, se parte de la representación obtenida por la aplicación del algoritmo *doc2vec* sobre cada uno de los comentarios individuales. Este algoritmo permite generar una representación vectorial de textos de cualquier longitud (desde párrafos a libros), de forma que se preserve su similitud semántica entre ellos de acuerdo a su distancia coseno en el espacio vectorial generado.

Para entender este algoritmo hay que partir del hecho de que es una iteración sobre el algoritmo conocido *word2vec*, que también permite capturar la similitud semántica si bien en este caso únicamente sobre palabras individuales.

Word2vec se basa en la **hipótesis distribucional**, que establece que palabras con el mismo significado aparecerán en contextos parecidos. Partiendo de este postulado, *word2vec* diseña un problema de clasificación en el que se pretende predecir la palabra más probable[2] a partir de aquellas que están

[2]Aquí se describe el modelo Continous Bag Of Word (CBOW), si bien es posible diseñar el problema a partir de la predicción de las palabras que acompañan a una dada, lo que se denomina Skip-gram.

dentro de una distancia determinada (lo que constituye una **ventana** de tamaño w incluyendo las palabras anteriores y posteriores). Cada palabra del vocabulario V se representa siguiendo el esquema *one-hot vector*. Es necesario además establecer un tamaño p de la representación vectorial a generar.

Así, tomando como ejemplo la frase *"el balón golpeó el ..."*[3], las palabras que tendrían mayor probabilidad de ser la correcta en un corpus deportivo serían palo (el de la portería), aro o tablero (de canasta), etc...

Este problema de clasificación se modelaría de acuerdo a una red neuronal con una sola capa oculta que tendría:

- Una capa de entrada que recibiría las w palabras contexto de tamaño $|V|$ bits.
- Una capa oculta de p neuronas.
- Una capa de salida *softmax* que emite una probabilidad para cada palabra del vocabulario.

Esta red neuronal se entrenaría a partir de un corpus representativo utilizando propagación inversa y gradiente descendente, *deslizando* la ventana sobre el mismo.

Intuitivamente, el resultado que debe obtenerse es que **palabras similares (porque aparecen en los mismos contextos) generen representaciones similares.**

En 3.1 se puede razonar sobre el funcionamiento del algoritmo. Suponiendo dos palabras w_i y w_j que aparecen en los mismos contextos (resumido en (q_1, q_2)) deben generar activaciones similares (alta o baja probabilidad dependiendo si el contexto se adapta o no al significado de la palabra). Por tanto, los pesos de las conexiones correspondientes son los que preservan ésa condición convergiendo hacia vectores similares[4].

Doc2vec es una generalización de este esquema en el que se añade un elemento adicional. Una "palabra virtual" que actúa como memoria del párrafo o documento actual.

Esta "palabra virtual" sirve de referencia sobre la temática o temáticas sobre las que versa. En el ejemplo *"el balón golpeo el ..."*, el modelo debería completar con "palo" en un texto sobre fútbol y "aro" en uno sobre baloncesto.

El modo de funcionamiento durante la fase de entrenamiento del modelo es similar, con la diferencia de que la palabra adicional no es del vocabulario, sino que es un *one-hot vector* que representa cada uno de los documentos o párrafos, y se repite para cada una de las ventanas generadas en ése documento.

En la fase de predicción, los pesos de la conexiones se mantienen fijas a excepción de las que corresponden a P_j, que convergerán a valores similares a otros textos de la misma temática.

[3]Estrictamente la palabra a predecir sería la situada en la posición central dentro de la ventana. Se ha modificado en este ejemplo para ilustrar el concepto de forma más intuitiva.

[4]Similares en el espacio euclídeo solamente si tienen frecuencias también similares, y de no ser así los vectores serán aproximadamente proporcionales. Por este motivo se suele usar la distancia coseno como medida de similaridad puesto que normaliza respecto a la frecuencia de las palabras.

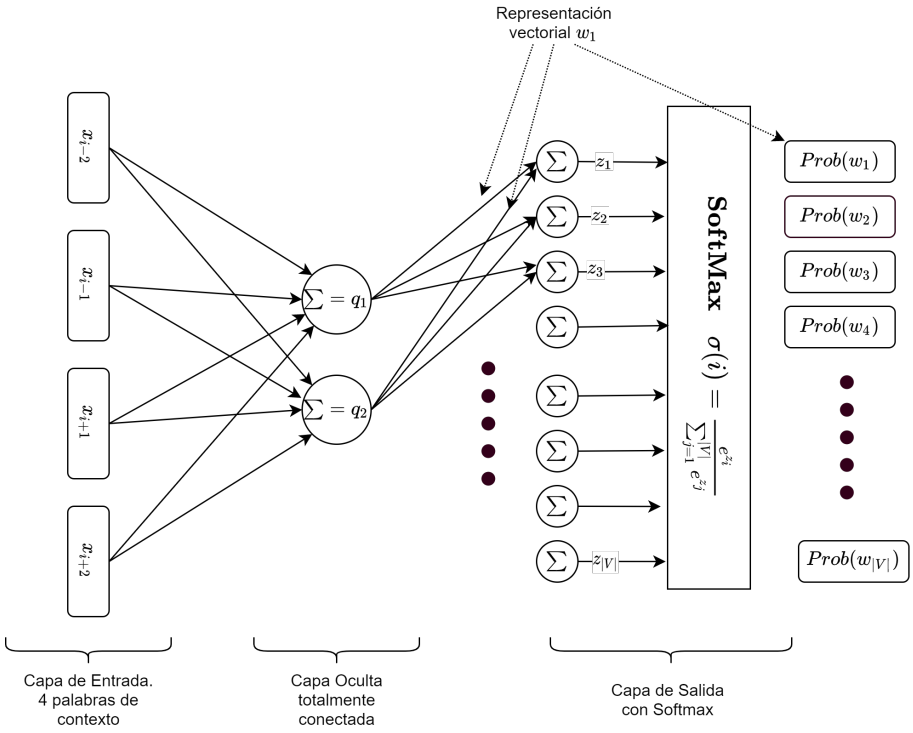

Figura 3.1: Algoritmo *word2vec*

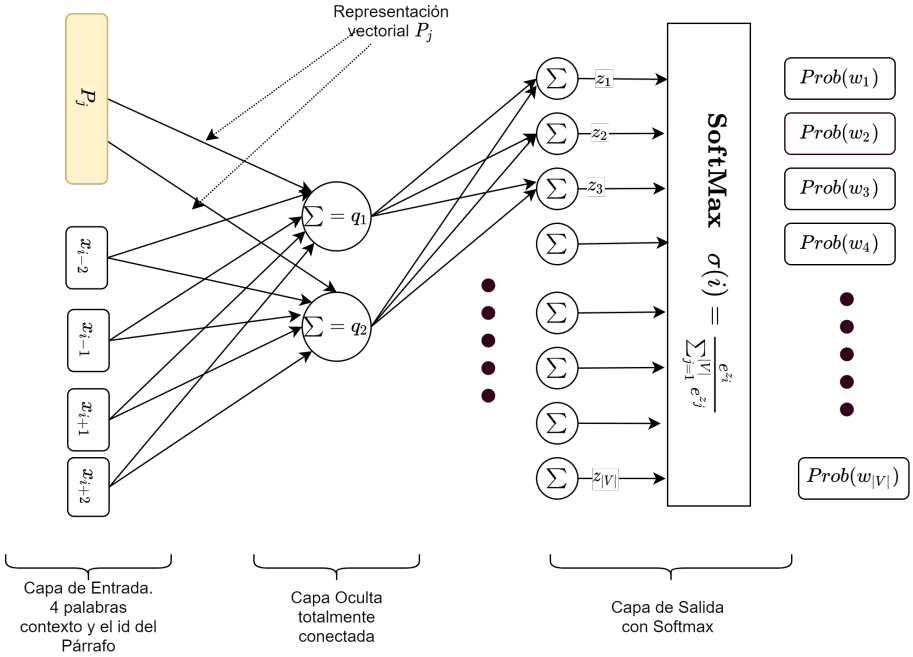

Figura 3.2: Algoritmo *doc2vec*

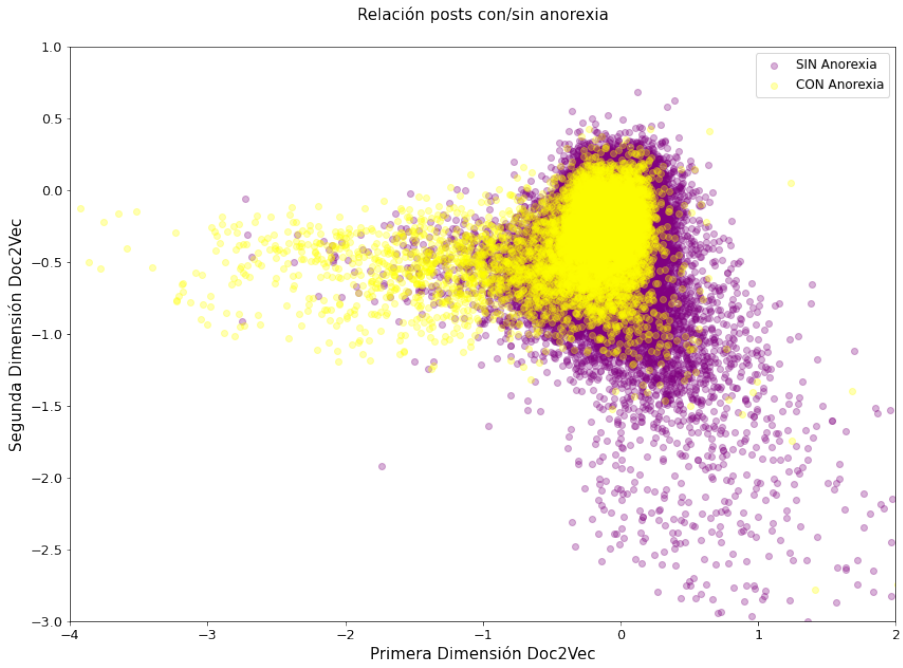

Figura 3.3: Distribución de los comentarios en las dimensiones generadas

Esta forma de obtener representaciones vectoriales se considera en general superior a otros esquemas tradicionales como *tf-idf* en *corpora* con temáticas muy variadas[5], manteniendo un nivel de complejidad razonable.

En el caso de esta prueba de concepto, se ha optado por generar una representación en únicamente **dos dimensiones** por los motivos que se harán evidentes en el apartado siguiente.

3.1.2. Generación de los *scores*

A partir de la representación generada en el paso anterior, es necesario elegir algún tipo de métrica que esté asociada a la probabilidad de que la persona sufra de anorexia.

Una manera de conseguirlo consiste atender a la **densidad de la proporción relativa** del espacio vectorial generado para cada documento, atendiendo a que, en aquellas regiones donde se acumulen *posts* de personas con anorexia en proporción superior a lo esperado (la ratio general del dataset, que en este caso sabemos que es aproximadamente 9:1, pero que en un contexto real debería ser la prevalencia de la enfermedad en la población general).

Como se puede ver en la figura 3.3, hay 3 tipos de zonas:

- Aquellas en las que predominan los comentarios de personas sin anorexia (a partir de $x < -1$ aproximadamente). Se ven ejemplos claramente característicos de personas con anorexia:

  Is there anything that I can I do that'll help me eat normally? To give a brief

[5] https://aisoftwareengineer.com/artificial-intelligence/nlp-tf-idf-vs-doc2vec-contrast-and-compare/

summary of what's happened to me. For most of high school I'd been on the thinner side..

How can I prevent a relapse? I have struggled in the past and have gone through a very hard break up.

Looking for help accepting weight 25yo guy here. About this time last year, I was in a hugely stressful job and started drinking heavily every weekend to cope

- Regiones donde sucede lo contrario, aproximadamente en $x > 1$

  Lethal Weapon - Series Premiere Discussion Premise: The 1987 movie....
  American Gods - 1x01 "The Bone Orchard" (TV Only Discussion)

- Zonas donde se superponen ambas regiones en diferentes proporciones.

Una manera de representar esta densidad de la proporción es utilizar el odds-ratio ponderado según la frecuencia base y *smoothed* (suavizado) para evitar divisiones por cero. Ésto tiene la ventaja de que las regiones vacías son equivalentes a aquellas que tienen una proporción igual a la frecuencia base y por tanto no constituyen evidencia en ningún sentido (son neutrales).

La técnica de suavizado más sencilla es el *add-n smoothing*, que consiste simplemente en añadir un número n a numerador y denominador. En este caso, se ha elegido utilizar el valor de 1, por lo que la fórmula a aplicar en una celda C del espacio vectorial V quedaría:

$$\frac{negativos}{positivos} \times \frac{1 + positivos_C}{1 + negativos_C} \tag{3.1}$$

Adicionalmente se puede aplicar el logaritmo sobre este valor, para que el resultado quede comprendido en el intervalo $[-\infty, +\infty]$, y sea más fácil visualizarlo en un mapa de calor como el de la figura 3.4.

En cuanto a las regiones del espacio delimitadas, la manera más sencilla para continuar con la prueba de concepto exploratoria en dos dimensiones es la de generar celdas rectangulares en el espacio bidimensional, aplicando en cada una de ellas la fórmula anterior. El resultado puede verse en la figura 3.4

De esta manera es posible asignar una puntuación a nuevos documentos, a partir del valor asignado a la celda en la que se encuentra.

Sería posible desarrollar otro tipo de modelos que produjesen áreas continuas sin las discontinuidades observadas, y que probablemente generalizasen mejor. No obstante, en esta fase de validación se considera suficiente la predicción por pertenencia a una celda determinada.

3.1.3. Combinación de los *scores*

Una vez generadas las puntuaciones individuales es necesario desarrollar una estrategia de combinación de las mismas para emitir una decisión definitiva.

Figura 3.4: Densidad de la frecuencia relativa normalizada $ln(\frac{negativos}{positivos} \times \frac{1+positivos_C}{1+negativos_C})$

Dado que estas puntuaciones admiten una interpretación probabilista, la manera que mejor preserva la interpretabilidad del modelo es la de adoptar un enfoque de *actualización bayesiana*[6].

Bajo este modelo, supongamos que el comentario que ocupa la posición i en la serie, está ubicado en la celda C_i:

$$Odds(Anorexia|C_i, ..., C_1) \tag{3.2}$$

$$= BayesFactor_{C_i} \times prior\, Odds \tag{3.3}$$

$$= \frac{P(C_i|Anorexia)}{P(C_i|\neg Anorexia)} \times Odds(Anorexia|C_{i-1}, ..., C_1) \tag{3.4}$$

Para calcular esta expresión tenemos que conocer $Odds(Anorexia)$ que en este *dataset* es $1:9$
En cuanto a $\frac{P(C_i|Anorexia)}{P(C_i|\neg Anorexia)}$, aplicando Bayes obtenemos:

$$\frac{P(C_i|Anorexia)}{P(C_i|\neg Anorexia)} = \frac{\frac{P(A|C_i)}{P(A)}P(C_i)}{\frac{P(\neg A|C_i)}{P(\neg A)}P(C_i)} = \frac{P(A|C_i)}{P(\neg A|C_i)}\frac{P(\neg A)}{P(A)} \tag{3.5}$$

En cuanto a $\frac{P(A|C_i)}{P(\neg A|C_i)}$ sabemos que es:

$$\frac{P(A|C_i)}{P(\neg A|C_i)} = \frac{positivos_{C_i}}{negativos_{C_i}} \tag{3.6}$$

[6]Ver por ejemplo https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2014/65200614be80c2c1efcd0f9f3db8c0e7_MIT18_05S14_Reading12b.pdf

Y en cuanto a $\frac{P(\neg A)}{P(A)}$:

$$\frac{P(\neg A)}{P(A)} = \frac{negativos}{positivos} \tag{3.7}$$

Por lo tanto el *Bayes Factor* es exactamente la expresión 3.1, **sin el *smoothing*** (que mantendremos).

$$BayesFactor_{C_i} \approx \frac{negativos}{positivos} \times \frac{1 + positivos_C}{1 + negativos_C} \tag{3.8}$$

Pero es que además, de la ecuación 3.4 se deduce fácilmente:

$$LogOdds(Anorexia|C_i, ..., C_1) \tag{3.9}$$

$$= ln(BayesFactor_{C_i}) + ln(prior\,Odds) \tag{3.10}$$

$$= \sum_{j=1}^{j=i} ln(BayesFactor_{C_j}) + ln(Odds(Anorexia)) \tag{3.11}$$

Por tanto la combinación de los *scores* es la **suma acumulativa** de las puntuaciones utilizadas en el apartado anterior.

Es importante destacar que la actualización bayesiana que estamos utilizando requiere que los eventos C_i sean **condicionalmente independientes** entre sí, lo que no se puede afirmar en este caso y es por tanto una simplificación importante que puede hacer que no sea del todo óptima.

3.1.4. Resultados y Estrategia de parada

En este punto, ya podemos implementar un clasificador que emita un resultado en cualquier momento de la secuencia de comentarios, si bien es necesario elegir un **punto de corte**, a partir del cual se emitirá un resultado positivo. Lo más sencillo es elegir un punto de corte del 50, si bien se puede ajustar según nos interese mejorar la precisión o la exhaustividad.

Podemos ver los resultados según los distintos puntos de corte evaluados sobre el conjunto de *test* de 2018:

Sorprendentemente, el $F1$ se mantiene muy estable a casi cualquier punto de corte.

En cuanto a la elección de una estrategia de parada, esta debe partir de la medición de la estabilidad en la respuesta. Una de las formas más sencillas es considerar que en cuanto el sistema ofrece el resultado positivo durante n veces seguidas, se termina la evaluación. Dado que no hay penalización por los resultados negativos, éstos nunca se interrumpen, y se sigue iterando hasta el final.

Con estos datos, y a partir de las ecuaciones de 2.2.2 se puede obtener una medida objetiva que compense la tasa de acierto con la celeridad en responder a los casos positivos de forma temprana.

El resultado para un punto de corte del $20\,\%$ tal y como se puede observar en 3.2 es que los

Cutoff	*F1*	*recall*	*precision*
10 %	0.58	0.44	0.86
20 %	0.58	0.44	0.86
30 %	0.53	0.39	0.84
40 %	0.53	0.39	0.84
50 %	0.53	0.39	0.84
60 %	0.53	0.39	0.84
70 %	0.53	0.39	0.84
80 %	0.54	0.39	0.89
90 %	0.54	0.39	0.89

Cuadro 3.1: $F1$ para distinos puntos de corte

n	$F1$	*Decisión(median)*	*speed*	$F1_{speed}$
1	0.53	4	0.99	0.52
3	0.60	7	0.98	0.59
5	0.64	9	0.97	0.62
10	0.59	14.5	0.95	0.56
20	0.55	26	0.90	0.49
50	0.49	55.5	0.79	0.39
100	0.51	105.5	0.6	0.31
200	0.37	211	0.3	0.12

Cuadro 3.2: $F1_{speed}$

mejores resultados se encuentran sobre el intervalo $[5, 10]$. Fuera de éste la penalización es mayor y el requisito de estabilidad se hace contraproducente también para los valores de F_1absolutos.

3.2. Implementación del Sistema basado en Clustering

La prueba de concepto ha demostrado que es posible implementar un sistema que siguiendo los 4 pasos planteados obtenga resultados útiles y tal vez hasta competitivos. No obstante, tiene las siguientes limitaciones:

- La capacidad de representar una gran variedad semántica viene limitada por el bajo número de dimensiones utilizadas.
- Aumentar las dimensiones a más de 3 imposibilita el utilizar el concepto de celdas (al menos de una manera sencilla). Además, la densidad en espacios hiperdimensionales se reduce de forma muy rápida.
- Requiere de cierto *tuning* manual, ya que los límites y tamaño y configuración de las celdas se eligen de forma manual por observación.

- Existen discontinuidades bruscas entre celdas contiguas que son producto del azar; bajo distintas configuraciones puntos próximos pueden tener valores muy distintos.

- No ofrece ningún tipo de información sobre por qué se asigna una puntuación u otra, lo que limita su interpretabilidad.

Por todos estos motivos, se plantea una iteración del modelo que cumpla con las siguientes características:

1. En cuanto al modelo vectorial se seguirá utilizando *doc2vec*, si bien se elegirá un mayor número de dimensiones para capturar mayor variabilidad semántica.

2. En lugar de un modelo de agrupación basado en celdas, se utilizará la asignación de *posts* en *clusters*. Por similitud conceptual, los algoritmos preferentes deberían ser aquellos que buscan zonas de densidad elevada de puntos como *DBSCAN* y sus derivados (*Hierarchichal DBSCAN*, etc...).

   Este tipo de clustering tiene la ventaja de que es posible identificar los temas a los que corresponde cada *cluster* (por ejemplo extrayendo las palabras o conceptos más distintivos), de forma que podemos extraer conocimiento.

   En cuanto a la determinación del *scoring*, es igualmente aplicable la deducción de las ecuación 3.4 y siguientes, sustituyendo la asignación a las celdas por la asignación al *cluster*.

3. La combinación de las puntuaciones individuales y el criterio de parada se mantienen inalterados.

Las primeras aproximaciones a este enfoque, con vectores de 100 dimensiones, daban como resultado que la mayoría de los documentos se clasificaran como ruido, mientras el resto eran asignados a una decena de clústeres, claramente insuficiente para el objetivo perseguido. Con *KMeans* los resultados eran algo mejores, pero aún insuficientes para capturar la variabilidad semántica necesaria.

El problema de la hiperdimensionalidad se solucionó a partir de lo expuesto en el artículo Angelov (2020). No sólo éso sino que se sustituyó la implementación propia por la ofrecida por la librería de *Github*[7] que acompaña a la publicación, dado que soluciona de forma más eficiente la generación del espacio vectorial, el clustering, y la identificación de los clusters con conceptos o temas de forma que puedan ser interpretados fácilmente.

Dado que es un componente fundamental en la solución, merece la pena revisar los pasos del algoritmo[8].

[7] https://github.com/ddangelov/Top2Vec

[8] **Todas** la figuras utilizadas en este apartado se han sacado de la página de *Github* y son obra de su autor, Dimo Angelov

Top2Vec

1. **Creación de la representación vectorial de los documentos.** Este paso es casi inevitable en cualquier aproximación de clasificación de textos. Para este caso, los algoritmos soportados para la transformación de los documentos, además de *doc2vec*, son *Universal Sentence Encoder* y *BERT Sentence Encoder*. Sin embargo, hay un matiz importante y es que el espacio vectorial, **es común para palabras y documentos**.

   De la descripción de los algoritmos *doc2vec* y *word2vec* realizada anteriormente no se deduce inmediatamente que ésto sea así. La justificación en el artículo parte del funcionamiento de la variante *Skip-Gram* de *word2vec*, que deduce el contexto a partir de la palabra central de la ventana, y de la variante *Distributed Bag Of Words(DBOW)* de *doc2vec*, que es muy similar sólo que la entrada en lugar de ser un *one-hot vector* del vocabulario es el id del documento.

   De esta similitud se arguye en el artículo que el espacio semántico generado por *word2vec* y *doc2vec* constituye **una representación continua de temas**. La matriz generada por *word2vec* $W'_{n,d}$ que contiene vectores contexto de dimensión d para las n palabras del vocabulario es una transformación lineal que aplicada a un vector de dimensión d genera un vector de dimensión n. Este vector es una medida de la importancia de las n palabras en el vector documento d y por tanto, el documento p puede identificarse con aquellas palabras que maximizan $softmax(\vec{p} \cdot W'_{n,d})$

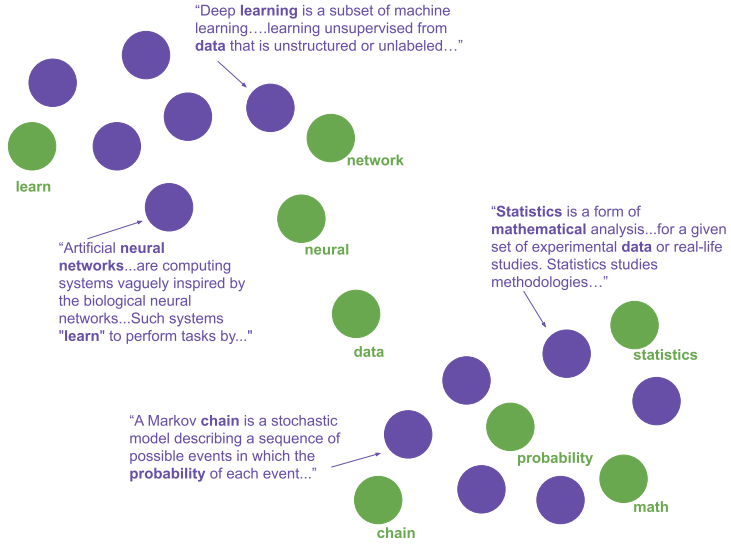

Figura 3.5: Generación de la representación vectorial conjunta palabras-documentos. Angelov (2020)

2. **Reducción de la dimensionalidad utilizando el algoritmo *UMAP*.** Este paso es el que estaba ausente en el planteamiento inicial, dado que no resulta evidente cual es la diferencia entre generar una representación vectorial de mayor dimensionalidad y luego aplicar un

algoritmo para reducirla en contraposición de simplemente elegir un tamaño menor para los *embeddings*. El artículo no aclara explícitamente esta circunstancia si bien hace referencia a que este algoritmo es capaz de preservar tanto la estructura global como la local, al igual que otros algoritmos de clustering como *t-SNE* (si bien con un comportamiento computacional mejor). Es posible que hacerlo así ayude a preservar de alguna manera los "matices" semánticos codificados en un espacio vectorial superior, si bien no se puede ofrecer una explicación matemática o empírica que lo sustente.

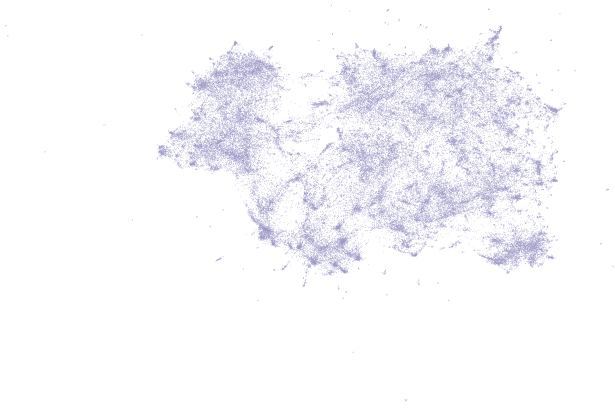

Figura 3.6: Reducción de la dimensionalidad. Angelov (2020)

3. **Identificación de las áreas densas de documentos usando *HDBSCAN*.**

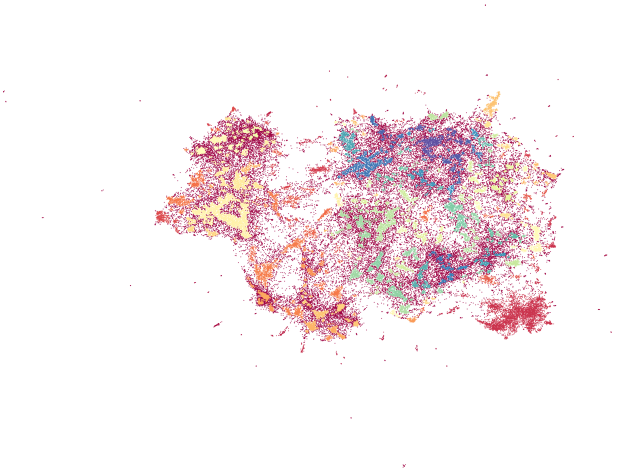

Figura 3.7: Clustering con HDBSCAN. Angelov (2020)

4. **Identificación del *vector-tema* o *topic vector*.** Para cada área densa, se calcula el centroide en la dimensión original, que se denomina vector-tema.

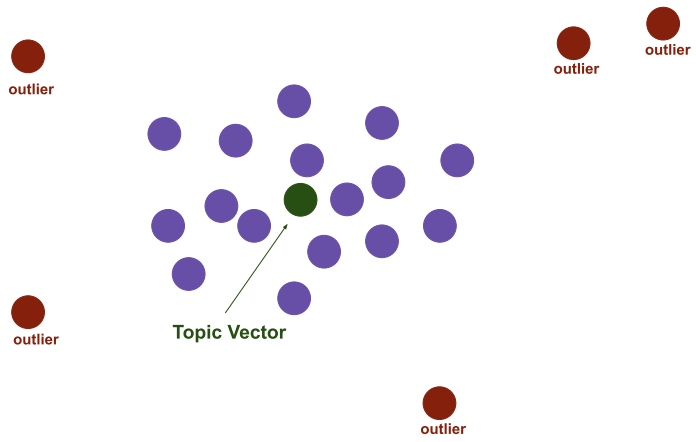

Figura 3.8: Identificación *topic vector.* Angelov (2020)

5. **Identificación de los *embeddings de palabras* más cercanos del vector tema.** Es importante destacar que se refiere a las **palabras y no a los documentos**, y que puede hacerse gracias a que se representan en un espacio vectorial conjunto. De esta manera es posible "describir" hasta cierto punto la temática de un cluster a partir de estas palabras más similares al vector tema.

   La alternativa que se planteaba para realizar este paso de no haber existido esta librería pasaba por obtener las palabras más frecuentes de las encontradas en los documentos de un *cluster* determinado. Dado que aquí se ha definido un espacio semántico compartido por palabras y documentos, no es necesario realizar este paso y es un enfoque mucho más eficaz, dado que se trabaja ya con representaciones vectoriales de las palabras que representan su significado.

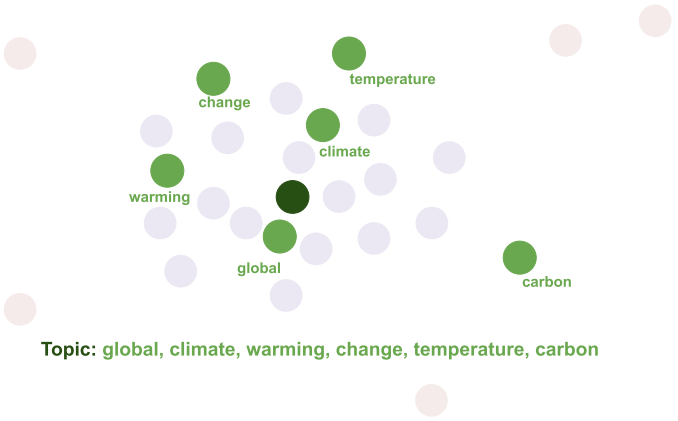

Figura 3.9: Identificación de palabras clave de un tema. Angelov (2020)

3.3. Evaluación de los resultados sobre *dataset* 2019

Con esta librería, es sencillo generar un proceso que realice la clasificación de acuerdo a los pasos especificados. En concreto, basta con:

1. Generar un modelo *Top2Vec* que identifica los distintos temas a partir del *dataset* de 2018 y asigna los documentos a cada uno de ellos
2. Se calcula el *score* de cada tema o cluster a partir de las instancias positivas y negativas
3. Cada nuevo comentario, se asigna a un cluster y se le asigna una puntuación de acuerdo al tema al que pertenece.
4. Se combinan de la manera descrita anteriormente, aplicando la estrategia de parada temprana.

Los parámetros utilizados de corte y valores consecutivos sobre el mismo son los identificados como mejores durante la prueba de concepto($n = 5$ y $cutoff = 20$). Los resultados finales son los del cuadro siguiente:

Dataset	$F1$	*precision*	*recall*	*Decisión (median)*	*speed*	$F1_{speed}$
2018 (training)	0.84	0.81	0.86	11	0.96	0.81
2019 (test)	0.64	0.66	0.62	16.5	0.94	0.60

Cuadro 3.3: Resultados del Algoritmo

3.4. Comparativa

Tomando como referencia los resultados de la edición de 2019 mostrados en el cuadro 2.2, el algoritmo presentado en en el capítulo anterior lograría las posiciones que se recogen en el cuadro 3.4 para cada una de las métricas que se consideran de interés: precisión, exhaustividad, F1, velocidad y latencia. Se excluye el cálculo de los ERDE por los múltiples problemas identificados con esta métrica.

Esta comparativa es puramente teórica y en ningún caso afirma que se hubiera conseguido esta posición de haber competido en ese momento. Las restricciones de tiempo y de desarrollo no son comparables. Por ejemplo, no ha sido necesario desarrollar un cliente específico que emitiese las recomendaciones individuales, lo que si tuvieron que implementar los equipos competidores.

Por otro lado los algoritmos y técnicas empleadas en este TFM ya existían en 2019 por lo que hipotéticamente podría haberse presentado una solución basada en los mismos componentes.

Lo que se deduce del cuadro es que la solución es competitiva en precisión, F1 y F1 ajustada, siendo tan sólo "mediano" en velocidad y exhaustividad. Ésto se puede ver de forma más explícita

[9]La mejor tiene un valor de 1, pero realmente con un valor de F1 de 0,03 lo que la hace poco interesante. Se añade la segunda mejor que es más comparable.

	TFM	***Mejor***	***Ranking***	
			Algoritmo	***Equipo***
(P)recision	0.66	1 (0.77)[9]	8/51	4/14
(R)ecall	0.62	1 (0.92)	30/51	10/14
F1	0.64	0.71	13/51	4/14
speed	16,5	1	35/51	11/14
latency F1	0,6	0.69	10/51	4/14

Cuadro 3.4: Comparativa de la propuesta con las soluciones de 2019.

en el cuadro 3.5 donde se muestran únicamente los mejores resultados logrados por cada equipo. Más allá del ganador claro, que tiene 6 percentiles de ventaja sobre los siguientes, hay un segundo escalón de 4 equipos (incluyendo "virtualmente" a este TFM) que logran resultados muy similares y cuyo orden definitivo probablemente sea resultado de las particularidades concretas de este *dataset*.

team	P	R	F1	latencyTP	speed	latency-weighted F1
CLaC	.64	.79	.71	7	.98	.69
lirmm	.74	.63	.68	21	.92	.63
INAOE-CIMAT	.67	.68	.68	20	.93	.63
TFM Actual	**.66**	**.62**	**.64**	**16.5**	**.94**	**.60**
UDE	.51	.74	.61	11	.96	.58
UNSL	.42	.78	.55	2	1	.55
LTL-INAOE	.47	.75	.58	11	.96	.55
BiTeM	.44	.70	.54	3	.99	.54
UppsalaNLP	.40	.42	.41	1	1	.41
BioInfo@UAVR	.32	.44	.37	1	1	.37
SSN-NLP	.48	.26	.34	2	1	.33
SINAI	.18	.95	.30	8	.97	.30
HULAT	.11	.30	.17	16.5	.94	.16
Fazl	.09	1	.16	34	.87	.14

Cuadro 3.5: Comparación mejores resultados de cada equipo incluyendo la solución propuesta en este TFM.

En la velocidad se ha sido probablemente demasiado conservador en la elección del criterio de parada con 5 comentarios consecutivos seguidos necesarios para detener el procesamiento. Un valor de 3 hubiese sido con alta probabilidad una mejor elección. La exhaustividad en cambio podría mejorarse probablemente sólo a costa de la precisión.

Los 7 percentiles de diferencia en F1 y 9 en F1 ajustados son importantes, pero hay que tener en cuenta que la solución ganadora, CLaC, es computacionalmente muy costosa, y seguramente menos interpretable.

Capítulo 4

Conclusiones y líneas de trabajo futuras

4.1. Interpretabilidad e inferencia del modelo desarrollado

Antes de analizar estos puntos, resulta necesario definir de manera precisa su significado en este contexto. En Molnar (2022), de entre varias definiciones alternativas propuestas para la interpretabilidad, se ha elegido ésta por su claridad:

> Interpretabilidad es el grado en el que un humano puede consistentemente predecir los resultados del modelo: a mayor interpretabilidad de un modelo de aprendizaje automático, más fácil es para alguien el entender por qué se han realizado las predicciones.

Por tanto, la interpretabilidad surge del análisis de **predicciones concretas.** Lo que en el ejercicio del primer capítulo se justificaba desde un razonamiento lógico (e.g: el sujeto en el comentario #32 pide información sobre laxantes lo que es un rasgo típico de un trastorno alimentario), debería tener su correspondencia en un modelo interpretable. Es decir, el sistema debe decirnos:

- Qué comentarios han "inclinado la balanza" hacia la predicción (positiva o negativa).
- Por qué esos comentarios son determinantes (qué rasgos los distinguen).

Un sistema que es interpretable proporciona confianza a sus usuarios porque sus predicciones vienen acompañadas de una justificación. En algunos casos puede ser incluso tan importante como la propia predicción.

Otro aspecto que mejora también la confianza en un sistema por parte de sus usuarios es que éste pueda ser descrito de forma sencilla y sin que sea necesario recurrir a complejas expresiones matemáticas. Una descripción que no requiere de conocimientos especiales podría ser la siguiente:

> El modelo asigna cada uno de los comentarios a unas categorías precalculadas de acuerdo a criterios de similitud semántica (e.g: tratan de temas similares, utilizan el mismo lenguaje, etcétera). Cada una de esas categorías tiene asignada una puntuación

		Previsión		Parada	
Username	Valor verdadero	Manual	TFM	Manual	TFM
TRsubject195	0	0	0	-	-
TRsubject1637	1	0	0	-	-
TRsubject1913	1	**1**	**0**	27/44	-
TRsubject5127	1	1	1	5/416	22/416
TRsubject3614	0	0	0	-	-
TRsubject9229	1	**0**	**1**	-	50/1224
TRsubject1604	0	0	0	-	-
TRsubject3339	0	0	0	-	-
TRsubject3132	1	**0**	**1**	-	144/259
TRsubject1496	0	0	0	-	-

Cuadro 4.1: Resultados evaluación automática vs manual

derivada de la proporción de personas que tienen anorexia en los datos históricos de los que se han extraído también las categorías. Esta puntuación puede ser combinada con las de los comentarios anteriores mediante una técnica conocida como actualización bayesiana. El resultado de esa combinación es la probabilidad de que una persona concreta padezca anorexia. En el caso de que esta puntuación se mantenga elevada durante un tiempo (varios comentarios consecutivos), se asumirá que esta persona está en riesgo de padecer esta enfermedad u otra del mismo espectro.

En cuanto al concepto de inferencia, a menudo se suele utilizar para describir la realización de una predicción por parte de un modelo a partir una nueva observación. No es el sentido con el que se emplea aquí. La inferencia en un sentido más amplio, incluye establecer nuevas hipótesis y relaciones a partir del análisis de datos. Por tanto a partir de los temas identificados y su proporción de positivos/negativos se estudiará su grado de correspondencia con las conductas extraídas de la literatura en Salud Mental, y si es posible establecer paralelismos o por el contrario hay que manifestar la existencia de contradicciones.

4.1.1. Análisis de la interpretabilidad

En el primer capítulo, se clasificaron "manualmente" una serie de usuarios seleccionados al azar. Al ejecutar el algoritmo sobre ellos podemos comparar sus predicciones con los resultados que se asignaron entonces. La comparativa se puede ver en el cuadro 4.1, en el que se marcan en negrita los casos en los que difieren las predicciones.

Por otro lado la evolución de la suma acumulativa nos ayudará a examinar las tendencias de cada usuario. Se muestra en la figura 4.1 la puntuación acumulada en cada momento para los 5 positivos y los 5 negativos respectivamente.

De estas comparativas, numérica y visualización, se extraen las siguiente conclusiones:

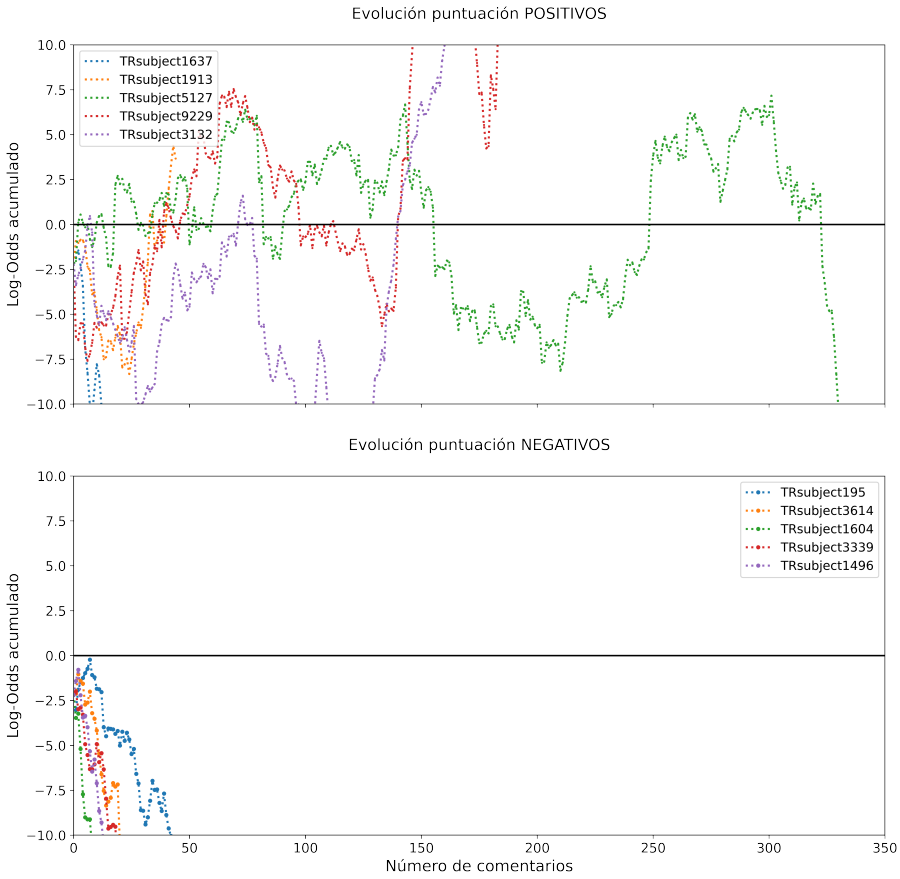

Figura 4.1: Suma acumulativa para los positivos (tp+fn) y negativos (fp+tn)

- En los negativos el algoritmo funciona tan bien como la evaluación manual. Además, éstos inmediatamente se alejan de la línea de corte y no vuelven a acercarse. Los resultados negativos de esta muestra resultan claramente negativos para un observador.

- En los positivos, el resultado es mejor que el de la evaluación manual (3/5 vs 2/5). En el positivo que no identifica el algoritmo y sí lo hace la persona (TRSubject1913) se da la circunstancia de que el usuario sólo tiene 44 comentarios en total. Un número tan reducido hace complicado que la puntuación se estabilice. Incluso, el último comentario se sitúa sobre la línea de corte, en la región de los positivos, pero no llega a tener 5 comentarios consecutivos en dicha zona. El otro positivo no identificado sin embargo, presenta un patrón muy alejado del resto.

  Aunque es de esperar que un clasificador humano mejorase rápidamente en sus evaluaciones y acabase superando al algoritmo con claridad, es relevante que los resultados sean comparables a los de una persona no entrenada.

  También abre una posibilidad interesante y es que en lugar de que las evaluaciones sean las obtenidas por el algoritmo, una persona se apoye en visualizaciones como las de la figura 4.1 para seleccionar los casos que presenten patrones "dudosos". En este caso el sistema actuaría como un componente de apoyo a la decisión, reduciendo la intervención humana sin eliminarla.

Username	topic #	pos #	topic words	Explicación
TRsubject3132	20	134	squats, lifting, cardio, squat, workout	Ejercicio
	321	42	depression, anxiety, bipolar, therapy, therapist	Enfermedad mental
	3210	50	dubai, diesel, golf, fountain, exchange	*Overfiting*
TRsubject5127	9	17	bipolar, therapist, psychiatrist, anxiety, therapy	Enfermedad mental
	3703	1	infection, bacteria, flu, pain, nausea	Enfermedad
	1850	19	anthony, opie, cumia, tacs, jim	Programa de radio
TRsubject9229	1064	38	ranger, ford, vehicle, car, cars	Coches
	2586	32	lift, squats,weights,lifting, cardio	Ejercicio
	2586	36	lift, squats, weights, lifting, cardio	Ejercicio

Cuadro 4.2: Predicciones positivas: comentarios destacados

No obstante, requeriría cierto entrenamiento ya que una persona no adiestrada probablemente no sabría interpretarla, o bien sustituirla por una donde las tendencias sean más perceptibles.

- Los puntos de decisión son difíciles de comparar porque sólo son relevantes en los positivos (en los negativos no penalizan) y ambas evaluaciones sólo coinciden en sus respuestas de verdaderos positivos en uno de los casos. Aún así ambas evaluaciones son tempranas, emitiendo resultado antes de procesar siquiera el 5 % de los comentarios.

Para que la interpretabilidad sea más completa resulta fundamental poder identificar los comentarios que han decantado el resultado en ese sentido y el motivo. Para ello, se realiza en las predicciones positivas los 3 comentarios con mayor puntuación anteriores a la toma de la decisión, que se recogen en el cuadro 4.2.

En dicho cuadro se puede ver la identificación de determinados comentarios con temas que son típicos de la anorexia (ejercicio, ansiedad y depresión, enfermedad, etc...). En otros por el contrario la relación es difícil de establecer, como en el caso en el que se hace referencia a un popular programa de radio [1] o simplemente sobre determinados modelos de coches. También, dado que los usuarios evaluados han sido extraídos de los propios datos de entrenamiento se puede observar lo que es un probable caso de *overfitting*. Los términos "dubai, diesel, golf" probablemente sólo son comunes en la producción de ése usuario en concreto.

De este análisis básico se concluye que es posible entender, en términos generales, los motivos tras las "decisiones" del algoritmo, e incluso identificar en qué comentarios están ofreciendo resultados que no son del todo fiables (*overfitting*, temáticas *a priori* generalistas, etc...).

4.1.2. Inferencia

Los clusters generados y sus puntuaciones resultan de interés desde el punto de vista del comportamiento de las personas con anorexia. Resulta lógico pensar que las temáticas sobre las que versan

[1] Este programa tiene un episodio dedicado a los sitios pro-anorexia que podría explicar esta relación https://www.youtube.com/watch?v=EnL4l_lKeeU

sus comentarios en las redes sociales coinciden con las conductas típicas: obsesión por la comida, realización de ejercicio intenso, ansiedad, etcétera. Si ésto es así, servirá de confirmación acerca de la calidad del algoritmo.

Por otro lado, pueden existir temas cuya relación con la enfermedad sea menos evidente y su análisis pueda revelar alguna clave que merezca la pena explorar. En el caso anterior hemos visto un ejemplo, un programa de radio *a priori* generalista. Y también una posible explicación; este programa tiene un capítulo dedicado a las páginas web que promueven la delgadez extrema y animan a seguir con los ayunos hasta la hospitalización e incluso la muerte.

Desde luego no será información novedosa para los profesionales que trabajen en este ámbito, pero el hecho de encontrar una relación también incrementa el grado de confianza en la solución.

Tampoco los manuales de salud mental, por razones evidentes, pueden recoger que temáticas son típicas de las personas que **no** sufren trastornos de la alimentación. En este análisis por el contrario se pueden buscar temas donde estén sobrerrepresentados los negativos, e intentar también conjeturar una relación causal.

El punto de partida sería la visualización de la figura 4.2. El *odds ratio (or)* ajustado por la frecuencia "basal" nos indica cómo de sobrerrepresentados están los positivos respecto a los negativos o viceversa. Por ejemplo, el primer cluster nos indica que hay 200 veces más positivos que negativos de lo que es normal en la muestra.

En lugar de usar una única métrica para toda la gráfica se ha decidido que cuando la relación se invierte, es decir los positivos están infrarrepresentados respecto a la frecuencia "basal", la métrica se sustituye por el *odd ratio* contrario: negativos respecto a positivos. Esta representación tiene el problema de que se produce una discontinuidad del $+1$ al -1, siendo que en realidad esos clusters están muy próximos en cuanto a *odds ratio*. Sin embargo, para la visualización esta discontinuidad es apenas perceptible y en cambio aporta mejor información.

En la figura 4.2 se puede ver que hay un pequeño número de clusters con *or* significativamente positiva y que estarían asociados con la anorexia. Una gran mayoría de clusters sin embargo serían "neutrales" y un número significativo, en torno a 500, que serían evidencia de que alguien **no** padece la enfermedad, si bien, tal y como se ha marcado en el gráfico realmente corresponden a grupos donde no hay ninguna instancia positiva. Es relevante destacarlo dado que se puede plantear el uso de alguna técnica de *smoothing,* alternativa al *add-one* empleado aquí, para extrapolar el *odds ratio* de estos clusters a partir de una expresión que produzca valores menos extremos.

De la figura 4.3 se deduce rápidamente por qué tienen una proporción de positivos tan amplia. Las temáticas a las que hacen referencia son:

- Atracones, restricción de comida, purgas, recaída, etc... todo conductas relacionadas con los desórdenes alimentarios.

- Ayuno, calorías, etc... que si bien no son tan extremos como en el caso anterior, coinciden también con conductas típicas.

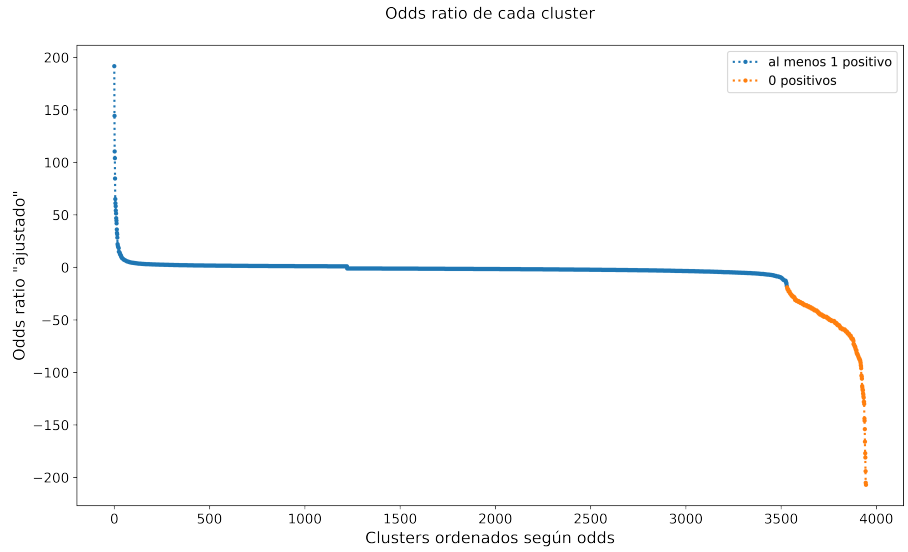

Figura 4.2: Visualización odds-ratio ajustado. Valores positivos muestran *or* pos : neg. Valores negativos *or* neg : pos

- Ingreso, terapia, psiquiatra... claramente relacionadas con la atención en salud mental.
- Desórdenes alimentario, terapia, recuperación... aquí falta un tema central más claro, pero también se advierte la relación.

Con estos ejemplos confirmamos las suposiciones iniciales. ¿Qué ocurre en el otro extremo de la gráfica? Se puede ver en la figura 4.4.

En este caso, sin embargo, en este caso las relaciones no son en absoluto evidentes. Hay un cluster presumiblemente relacionado con la lucha contra el *spam*, otro sin un tema central identificable, otro sobre impresión digital y uno último sobre *Dragon Ball*. Resulta aventurado intentar encontrar ninguna lógica detrás de estos resultados y es probable que estos resultados sean puramente aleatorios. Es por tanto una posible mejora del sistema el reducir la influencia de estos clústeres sobre la predicción final.

4.2. Líneas de trabajo posteriores

A partir de la implementación propuesta y sus conclusiones, se abren distintas vías que podrían explorarse para mejorar los resultados del algoritmo.

Exploración de hiperparámetros

Los hiperparámetros elegidos en los distintas etapas, como la puntuación de corte o *cutoff* o el número de dimensiones de *doc2vec,* han sido bien las que se han extraído de la prueba de concepto

Figura 4.3: "Wordclouds" de los 4 clusters con mayor proporción de positivos.

o bien aquellas que las librerías tienen definidas como parámetros por defecto y que suelen estar optimizadas para los casos de uso más generales.

Una revisión sistemática de los hiperparámetros de los algoritmos que intervienen (*doc2vec*, *hdbscan*, etc...) podría potencialmente mejorar los resultados generales del sistema de forma significativa. No obstante, dado que esta exploración es computacionalmente intensiva se ha preferido no explorar de forma completa esa vía y dedicar el tiempo que requeriría a intentar extraer conclusiones sobre el comportamiento de la solución.

Otros métodos de combinación de *scores* y criterio de parada

El método de combinación de puntuaciones expuesto en la subsección 3.1.3 no es otra cosa que la aplicación de la actualización bayesiana y por tanto resulta fácilmente defendible. Este mecanismo permite combinar diferentes observaciones e ir actualizando la probabilidad asignada de forma acorde, dependiendo tan sólo de la creencia actual (*prior*) y de la puntuación de la nueva evidencia.

Sin embargo, parte de una asunción de independencia condicional que en el caso del dataset es difícilmente asumible. Según ésta, el *cluster* al que pertenece el siguiente comentario sería independiente de los anteriores, siendo determinado únicamente por la condición de positivo o negativo del

Figura 4.4: "Wordclouds" de los 4 clusters con mayor proporción de negativos.

usuario.

La independencia condicional implica que $P(Cluster_j|Cluster_i, Anorexia) = P(Cluster_j|Anorexia)$. Sin embargo, ésto es difícil de sostener, un usuario concreto tenderá a tener varios comentarios en un mismo cluster (o uno próximo), y por tanto esta simplificación, a pesar de ser útil, no se cumple en la práctica.

Ésto invita a explorar otras manera de combinar los *scores*. De la revisión de las aproximaciones de los distintos equipos destaca la elección de redes *LSTM* como método de combinación más común. Este tipo de solución podría potencialmente, si se mantiene el identificador de *cluster* como una de las variables de entrada, detectar patrones secuenciales complejos. Por ejemplo, la aparición de varios comentarios en un mismo cluster podría tener un cambio similar a que apareciese un sola vez (entendiendo que la repetición no aporta información nueva), o que la combinación de secuencias de atracón-remordimiento consecutivas tienen una interpretación distinta que a la inversa (mientras la actualización bayesiana es indiferente al orden).

También merece la pena explorar con distintos criterios de parada alternativos al elegido que se basa en el mantenimiento del criterio positivo durante un número consecutivo de comentarios. En la figura 4.1, se puede observar como los patrones de la serie temporal son muy diferentes en los

positivos que en los negativos. Sin embargo los positivos cruzan varias veces la línea de corte, lo que podría interpretarse como un resultado todavía dudoso.

La aplicación del análisis en el que se tiene en cuenta el número de comentarios procesados tiene **altas probabilidades** de mejorar significativamente los resultados. Por ejemplo, un *log-odds* de -10 en el comentario 50, tiene una interpretación muy distinta en el comentario 200 y es un hecho que no se está teniendo en cuenta en la implementación desarrollada.

Pertenencia "parcial" a varios clusters

Uno de los problemas de la prueba de concepto a los que se alude en la sección 3.2 es que la pertenencia a una celda concreta, respecto a una adyacente puede arrojar resultados muy diferentes. Sin embargo en la implementación basada en clústeres no se evitan estas discontinuidades y comentarios cercanos pueden acabar con puntuaciones muy diferentes.

Dado el alto número de temas identificados, el problema es probablemente menor que en la prueba de concepto. Sin embargo es posible reducir el impacto de forma completa dado que *Top2vec* permite obtener probabilidades de pertenencia a cada *cluster*. En este caso, se ha realizado la asignación atendiendo al criterio de la mayor probabilidad, pero sería posible asignar a cada comentario **una puntuación ponderada** de acuerdo a su probabilidad de pertenencia a cada uno de los temas existentes.

Nuevas fuentes de datos (overfit)

El primer paso del algoritmo es la generación de los *embeddings*, con *doc2vec* u otro algoritmo similar. Este paso se ha realizado a partir del *dataset* proporcionado, pero es posible, o bien partir de un modelo preentrenado como hacía alguno de los participantes, o bien obteniendo una mayor cantidad de datos con comentarios de muchos más *subreddits* y que capturen más campos semánticos.

Ésto ayudaría a eliminar el probable *overfit* identificado en la subsección 4.1.2, ya que con un número más elevado de datos es más difícil que los clústeres acaben identificando a usuarios concretos.

Bibliografía

American Psychiatric Association (2013). *Diagnostic and statistical manual of mental disorders: DSM-5*. Autor, Washington, DC, 5th ed. edition.

Angelov, D. (2020). Top2vec: Distributed representations of topics. *CoRR*, abs/2008.09470.

Aragón, M. E., López-Monroy, A. P., and y Gómez, M. M. (2019). Inaoe-cimat at erisk 2019:detecting signs of anorexia using fine-grained emotions.

Burdisso, S. G., Errecalde, M., and Montes-y-Gómez, M. (2019a). A text classification framework for simple and effective early depression detection over social media streams. *CoRR*, abs/1905.08772.

Burdisso, S. G., Errecalde, M., and y Gomez, M. M. (2019b). Unsl at erisk 2019: a unified approach for anorexia, self-harm and depression detection in social media.

Díaz-Marsá, M., Luis, J., and Sáiz, J. (2000). A study of temperament and personality in anorexia and bulimia nervosa. *Journal of Personality Disorders*, 14(4):352–359.

Duffy, M. E., Rogers, M. L., Joiner, T. E., Bergen, A. W., Berrettini, W., Bulik, C. M., Brandt, H., Crawford, S., Crow, S., Fichter, M., Halmi, K., Kaplan, A. S., Klump, K. L., Lilenfeld, L., Magistretti, P. J., Mitchell, J., Schork, N. J., Strober, M., Thornton, L. M., Treasure, J., Woodside, B., Kaye, W. H., and Keel, P. K. (2019). An investigation of indirect effects of personality features on anorexia nervosa severity through interoceptive dysfunction in individuals with lifetime anorexia nervosa diagnoses. *International Journal of Eating Disorders*, 52(2):200–205.

Elena Fano1, J. K. and Nivre1, J. (2019). Uppsala university and gavagai at cleferisk: Comparing word embedding models.

Elham Mohammadi, H. A. and Kosseim, L. (2019). Quick and (maybe not so) easy detection of anorexia in social media posts.

Heinzerling, B. (2019). Nlp's clever hans moment has arrived. *The Gradient*.

Howard, J. and Ruder, S. (2018). Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Losada, D. E. and Crestani, F. (2016). A test collection for research on depression and language use. In *Lecture Notes in Computer Science*, pages 28–39. Springer International Publishing.

Losada, D. E., Crestani, F., and Parapar, J. (2018). Overview of eRisk: Early risk prediction on the internet. In *Lecture Notes in Computer Science*, pages 343–361. Springer International Publishing.

Losada, D. E., Crestani, F., and Parapar, J. (2019). Overview of eRisk 2019 early risk prediction on the internet. In *Lecture Notes in Computer Science*, pages 340–357. Springer International Publishing.

Masood, R., Hu, M., Fabregat, H., Aker, A., and Fuhr, N. (2020). Anorexia topical trends in self-declared reddit users. In Cantador, I., Chevalier, M., Melucci, M., and Mothe, J., editors, *Proceedings of the First Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020*, volume 2621 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *CoRR*, abs/1308.6297.

Molnar, C. (2022). *Interpretable Machine Learning*. 2 edition.

Naderi, N., Gobeil, J., Teodoro, D., Pasche, E., and Ruch, P. (2019). A baseline approach for early detection ofsigns of anorexia and self-harm in reddit posts.

Ortega-Mendoza, R. M., Farias, D. I. H., and Montes-Y-Gomez, M. (2019). Ltl-inaoe's participation at erisk 2019: Detecting anorexia in social media through shared personal information.

Ragheb, W., Azé, J., Bringay, S., and Servajean, M. (2019). Attentive multi-stage learning for early risk detection of signs of anorexia and self-harm on social media.

Razan Masood, Faneva Ramiandrisoa, A. A. (2019). Ude at erisk 2019: Early risk prediction on the internet.

Sociedad Española Médicos Generales y de Familia (2018). Los trastornos de la conducta alimentaria son la tercera enfermedad crónica más frecuente entre adolescentes. `https://www.semg.es/images/stories/recursos/2018/agenda_actividades/nota_prensa_20181130.pdf`. [Online; accessed 20-February-2021].

Vashisth, P. and Meehan, K. (2020). Gender classification using twitter text data. pages 1–6.

Anexo A

Herramientas utilizadas

Como lenguaje de programación, *Python* es prácticamente la única alternativa que ofrece garantías. Aunque existen implementaciones de procesamiento natural en diversos lenguajes, lo cierto es que todas las soluciones analizadas en la sección 2.3 han utilizado este lenguaje.

Las librerías empleadas han sido:

- **Gensim** para la generación de *embeddings* con *doc2vec,* dado que tiene una implementación muy completa.
- **Top2vec**, ya mencionada varias veces en esta memoria, que es una solución de topic modeling muy adaptada al problema.
- **Pandas** que es la solución más empleada para la manipulación de datos tabulares (*data frames* como los nativos en el lenguaje de programación *R*). A pesar de que han surgido recientemente algunas alternativas más eficientes como *Polar.rs*, lo cierto es que para el volumen de datos manejado la diferencia no es significativa y el hecho de haberla utilizado antes durante muchos años ha decantado la balanza a su favor.
- **Matplotlib** para la generación de gráficos. De nuevo, existen alternativas más modernas en este espacio pero matplotlib a pesar de su API algo compleja.
- **Sklearn** para el cálculo de algunas métricas y exploración de varios algoritmos.
- **Jupyter Lab** para la codificación de las soluciones, tanto en formato *notebook* para las partes más interactivas como para los ficheros *.py* para en aquellos que estaban orientados a el procesamiento desatendido.

Como *datastore* se ha elegido *SQLite*. Los datos proporcionados consistían de ficheros *XML* individuales, lo que hacía algo engorroso su manipulación y sobre todo su exploración. El hecho de almacenarlos en un formato tabular simplifica todo esta fase y permite realizar muchas manipulaciones de forma declarativa de forma más sencilla que con *pandas*. *DuckDB* hubiese sido incluso

una mejor solución a tener en cuenta, por su mejor soporte de SQL y por su orientación a tareas analíticas (OLAP) frente a transaccionales (OLTP).