COMILLAS
UNIVERSIDAD PONTIFICIA

Facultad de Ciencias Económicas y Empresariales

Análisis de Percepciones y Sentimientos sobre la Concepción del Amor a través de Reddit

Autor: Leire Reneses Rodríguez
Director: Jenny Alexandra Cifuentes Quintero

MADRID | Junio 2025

Resumen

Este Trabajo de Fin de Grado explora cómo se expresa y percibe el amor romántico en redes sociales a lo largo de las estaciones del año, utilizando como fuente principal publicaciones de Reddit. A través de técnicas de procesamiento de lenguaje natural y modelos de inteligencia artificial como BETO, se analizó una base de datos de más de 5000 textos que mencionan directa o indirectamente el amor. La metodología se estructuró en tres fases: análisis de n-gramas, modelado de tópicos y análisis de sentimiento. A través del análisis de n-gramas y el modelado de tópicos se ha hallado las conversaciones más frecuentes y se han podido esquematizar las emociones estacionales. El análisis de sentimiento, por otra parte, es predominado por sentimientos neutros y negativos. Se identificó una estructura del discurso amoroso que sugiere un amor más introspectivo que eufórico. El estudio evidencia que el amor no es una experiencia estática ni homogénea, sino que varía según el tiempo, el lenguaje y el contexto social.

Palabras Clave: Amor, análisis de sentimiento, Reddit, procesamiento de lenguaje natural, estacionalidad emocional, modelo BETO, emociones, redes sociales, lenguaje digital, inteligencia artificial, percepción afectiva.

Abstract

This thesis explores how romantic love is expressed and perceived on social media throughout the seasons, using Reddit posts as its primary source. Using natural language processing techniques and artificial intelligence models such as BETO, a database of more than 5,000 texts that directly or indirectly mention love was analyzed. The methodology was structured in three phases: n-gram analysis, topic modeling, and sentiment analysis. Through n-gram analysis and topic modeling, the most frequent conversations were identified and seasonal emotions were schematized. Sentiment analysis, on the other hand, is dominated by neutral and negative sentiments. A structure of love discourse was identified that suggests a more introspective than euphoric love. The study shows that love is not a static or homogeneous experience, but rather varies according to time, language, and social context.

Keywords: Love, sentiment analysis, Reddit, natural language processing, emotional seasonality, BETO model, emotions, social media, digital language, artificial intelligence, affective perception.

Índice general

Índice de figuras

Acrónimos

PLN	Procesamiento de Lenguaje Natural
NRC	National Research Center Canada
VADER	Valence Aware Dictionary and sEntiment Reasoner
BERT	Bidirectional Encoder Representations from Transformers
LDA	Latent Dirichlet Allocation
TF-IDF	Term Frequency - Inverse Document Frequency
BETO	Bidirectional Encoder Representations from Transformers

Capítulo 1

Introducción

1.1. Motivación

El amor es el motor de la comprensión y es esencial en nuestras vidas para experienciar un bienestar emocional y convivir de la mejor manera posible en el contexto social. Sin importar el momento histórico al que nos refiramos, el amor siempre ha trascendido cualquier contexto, ya sea social o cultural, siempre está presente en nuestras vidas. No obstante, se adapta y se transforma según los cambios sociales y tecnológicos que se manifiesten. En este sentido, comprender el amor y explorarlo nos ayuda a entender las dinámicas interpersonales y los valores y expectativas generados que afectan a las personas en su vida diaria. Cómo se habla del amor nos permite establecer patrones y tendencias en las personas que varían a lo largo del tiempo, y esto lo podemos encontrar en línea, donde las personas comparten sus vivencias y sus complicaciones en vínculos emocionales y románticos, siendo Reddit la plataforma que más se enfoca en asesoramiento y resolución de preguntas a través de comunidades. Se explora lo que se concibe por ser amado, lo que se espera de una relación, las emociones que genera sufrir una pérdida afectiva, etcétera.

En el estudio de las percepciones y emociones en las relaciones, las redes sociales serán clave, ya que se han convertido en un foco esencial desde que han aparecido en la sociedad. Han cambiado la comunicación entre las personas. Son un nuevo contexto donde expresarse, donde existe la posibilidad de hacerlo de manera anónima y por consecuencia de manera más transparente. Las redes suponen, por lo cual, una valiosa fuente para recabar información sobre percepciones y opiniones, en particular Reddit, que destaca por ser una plataforma prácticamente exclusivamente anónima. Reddit fomenta la honestidad y la transparencia, creando un espacio de discusiones abiertas a opiniones y a la comunicación de problemas ya sean prácticos o emocionales.

Para analizar el vasto volumen de datos que podemos extraer en las redes, es esencial utilizar una herramienta que identifique y agrupe los temas principales dentro de los datos textuales, como lo son las técnicas de procesamiento de lenguaje natural (NLP). Por otra

parte también será necesario identificar el tono emocional y las actitudes de los usuarios hacia temáticas específicas, lo cual puede ser llevado a cabo por un modelado de tópicos. Las dos técnicas mencionadas, apoyadas por algoritmos de *machine learning* (ML), proporcionan insights sobre las percepciones y emociones humanas.

No será la primera vez que se haga un estudio sobre las relaciones humanas durante las redes sociales. Por ejemplo, Rama Kiran Garimella et al. (2014) estudió las rupturas románticas en Twitter. El autor concluye que los datos de las redes sociales pueden proporcionar información sobre experiencias que son comunes en la vida de la mayoría de las personas, basándose en la identificación de indicadores como el 'stonewalling", la cercanía pre y post relación, y el fenómeno de "desamigo por lotes".

En este contexto, el presente trabajo de grado propone analizar las percepciones del amor en diferentes épocas del año en la era digital a través de Reddit. Debido a la importancia y la enorme presencia del amor en todas las interacciones que vivimos, se ha considerado que el amor es un tema de interés universal. Su comprensión nos ayuda a mejorar nuestra calidad de vida y nuestra salud mental.

1.2. Objetivos

El objetivo principal del presente Trabajo de Fin de Grado en Business Analytics es analizar las percepciones y sentimientos sobre el amor en la población Española de Reddit en diferente épocas del año mediante técnicas de modelado de tópicos y análisis de sentimientos, para identificar patrones y tendencias en la expresión de este concepto en el contexto de la era digital. El objetivo se puede desglosar en objetivos específicos:

- Investigar las opiniones y percepciones sobre el amor en diferentes épocas del año y días concretos, contrastando entre ellas y averiguando el sentimiento general en cada época de la población española.

- Contextualizar la relevancia del concepto de amor en la sociedad actual, explorando su impacto en las interacciones y valores sociales de las comunidades en línea, en concreto en espacios de discusión como Reddit.

- Identificar y analizar las técnicas de procesamiento automático de textos más relevantes para extraer información sobre percepciones ciudadanas en redes sociales, evaluando su aplicabilidad en el análisis de datos textuales de plataformas en línea.

- Identificar las temáticas predominantes relacionadas con el amor en las publicaciones de Reddit mediante la aplicación de técnicas de modelado de tópicos, con el objetivo de comprender las áreas principales de discusión y su relación con las percepciones sociales.

- Analizar el tono emocional de las publicaciones extraídas de Reddit utilizando técnicas de análisis de sentimientos, para evaluar las emociones predominantes y los patrones afectivos en las percepciones del amor en la población española.

1.3. Estructura del documento

La estructura de este trabajo consta de 5 capítulos organizados que nos darán el contexto necesario para comprender cómo se han hallado las conclusiones. El primer capítulo, la introducción, contendrá la motivación y los objetivos que se pretenden lograr con este estudio. El segundo capítulo será una revisión de la literatura relevante en términos de metodología y psicología humana. El tercer capítulo, la metodología, explicará cada uno de los pasos realizados para obtener los resultados. El cuarto capítulo expondrá los resultados obtenidos a través de la metodología y proporcionará insights sobre estos. El quinto capítulo presentará las conclusiones del estudio y sus limitaciones, incluyendo además posibles líneas futuras de investigación.

Capítulo 2

Analizando percepciones en el campo de la energía en X a través de técnicas de Machine Learning: Revisión de la literatura

Entre las redes sociales, Reddit destaca por ser una plataforma abierta y transparente que dota a los investigadores de datos abiertos y gratuitos. Reddit permite que cualquiera acceda a sus publicaciones y comentarios, siempre manteniendo el anonimato de sus usuarios. Davidson (2023) describe esta red social como un lugar de contenido espontáneo y no guiado por investigadores que tiene temas variados organizados en subreddits. Además, los usuarios son, en su mayoría, anónimos por lo que tienden a expresarse con libertad. Tiene millones de usuarios que hablan en todo tipo de comunidades: se ha convertido en un lugar perfecto para estudiar cómo pensamos, debatimos y cambiamos de opinión (Medvedev y Lambiotte, 2019). Además, Reddit se caracteriza por tener mensajes concisos y por lo tanto son fáciles de analizar. Davidson (2023) demuestra con su aplicación práctica del análisis de datos de Reddit, la aplicabilidad de análisis sobre estos datos en un caso concreto: la renuncia masiva de docentes tras la pandemia. En realidad, lo que ofrece Reddit es una ventana directa a conversaciones actuales, espontáneas y muy humanas, lo que lo convierte en una plataforma muy potente para explorar cómo se sienten y se expresan ciertos grupos sociales ante situaciones específicas. Por este motivo, se ha considerado que Reddit es la plataforma social más adecuada para analizar las percepciones de los individuos del amor romántico durante diferentes momentos del año.

Davidson (2022) resalta también la importancia de tener en cuenta los problemas de representatividad que supone analizar datos en línea, ya que la mayoría de usuarios son experimentados en el mundo digital, es decir, jóvenes, y particularmente en Reddit existe una predominancia del sexo masculino. Tener en cuenta las limitaciones de este análisis y de las

fuentes de datos es esencial. Estas sujeciones serán consideradas durante el análisis y delimitarán las asunciones necesarias para el estudio.

¿Por qué las redes sociales son importantes en el contexto de la percepción y las expectativas en el amor? Simplemente, las han moldeado. Han creado un espacio de creciente aceptación de la diversidad amorosa y sexual, e incluso de modelos relacionales no tradicionales entre jóvenes (Fernández, 2024). Ya no es tan común encontrarse con creencias idealizadas (como la idea de la "media naranja" o la omnipotencia del amor). Conceptos anteriormente muy presentes en la sociedad y causantes de expectativas poco realistas que generan decepción y no permiten sentir satisfacción en la pareja (Fernández, 2024). Las redes sociales pueden considerarse un espacio donde, entre otras cosas, se moldean las expectativas y percepciones de cómo debería ser una pareja. Además, Blanco (2018) establece en su investigación que el amor romántico es una construcción social y por lo tanto va de la mano de un discurso social afectado enormemente por los medios de comunicación. Los medios legitiman ciertos modelos amorosos entre otros.

De entre las redes sociales Reddit es un medio que destaca por su transparencia. Además de ser anónimo en su mayoría, cuenta con la ventaja de no tener sesgo del experimentador. Según un estudio de Holman et al. (2015), la mayoría de investigaciones y experimentos cuentan con un sesgo generado por la propia presencia del experimentador y por el hecho de hallarse en un experimento. Esto ocurre en métodos de recolección de información tradicionales, como las encuestas o las entrevistas a usuarios. En cambio, Reddit, al proporcionar datos generados espontáneamente por los usuarios sin intervención del investigador, no cuenta con este tipo de sesgo y resulta una fuente de información más verdadera. Por los motivos anteriores, los medios de comunicación son un espacio adecuado para identificar patrones sociales. Además, supondrá una ventaja contar con un espacio más variado y con perspectivas menos idealizadas.

Tras una revisión exhaustiva de la literatura existente sobre la percepción y expectativas en el amor romántico, se ha podido comprobar que existen muy pocas investigaciones sobre este tema en particular. No obstante, existen una gran cantidad de estudios relacionados que ofrecerán una orientación en términos de metodología y marco teórico para la investigación. Una investigación relevante, tanto por metodología como por hallazgos, es la investigación sobre los ciclos sexuales de Wood, Varela, Bollen, Rocha y Gonçalves-Sá (2018). Los autores investigan si las variaciones en la reproducción humana a lo largo del año se explican mejor a través de factores biológicos o culturales. Se examinaron patrones sobre búsquedas relacionadas con el sexo en 130 países entre 2004 y 2014 a través de Google Trends durante celebraciones culturales y religiosas, hallando máximos locales en festividades culturales (Navidad en países de mayoría cristiana y Eid al-Fitr en países de mayoría musulmana). Además, el estudio apoya este argumento con más información, estableciendo que los picos preceden en aproximadamente nueve meses a aumentos en las tasas de natalidad. La investigación también incluye un análisis de sentimientos en Twitter, que mostró que durante las

festividades, las emociones colectivas eran más positivas, felices, seguras y calmadas. Una limitación de este estudio es la falta de registros en algunos países, lo que sesga el análisis hacia los países cristianos del hemisferio norte.

Otro estudio relevante en términos de metodología es el de Xu et al (2024). Este estudio ha evaluado la percepción pública de ChatGPT en Reddit a través de 23.733 publicaciones y comentarios relacionados con ChatGPT. Para examinar las actitudes del público, este estudio realiza un análisis de contenido mediante un modelado de tópicos con el algoritmo de Asignación de Dirichlet Latente (LDA) y, aunque no se especifica explícitamente, el uso de "Top 20 most common words" y n□grams es implícito para visualizar la frecuencia de términos. Como en el presente trabajo, las publicaciones y comentarios de los usuarios se clasifican en diferentes categorías a través de un análisis de sentimiento. Cada texto entró en una categoría (positivo, negativo o neutro). Además, se graficaron tendencias diarias de sentimientos para ver cómo cambiaron a lo largo del tiempo. Los autores hallan siete temáticas relacionadas con ChatGPT que pueden agruparse en tres categorías: percepción del usuario, métodos técnicos e impacto social. Los resultados del análisis de sentimiento muestran que el 61,6 por ciento de las publicaciones y comentarios tienen opiniones favorables sobre ChatGPT. También se destacan la capacidad de ChatGPT para generar conversaciones naturales con los usuarios, sin depender de un procesamiento complejo del lenguaje natural. Ofrecen sugerencias a los desarrolladores de ChatGPT para mejorar su diseño y funcionalidad de usabilidad.

Por otra parte, Göçen et al (2024) han examinado la evolución de la percepción sobre la educación superior con el objetivo de descubrir temas y sentimientos clave en el discurso global en un contexto de rápidos avances tecnológicos. Para ello analizaron 157.943 tweets de 84.423 usuarios durante un período de cinco meses, durante los cuales llegaron a su auge las herramientas de inteligencia artificial, en particular ChatGPT. En la extracción de los datos se utilizaron palabras clave específicas como "higher education", "university", "college" y relacionadas, con el objetivo de construir un corpus directamente vinculado al objeto de estudio. El preprocesamiento textual incluyó un proceso de limpieza en el que se eliminaron los duplicados y los documentos vacíos, además de una normalización textual (conversión a minúsculas, remoción de signos de puntuación, enlaces, menciones y hashtags). Se llevó a cabo una tokenización y remoción de stopwords y se generaron n-gramas para comprender la frecuencia de conceptos clave y sus contextos. El estudio utilizó técnicas de análisis de sentimientos, modelado de tópicos (LDA) y análisis descriptivo de lenguaje. LDA identificó temas latentes (tópicos) y se seleccionó el número de tópicos que tenía la coherencia más alta, es decir, el que producía temas con palabras mejor agrupadas semánticamente. Este proceso es clave para evitar terminar con temas demasiado generales o demasiado fragmentados y por lo tanto difíciles de interpretar. Entre los temas encontrados se destacan: financiación universitaria, acceso equitativo a la educación, adaptación institucional a la IA y futuro del trabajo y valor del título universitario. En el análisis de sentimientos cada tweet fue clasificado automáticamente según su polaridad emocional: positivo, negativo o neutral, y se observó

que las reacciones del público a la IA en la educación superior fueron inicialmente negativas, mientras que los tweets sobre educación superior se caracterizaron por la positividad y el optimismo. A pesar de los problemas conocidos (como las deudas estudiantiles), el discurso en redes tenía un tono optimista. Los tweets negativos que hablaban de problemas estructurales generaron críticas sobre el coste elevado de estudiar, el acceso desigual a la educación y la falta de adaptación tecnológica.

De manera similar, Feldhege et. al (2019) realizaron una investigación en las comunidades en línea sobre la depresión y los temas predominantes en las comunidades en línea sobre la depresión y su relación con la participación en forma de comentarios. Para ello se creó un modelado de tópicos con 26 temas y 16.291 publicaciones mediante LDA. La proporción de los temas en el corpus se correlacionó con cinco medidas de participación: suma de puntuaciones, número de comentarios, proporción de publicaciones a comentarios, frecuencia de publicación y número de palabras. Los datos fueron extraídos del subreddit r/depression, y tras la limpieza, lematización y eliminación de stopwords se realizó un modelado de tópicos en el que se identificaron los 26 temas latentes. De los tópicos generados, 7–8 por ciento del corpus se refirió a "Romantic relationships" y "Ending Relationships". Tópicos como "Offering Support" y "Small Talk" mostraron que el subreddit funciona como una red de ayuda emocional, donde los usuarios también conversan sobre amor y amistades, lo que confirma que Reddit es un espacio apropiado para desarrollar este trabajo. La limitación más significativa en esta investigación es que no se disponía de información sobre los datos demográficos ni el estado de salud mental de los usuarios, y, como relatan los autores, el modelado de tópicos no puede capturar elementos de estilo y tono del texto.

Rosamma KS (2024) también utilizó la plataforma Reddit para evaluar los temas principales discutidos en Reddit relacionados con el estrés y la ansiedad, y analizar el sentimiento general de las publicaciones. Al igual que en las investigaciones anteriores, se realizó la tokenización, eliminación de stopwords y lematización de los documentos. A través de 3.765 publicaciones de Reddit se identificaron cinco categorías principales de tópicos mediante el algoritmo LDA: descontento general y falta de dirección, ataques de pánico y ansiedad, síntomas físicos de ansiedad, estrés y problemas de salud mental y búsqueda de ayuda para la ansiedad. Posteriormente el análisis de sentimientos TextBlob mostró una puntuación neutral en su mayor parte. La autora utiliza diversos tipos de visualizaciones, como nubes de palabras, gráficos de barras y gráficos circulares. Este estudio es fundamental para los profesionales e investigadores de la salud mental ya que genera una base teórica de la que partir a la hora de investigar las causas y temas de interés que habitan en las personas con problemas de ansiedad.

La revisión de la literatura relacionada con la metodología del presente trabajo muestra una estructura clara y organizada, que servirá como referencia a la hora de establecer la metodología. De la misma forma se ha podido apreciar la cantidad de datos necesarios para llevar a cabo un muestreo significativo y se han descubierto algoritmos y métodos clave que nos

servirán en adelante. Asimismo, se resalta la presencia y capacidad del modelado de tópicos para identificar los principales temas de conversación. Resulta útil para todo tipo de investigaciones que quieran reducir la complejidad de un conjunto de documentos textuales con facilidad de implementación.

Por otra parte, se ha realizado una revisión de la literatura científica sobre el amor romántico en general, para establecer ciertas bases psicológicas previo al estudio. Un estudio que conceptualiza el amor de una manera concreta es "Para Entender el Amor Romántico" (Carrillo L., 2023). El autor aporta una perspectiva crítica y muy interesante hacia el amor romántico. Establece que el amor romántico no es natural, sino que se surge de construcciones sociales y culturales y a veces se basa en mitos como la media naranja, el amor eterno, los celos como prueba de amor o el sufrimiento como parte inherente de una relación, es decir, la normalización del sufrimiento y la generación de expectativas inalcanzables que inevitablemente conducen a la frustración. Por otra parte, Carrillo (2023) distingue entre dos tipos de amor: el amor romántico, caracterizado por la pasión, la intensidad y la idealización del otro, y el amor de compañía, caracterizado por el compromiso, la intimidad y el cuidado mutuo. La perspectiva de Carrillo sobre el amor de compañía se alínea con la teoría triangular del amor de Robert Sternberg, que se compone de las siguientes características principales: la intimidad, que se refiere a la conexión emocional, la cercanía y la confianza; la pasión, que se relaciona con la atracción física y sexual, y el compromiso, que implica la decisión consciente de mantener y trabajar en la relación a largo plazo (Almudena Prats, 2024).

De manera similar, Wormley, Schaller y Varum (2023) han estudiado cómo las estaciones afectan al comportamiento y la psicología humana. Recogen estudios científicos para investigar sobre el amor romántico y observan que las emociones positivas tienden a disminuir durante los meses de invierno. Además, los autores determinan que los ciclos estacionales crean variaciones en la psicología humana a través de la interacción de factores meteorológicos, ecológicos y socioculturales. La exposición a la luz solar afecta la producción de melatonina y serotonina, neurotransmisores relacionados con el sueño y el estado de ánimo, y por eso la escasez de luz solar en ciertas épocas del año fomenta el trastorno afectivo estacional. Similarmente, las temperaturas extremas durante el verano y el invierno generan más irritabilidad y agresividad, mientras que las temperaturas moderadas de primavera y otoño fomentan la sociabilidad por lo que las personas están más abiertas a entablar nuevas relaciones sentimentales. De entre los factores ecológicos que afectan a la psicología humana estudiados, se ha considerado relevante la presencia o ausencia de vegetación (en diferentes épocas del año), ya que los espacios verdes disminuyen el estrés y por lo tanto fomentan un mejor estado de ánimo dentro de las parejas y en la disposición de crear nuevas relaciones. Esta investigación, como la anterior, establece que las festividades generan más disponibilidad afectiva, pero también expone un contraargumento, ya que apoya que las festividades fomentan los extremos, y por lo tanto también es una época en la que el estrés y la ansiedad están muy presentes en muchas personas, ya sea por motivos financieros o emocionales. En

resumen, las estaciones impactan en una amplia gama de fenómenos afectivos, cognitivos y conductuales.

Debido a la fuente de información que se empleará, nuestros datos serán en su mayoría provenientes de jóvenes que expresan sus inquietudes en las redes. Por esto, conviene centrarse en la juventud y sus percepciones en la previa revisión de la literatura. Así, un estudio académico relevante es "Creencias Sobre el Amor y Bienestar Durante la Adultez Emergente" (2024). Este estudio emplea escalas basadas en la teoría de estilos de amor de John Lee, recogiendo datos a través de un cuestionario realizado por 631 jóvenes entre los 18 y 29 años (el rango de edades considerado como la adultez emergente) de manera voluntaria. Las escalas empleadas clasifican a los participantes en categorías de creencias sobre el amor romántico, viendo en qué medida aceptan mitos o ideas idealizadas sobre el amor. Éstas son: la Escala de Actitudes hacia el Amor, la Escala de Mitos del Amor Romántico y la Escala de Florecimiento. Se llega a la conclusión de que las mujeres tienden hacia estilos de amor pragmáticos, dan más importancia a la estabilidad y el compromiso; los hombres se inclinan más hacia estilos lúdicos, priorizan el disfrute y la diversión. Los hombres también mostraron una tendencia hacia estilos altruistas basados en el cuidado y la entrega desinteresada.

De manera similar, Granero y Piedra (2023) realizaron un estudio exploratorio de las percepciones de la población adulta-joven, entre 18 y 40 años residentes en la Comunidad de Madrid sobre el posible papel del amor romántico frente a otras alternativas afectivo-amorosas. Se realizaron 35 entrevistas individuales donde los participantes aportaron su definición personal del amor, los tipos de relaciones vividas, su uso de redes sociales en la vida afectiva y se evaluó también la influencia de movimientos como el feminismo y el LGTBIQ+. La muestra se eligió por conveniencia y por bola de nieve, siempre tratando de asegurar una diversidad de identidades de género, orientaciones sexuales, nivel educativo y situación laboral. A través del análisis temático usando herramientas de análisis cualitativo se llegó a la conclusión de que el amor romántico se mantiene como referente hegemónico, generalmente buscando nuevas formas de relación más igualitarias en términos de género y de orientación sexual. Otro hallazgo interesante del estudio es la posibilidad de desligamiento de las relaciones sexuales del anspecto emocional. Los autores resaltan el papel mediador de las redes en la construcción de nuevas realidades amorosas, lo cual apoya la conveniencia de realizar este trabajo a través de Reddit. No obstante y aunque las redes incrementen la aceptación de la diversidad afectiva, también presentan riesgos de pérdida de la identidad y la seguridad, al hacer accesible un abanico de posibilidades ideológicas en la orientación romántica y sexual. Las conclusiones de esta investigación se ven respaldadas por Sacoto, Moreta-Herrera y Jayo (2020), que tras un estudio descriptivo con 590 jóvenes, indican que entre ellos se combina el rechazo por la versión tradicional del amor con nociones modernas. Otros hallazgos de relevancia son que los jóvenes mantienen valores tradicionales en cuanto a compromiso y fidelidad, y que son las mujeres las que lideran el cuestionamiento y la redefinición de las normas tradicionales en las relaciones de pareja.

En conjunto, esta revisión evidencia cómo las concepciones del amor romántico en la juventud actual están atravesadas por tensiones entre discursos tradicionales y nuevas formas de vinculación emocional. Podemos ver también que el concepto del amor se explora como una construcción social reflejada en las redes sociales. Generalmente los jóvenes cuestionan los modelos idealizados del amor. Estos hallazgos otorgan un marco conceptual sólido y justifican la pertinencia de analizar cómo se expresa el amor en plataformas digitales, especialmente entre los jóvenes. El amor, como todos los fenómenos sociales, no es una experiencia universal y estática, sino que está influenciado por el discurso social y las expectativas, como los roles de género.

Capítulo 3

Metodología de Análisis de Datos

En este capítulo se presenta la metodología a seguir en el desarrollo del trabajo con el fin de entender las percepciones y los sentimientos en el amor romántico a lo largo de diferentes épocas del año. Analizaremos las publicaciones extraídas de comunidades específicas para entender no sólo lo que dicen sino también las palabras elegidas, las emociones predominantes, las narrativas que se repiten y el tono empleado. Para lograr nuestro objetivo, seguiremos cinco etapas de desarrollo del proyecto. Éstas son: preparación de los datos, pre-procesamiento de los datos, análisis descriptivo de los n-gramas, modelado de tópicos y análisis de sentimiento.

Cada uno de los pasos anteriores es esencial para asegurar que nuestros datos son de calidad (especialmente la preparación de los datos y el pre-procesamiento) y que nuestro análisis es significativo. La etapa de preparación de los datos tiene como objetivo descargar una amplia base de datos que posteriormente nos permita eliminar contenidos irrelevantes, vacíos, repetidos, etc., sin que afecte a la calidad del análisis por escasez de observaciones. Posteriormente se realizará un pre-procesamiento de los datos, es decir, una limpieza y una estandarización para reducir la dimensionalidad del corpus. Esta limpieza incluye la tokenización (transformación de frases en tokens individuales), la eliminación de caracteres no alfabéticos, stopwords, elementos extremos y duplicados y la lematización. Por otra parte, el análisis descriptivo de n-gramas nos servirá para identificar palabras y conjuntos de palabras que aparecen juntas en el corpus, y de esta manera entender su relevancia dentro de éste. Posteriormente empezaremos el modelado de tópicos, que nos indicará los temas principales en relevancia y frecuencia que aparecen en los documentos del corpus. Para finalizar el análisis, llevaremos a cabo un análisis de sentimientos que aportará un mayor entendimiento de las emociones presentes en los textos a lo largo de diferentes épocas del año.

3.1. Adquisición de los Datos

El análisis de sentimiento de los contenidos de Reddit demanda una base de datos coherente, representativa y suficientemente amplia de publicaciones (reddits). En este sentido, es necesario determinar una herramienta eficiente en la extracción de datos de Reddit y establecer filtros. La adquisición de los datos tiene como objetivo encontrar comunidades de Reddit en las cuales las discusiones sean orientadas a las relaciones amorosas. Además, es importante tener en cuenta que muchos de los datos que conseguimos se hallan vacíos o repetidos, por lo que debemos de contar con una base de datos muy amplia y con mucho márgen de eliminación de contenido. A continuación se explicará cómo se obtendrá y preparará este corpus.

La herramienta empleada como entorno de desarrollo principal será Python, aprovechando la librería Praw para conectarse a la API de Reddit, lo cual nos permite una autenticación segura y extracción de posts de distintos subreddits de manera gratuita y con manejo de límites de tasa. A través de un script principal de adquisición, tras autenticar la aplicación en Reddit mediante credenciales, podremos iterar sobre búsquedas y navegación de contenido y de esta manera se poder controlar y reproducir los datos deseados. Para que la recogida de datos tolere interrupciones y podamos evitar la pérdida de datos en caso de errores o límites de tasa se establecerán mecanismos de registro de progreso y de guardado de puntos de control.

Para emplear la herramienta de extracción de datos es necesario encontrar subreddits donde sea probable hallar conversaciones que hablen sobre relaciones amorosas y percepciones afectivas en españa, como r/es o r/relaciones. Este será nuestro primer filtro. Para elegir los subreddits se realizará un estudio previo para asegurar la cobertura de comunidades generales y comunidades especializadas en relaciones y amor. Dado que Reddit no siempre provee metadatos de ubicación geográfica, se tendrá en cuenta la dificultad de filtrar por país, y por este motivo se filtrará a través del idioma y encontrando comunidades españolas. Es importante reconocer que la muestra puede incluir usuarios de diversas localizaciones geográficas. Esta limitación se tendrá en cuenta y se reducirá en la medida de lo posible.

El segundo filtro utilizado será la fecha. Es esencial tener información de diferentes épocas del año para poder capturar el efecto temporal en las percepciones del amor. Así, se elegirá un filtro con un rango temporal de los últimos 7.000 y 10.000 días, dotandonos de datos de diferentes años y situaciones sociales.

Teniendo como referencia la revisión de la literatura y de trabajos previos sobre las expectativas y percepciones del amor se decidirán las palabras clave y expresiones relevantes con las que realizaremos el tercer filtro para realizar consultas de búsqueda en títulos y cuerpos. Por ejemplo: “amor”, “cita”, “romántico”, “relación”, “pareja” “cita”, “romántico”, “relación”, “pareja”, “enamorado”, etc. Cada una de las observaciones tendrá las siguientes variables: la publicación, el título, la fecha, la URL, el subreddit, el autor, el score (la resta

entre los “me gusta” o “upvotes” y los “no me gusta” o “downvotes” del post, número de comentarios, si contiene video o no, si contiene imagen o no, el ratio de “upvotes” y por último, el permalink (link permanente de la publicación). Si la recolección da como resultado muy pocos casos en ciertos periodos estacionales, se podrá ampliar el rango de subreddits o ajustar términos.

3.2. Pre-procesamiento de los Datos

Una vez la recogida de datos se haya llevado a cabo, se realizará la etapa de pre-procesamiento de los datos, cuyo objetivo es depurar u normalizar el texto obtenido para convertirlo en datos limpios, homogéneos y representativos. Esta etapa tiene varias fases, comenzando con la configuración y descarga de recursos lingüísticos. Esto consiste en verificar que los paquetes de tokenización (punkt) y de *stopwords* en español están descargados.

La segunda fase aborda la carga de los datos crudos y la detección del idioma y filtrado de contenidos en español. Esto es muy importante ya que cualquier análisis debe ser realizado en español y de lo contrario los programas no serán capaces de detectar tópicos y emociones. Revisaremos además que la base de datos incluye las columnas necesarias nombradas en la etapa anterior.

Procederemos con un filtrado temático para asegurar que el subconjunto de textos trate efectivamente de temas relacionados con el amor, la pareja o las relaciones, ya que puede haber residuos irrelevantes después de la búsqueda por palabras clave. Para ello se definirá un conjunto de palabras clave representativas y se verificará para cada texto si contiene alguna de estas palabras, eliminando aquellos que no cumplan con este criterio.

A continuación eliminaremos los duplicados, paso esencial antes de aplicar los pasos costosos en tiempo de tokenización y lematización, ahorrando así tiempo de cómputo y evitando el sesgo en conteos de palabras por publicaciones repetidas. Finalizada la limpieza inicial, pasaremos al procesamiento de texto, fase que consta de varias sub-etapas aplicadas secuencialmente a cada texto restante. Lo primero será la tokenización, dividiendo el texto en unidades léxicas (palabras o símbolos) a partir del tokenizador de NLTK en español. Después, limpiaremos los tokens para quedarnos sólo con los que aportan valor al análisis (se eliminarán URLs, menciones, números, etc.). En la misma línea, se convertirán todos los tokens a minúsculas, reduciendo las dimensiones al coincidir más palabras. Eliminaremos las *stopwords* (palabras muy frecuentes que aportan poco valor analítico) con la lista de *stopwords* provista por NLTK y reduciremos cada token a su raíz con lematización, de tal forma que variantes morfológicas de la misma palabra se unan a una forma común. Esto se hará a través del stemmer en español *SnowballStemmer* de NLTK. Tras el procesamiento, algunos registros pueden quedar vacíos y serán eliminados, al igual que las palabras extremas, que son los token que aparecen en muy pocas ocasiones o aparecen en casi todos los documentos.

Tras la limpieza, en algunos casos, dos textos originalmente distintos pueden volverse idénticos, por lo cual se realizará una segunda eliminación de duplicados. Por último, obtendremos métricas descriptivas sobre el corpus obtenido para saber sobre su tamaño, la distribución de términos clave y la riqueza de vocabulario. Los datos pre-procesados se guardarán en formato CSV para poder convertirlo a Excel.

Con estos pasos se obtendrán unos datos estructurados y útiles para el procesamiento, con eficiencia computacional y mayor calidad. A partir de esta base de datos se hará un análisis descriptivo utilizando n-gramas.

3.3. Exploración de N-Gramas

En este apartado se detalla el procedimiento para extraer y analizar secuencias de palabras (n-gramas) incorporando una dimensión temporal (estaciones del año). El objetivo es identificar las combinaciones léxicas más relevantes en cada periodo. Un n-grama se define como una secuencia contigua de n tokens en una observación. En esta línea, un unigrama se caracteriza por tener n=1 términos, un bigrama n=2 términos y un trigrama n=3 términos. Los n-gramas nos permiten detectar frases o expresiones frecuentes en los posts de Reddit relacionados con el amor, y a partir de estas expresiones se puede analizar cómo cambian las combinaciones en función de la estación, además de aportar una evaluación cuantitativa de la relevancia de los conjuntos de palabras en el corpus completo. Por estas razones, el análisis de n-gramas supone una herramienta útil en el análisis exploratorio de los datos.

La herramienta cuantitativa que vamos a utilizar es TF-IDF (*Term Frequency - Inverse Document Frequency*). TF-IDF pondrá la frecuencia de aparición en cada documento respecto a la frecuencia en el corpus completo. Esto quiere decir que pondera la frecuencia de aparición en cada documento respecto a la frecuencia en el corpus completo, aportando una medida de importancia relativa. De esta manera un término muy frecuente en un documento tendrá un TF muy alto, lo que indica relevancia, y un término muy común en todos los documentos tendrá un IDF muy bajo, indicando que aporta poca información.

Partiendo del archivo Excel generado en la fase de pre-procesamiento, se extraerá, para cada observación, el mes, día y año, creando columnas auxiliares, lo que nos permitirá determinar la estación del año correspondiente. Procederemos trabajando con grupos de periodos (estaciones) y se analizará que existan suficientes registros en ese grupo para un análisis fiable. Antes de ponderar con TF-IDF se controlará la frecuencia absoluta de cada n-grama en el conjunto para entender la distribución cruda de secuencias de palabras. El siguiente paso será obtener la matriz TF-IDF para seleccionar los n-gramas relevantes ordenándolos por orden descendente de relevancia. Así extraemos para cada periodo los términos o secuencias con mayor relevancia relativa en contexto temporal. Con los textos convertidos en vectores, se medirá qué tan parecidos son entre sí usando métricas como la similitud del coseno. Identi-

ficaremos relaciones o temas comunes entre los documentos y agruparemos aquellos textos que tienen características similares. Es importante señalar que el análisis de n-gramas trabaja con texto preprocesado, por lo que la calidad de los resultados depende de la efectividad del pre-procesamiento.

3.4. Modelado de Tópicos

El análisis descriptivo de n-gramas nos da un marco conceptual sobre la estructura de los datos a partir del cual podemos adentrarnos en el modelado de tópicos. El modelado describe las temáticas latentes en grandes volúmenes de datos textuales, en este caso, para las diferentes estaciones del año. El algoritmo que emplearemos en este trabajo será *Latent Dirichlet Allocation* (LDA), que se enfoca en hallar patrones temáticos extraídos de redes sociales, en este caso, Reddit. El proceso de modelado de tópicos consiste en varias etapas:

1. Carga y procesamiento de los datos
2. Preparación del corpus
3. Selección del número óptimo de tópicos
4. Entrenamiento de modelos LDA y extracción de tópicos
5. Exportación de resultados
6. Visualización de resultados

Los datos serán importados desde un archivo Excel, conteniendo todas las columnas que estructuramos en el pre-procesamiento. Asimismo, los registros vendrán incorporados con la columna de su estación correspondiente, de la misma manera que el análisis de n-gramas. La preparación del corpus consistirá en el descarte de textos con menos de dos tokens. Uno de los aspectos más decisivos en este análisis es la selección del número de tópicos, pues de ello dependen los resultados obtenidos y la calidad del análisis. El número de tópicos podrá ser elegido en el rango 2-10. Se realizará un LDA para cada valor con una medida de similitud semántica entre las palabras más representativas de cada tópico: *coherence score*. Cuanto más alta sea esta métrica para un valor de tópicos, más coherentes son los tópicos internamente. Por este motivo el modelo elegirá el número de tópicos que mejor *coherence score* tenga. Es importante que este proceso sea realizado para cada estación.

Una vez determinado el número de tópicos por estación podemos extraer las palabras claves más representativas de cada tópico, también determinando otros datos relevantes como su probabilidad de aparición y la coherencia del modelo. Los datos serán resumidos, exportados y visualizados para su análisis y entendimiento y para hacer comparaciones entre las distintas estaciones del año.

3.5. Análisis de Sentimientos

Finalizado el modelado de tópicos podemos analizar el tono emocional de las expresiones recopiladas. Esta técnica clasifica las emociones del texto asignando una polaridad positiva, negativa o neutra. Trabajos de similar metodología emplean el análisis de sentimiento VADER, pero para este estudio se ha optado un modelo de tipo transformer: BETO (una adaptación de BERT en español). Es un modelo de aprendizaje profundo pre-entrenado que capta los matices del lenguaje en textos breves. Resulta adecuado ya que está adaptado a textos informales y espera la presencia de expresiones coloquiales y errores gramaticales. Además, a cada texto le otorga una probabilidad asociada a cada clase (positivo, negativo o neutral). Los lexicones tradicionales a veces pasan por alto ambigüedades y subjetividades, por lo que este método es una buena opción, más sofisticado y completo. BETO incorpora técnicas de ajuste contextual (expresiones de negación invierten parcialmente la polaridad, conectores de contraste cambian el tono emocional, etc.). También se generarán métricas como el score promedio y el número de publicaciones extremas para analizarlas.

Partiendo del archivo de Excel con la base de datos (limpia y normalizada) se realizará el análisis de sentimiento para cada estación, para así poder comparar entre ellas. Después, se detectarán las características textuales complementarias, lo que permitirá captar matices del lenguajes. Estos serán los negadores (no, nunca, jamás, etc.), los correctores de contraste (pero, sin embargo, por el contrario, etc.), los signos de puntuación enfática (!, ?) y patrones de tres o más caracteres idénticos consecutivos. BERT considera las características complementarias en la asignación de sentimiento, por ejemplo, cuando hay más de una puntuación enfática se considera que hay mayor intensidad emocional.

Para cada texto se generarán dos valores: un score base y un score ajustado que incorpora factores contextuales. El modelo clasificará cada entrada entre muy positivo, positivo, neutral, negativo y muy negativo. Los resultados se exporarán a un Excel para su análsis y visualización.

Capítulo 4

Resultados

El apartado de resultados presentará el proceso realizado con los datos desde su recogida hasta su descarga final como base de datos estandarizada. Todo este proceso está presentado en el Capítulo 3, sirviendo este apartado como desarrollo de cada paso del proceso propuesto. Se explicará el código utilizado (que se puede encontrar al final del trabajo) y así entenderemos los datos y sus características. Ésto se logrará a partir de un análisis exploratorio de los datos. Posteriormente realizaremos un modelado de tópicos y un análisis de sentimiento. Durante todo el proceso se realizará un análisis sobre los datos y se mostrarán los resultados obtenidos a partir del código.

4.1. Adquisición y preparación de los datos

En este apartado se presentan los resultados del proceso de adquisición de los datos a partir de un filtrado de datos de Reddit. Se hablará de métricas de volúmen, rango temporal y estadísticas descriptivas de engagement y procedencia de los posts, métricas que resultan relevantes para evaluar las percepciones y expectativas en el amor durante diferentes épocas del año.

De ahora en adelante, cuando hablemos de palabras clave nos estaremos refiriendo a la siguiente lista: “ex-”, “ex”, “mi ex”, “su ex”, “’solter”, “casad”, “divorciad’, “quedada”, “salida”, “’ligue”, “rollo”, “aventura”, “conocer gente”, “busco”, “buscar”, “’dating”, “match”, “app”, “’san valentin”, “valentine”, “’aniversario”,”’boda”, “casarse”, “’compromiso”, “pedida”, “’corazón”, “querer”, “’amar”, “’sentimientos”, “emociones”, “’conquistar”, “seducir”, “primera cita”, “primera vez”, “’conocí”, “presentar”, “presentó”, “pareja”, “relación”, “relaciones”, “amor”, “novio”, “novia”, “esposa”, “marido”, “matrimonio”, “cita”, “citas”, “dating”, “salir”, “ligar”, “enamorar”, “crush”, “tinder”, “badoo”, “soltera”, “soltero”, “ruptura”, “separación”, “boda”, “aniversario”, “san valentín”, “valentine”, “corazón”, “querer”, “conocer”, “’sexo”, “celos” y “enamorado”.

En la extracción de datos los filtros se han aplicado en diferentes fases. Primero se ha

hecho un filtrado por palabras clave ordenando primero por relevancia, luego por novedad, luego los *hot* por mes, luego los top de cada año y de cada més y por último por mayor cantidad de comentarios. El segundo filtrado funciona de manera invertida: primero se ha filtrado por novedad, *hot*, rising y controversial, y luego se han seleccionado solamente las publicaciones que contenían las palabras clave. En la tercera extracción se han extraído publicaciones cronológicamente a través de 100 páginas con el filtrado de las palabras clave. Por último, se ha realizado una búsqueda por autores populares filtrando por palabras clave.

Tras aplicar los filtros de idioma y temática, se obtuvo un total de 10.180 de publicaciones en Reddit para el análisis. El rango temporal de las publicaciones abarca desde el 12 de septiembre de 2007 hasta el 15 de mayo de 2025, lo cual cubre aproximadamente 18 años de actividad. Es importante tener un mínimo de 7.000 publicaciones antes de realizar la limpieza para que el análisis sea significativo, ya que una gran cantidad de estos registros se eliminarán y nuestro corpus se verá enormemente reducido. Por esto, se ha considerado que 10.180 publicaciones son suficientes para observar variaciones estacionales y posibles tendencias a lo largo de los años.

El incremento de popularidad de Reddit en España no ha sido hasta los últimos años, lo cual explica la distribución mensual de publicaciones extraídas en la Figura 4.1, en la que la mayoría de los datos son de los últimos tres años. La tendencia ascendente en la cantidad de publicaciones por mes también se explica por la dificultad que supone recolectar datos más antiguos. Otra posible explicación de la distribución es que hasta octubre de 2022 no surgió la comunidad “relaciones” en Reddit, comunidad de la cual hemos extraído la mayoría de nuestros datos. Se ha considerado que la diferencia de cantidad de posts entre diferentes años no supondrá un problema, ya que las estaciones de diferentes años serán analizadas en conjunto, y lo que se comprará no serán diferentes años sino diferentes épocas del año.

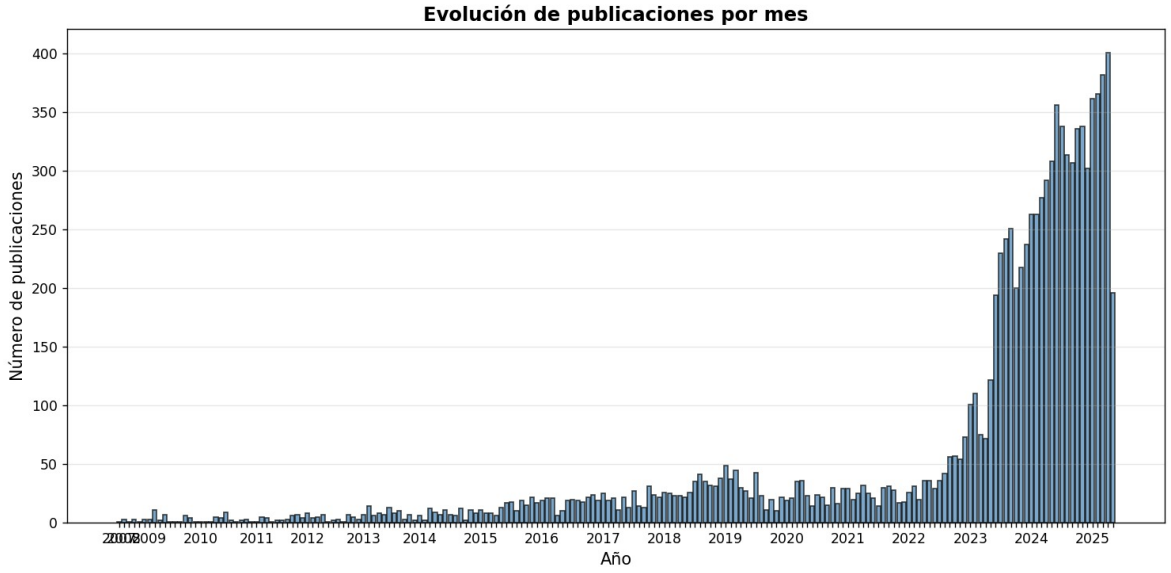

Figura 4.1: Cantidad de publicaciones en Reddit mensualmente. Elaboración propia

Para explorar la distribución de los scores nos centraremos en los upvotes, ya que son las publicaciones con popularidad y el valor del *score* se calcula restando los *downvotes* a los upvotes. La Figura 4.2 muestra la distribución *upvote ratio*, calculado dividiendo el número de *upvotes* entre el número total de votos de la publicación. La distribución de los *scores* tiene un valor medio de 28,57, un valor mínimo de 0 y un valor máximo de 2.257 (estos valores se han calculado fácilmente en una hoja de Excel). La mayoría de los posts presentan un *score* bajo/moderado, con un pequeño porcentaje de publicaciones que alcanzan un *score* significativamente alto, lo que sugiere que sólo un subconjunto de posts sobre relaciones obtiene mayor atención o comparte experiencias particularmente resonantes.

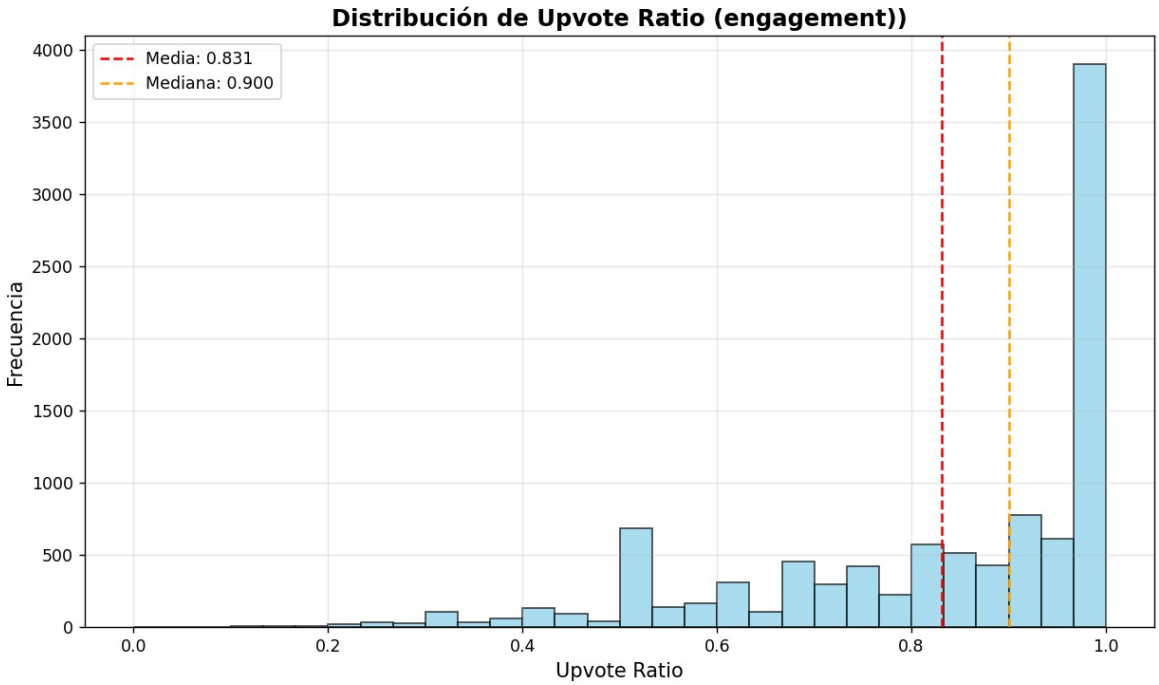

Figura 4.2: Distribución del *upvote ratio*.
Elaboración propia

En relación a la procedencia de las publicaciones encontramos una distribución entre cuatro subreddits: r/relaciones, r/spain, r/es y r/españa. La Figura 4.3 nos enseña que la gran mayoría de publicaciones encontradas son del subreddit r/relaciones, con 5.719 publicaciones, ya que es el que más ha cumplido con los filtros establecidos. Con esta información podemos concluir que los filtros han sido bien elegidos, ya que, como es lógico, el espacio donde más posts deberíamos encontrar es en una comunidad dirigida al debate sobre las relaciones amorosas.

Después de este pequeño análisis descriptivo hemos logrado entender mejor la estructura de los datos, y procedemos a hacer una limpieza de duplicados y textos vacíos, lo que corresponde a la fase de pre-procesamiento de los datos.

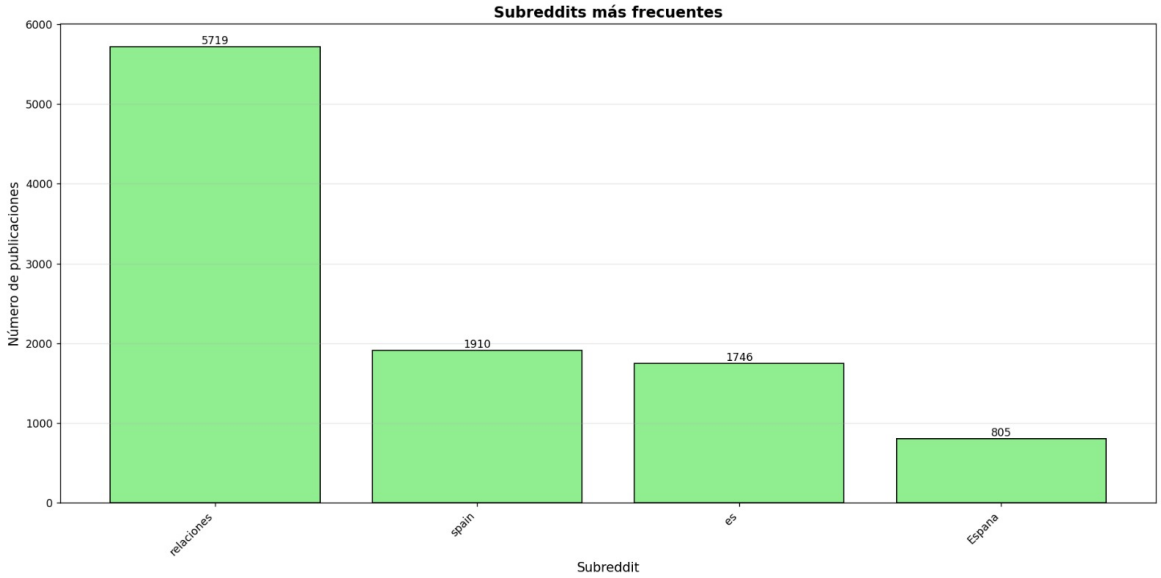

Figura 4.3: Distribución de los datos por subreddits. Elaboración propia

4.2. Pre-procesamiento de los datos

En este apartado se presentan los resultados de la fase de pre-procesamiento descrita en el Capítulo 3, aplicada al conjunto de 10.180 publicaciones de Reddit. Este proceso es esencial para eliminar el ruido y transformar el corpus en una muestra representativa. Para ello, se ha realizado una eliminación de duplicados y textos vacíos o que solo contengan espacios (lo cual mejorará la eficiencia del programa al garantizar que el pipeline de tokenización y lematización trabaje únicamente con entradas potencialmente útiles). El número de publicaciones quedó reducido a 7.510. Debemos tener en cuenta que tras hacer la limpieza esta etapa se repetirá, ya que dos textos pueden convertirse en copias exactas. Posteriormente, se han separado todos los textos en tokens, lo cual nos permite también eliminar los tokens no deseados: URLs, menciones, hashtags, números puros, puntuación y tokens con caracteres no alfabéticos. De los tokens extraídos, se han descartado todos aquellos en la lista de *stopwords* de NLTK en español, ampliada con *stopwords* específicas para este corpus: ’http’, ’https’, ’www’, ’jpg’, ’png’, ’gif’, ’pdf’, ’docx’, ’xlsx’, ’pptx’, ’zip’, ’rar’, ’com’, ’org’, ’net’, ’si’, ’sí’, ’no’, ’más’, ’solo’, ’también’, ’ya’, ’así’, ’aquí’ y algunas temporales como ’ayer’ y ’hoy’. Tras este filtro el vocabulario se concentra en términos con potencial semántico relevante. A partir del corpus restante se ha aplicado la lematización de Spacy en español para agrupar variables morfológicas, resultando en una lista de tokens en la que cada elemento se transforma en su lema. Tras estas etapas, se reconstruyó la columna ’Texto Final’ para que contenga los tokens lematizados concatenados en cada registro. Finalmente, se eliminaron aquellas publicaciones cuyo ’Texto Final’ resultó vacío, lo cual es crucial para evitar errores en etapas de modelado (vectorización de textos vacíos) y asegurar que cada registro aportará términos al corpus. Para reducir el ruido residual y la dimensionalidad se aplicó el filtro de

palabras extremas a partir del cálculo de frecuencias total de cada token en todo el corpus. Aquellas con frecuencia menor o mayor que un umbral establecido (los token que aparecen en menos de 3 documentos o que aparecen en más del 95 por ciento de los documentos) se han considerado palabras extremas. Tras el pre-procesamiento de los datos, el corpus se ha reducido de 10.180 publicaciones a 5.639, alineándose con la expectativa de descartar ruido y mantener un corpus de tamaño suficiente.

Las palabras más frecuentes y su frecuencia hallada (tras la lematización) son: decir: 25090, hacer: 24502, él: 18416, querer: 1498, sentir: 11145, poder: 11088, yo: 10834, hablar: 9284, cosa: 9259, pasar: 9009, ver: 8894, relación: 8874, vez: 7457, ser: 7421, ir: 6868, amigo: 6838, tiempo: 6780, dejar: 6365, dar: 6217, persona: 6040. La frecuencia de cada palabra queda representada en la Figura 4.4. Palabras como mujer, novio, relación, gustar, chica y pareja nos enseñan que el corpus realmente trata temáticas relacionadas con el amor romántico, lo que demuestra que el filtrado y la limpieza han conservado la temática central del corpus. La nube de palabras también revela emergencias de términos secundarios (por ejemplo, referencias de tiempo), lo que anticipa posibles tópicos interesantes a explorar en la siguiente fase.

Figura 4.4: Nube de palabras frecuentes en el corpus. Elaboración propia

El corpus restante es una base de datos limpia con la que podremos examinar combinaciones frecuentes de palabras (n-gramas) y profundizar en los temas y expresiones recurrentes de las discusiones en Reddit.

4.3. Análisis de N-gramas

Tras ejecutar el código diseñado para la extracción y análisis de n-gramas con componente temporal, se generaron resultados diferenciados por estaciones del año (primavera, verano, otoño, invierno). Esto quiere decir que los grupos temporales se analizaron por separado ponderando su importancia relativa a través de la métrica TF-IDF. Para cada categoría (unigrama, bigrama y trigrama), se mencionan ejemplos de los top 5 n-gramas detectados y se interpreta su posible significado en el contexto analizado, naturalmente, estos ejemplos se repetirán para bigramas y trigramas. Cada estación se ha aproximado clasificando por meses.

Empezamos con la primera estación. En primavera los cinco unigramas más relevantes (con mayor valor TF-IDF) han sido: hacer, decir, querer, sentir y poder. Podemos ver representada su frecuencia en la Figura 4.5. Los verbos "hacer" o "decir" sugieren una fase activa de acción y comunicación, podría ser para planificar actividades o para expresar sentimiento. Por otra parte, "querer" y "sentir" reflejan la exploración emocional y apertura afectiva típica de nuevos comienzos. Por último, "poder" puede aludir a la confianza o al empoderamiento dentro de las relaciones sentimentales. Las siguientes estaciones, verano, otoño e invierno, vienen caracterizadas por los mismos cinco unigramas que la primavera. Aunque el contexto temporal vaya cambiado, la manutención de expresiones emocionales similares sigue siendo la norma. Esto nos lleva a concluir que, o bien las personas tienen una forma de hablar muy constante a lo largo del año o bien no existe una real diferencia en las percepciones del amor durante diferentes épocas del año. Pasemos a explorar los bigramas.

Los cinco bigramas predominantes en primavera son: hacer sentir (TF-IDF = 0.0774, Frecuencia = 296), decir querer (siguiendo el mismo orden: 0.0733, 314), poder hacer (0.0671, 257), sentir mal (0.0638, 256) y querer hacer (0.0635, 238). "Hacer sentir" sugiere afectar a alguien de alguna manera con acciones o palabras, probablemente en contextos de afecto o malestar emocional. "Querer hacer" complementa "decir querer" en la motivación hacia iniciativas afectivas. De manera similar, "decir querer" también puede relacionarse con la expresión de intención o de afecto. Podría estar remarcando la importancia de la comunicación directa de sentimientos. "Poder hacer" se inclina más al empoderamiento y la reflexión sobre la capacidad de actuar, mientras que sentir mal sugiere reflexiones sobre malestares emocionales.

Por otra parte, los cinco bigramas predominantes en invierno son: hacer sentir (0.0793, 262), decir querer (0.0785, 291), primero vez (0.0661, 226) y sentir mal (0.0594, 217). Similar a primavera, señala discusiones sobre cómo ciertas acciones o palabras influyen en las emociones durante el verano. Por otra parte, "primero vez" sugiere experiencias iniciales en un contexto afectivo o íntimo, característico de experticias durante el verano. "Hacer sentir" en este caso podría estar relacionado con encuentros o experiencias intensas durante el verano. "Decir querer" indica que incluso en verano, la comunicación de afecto permanece central.

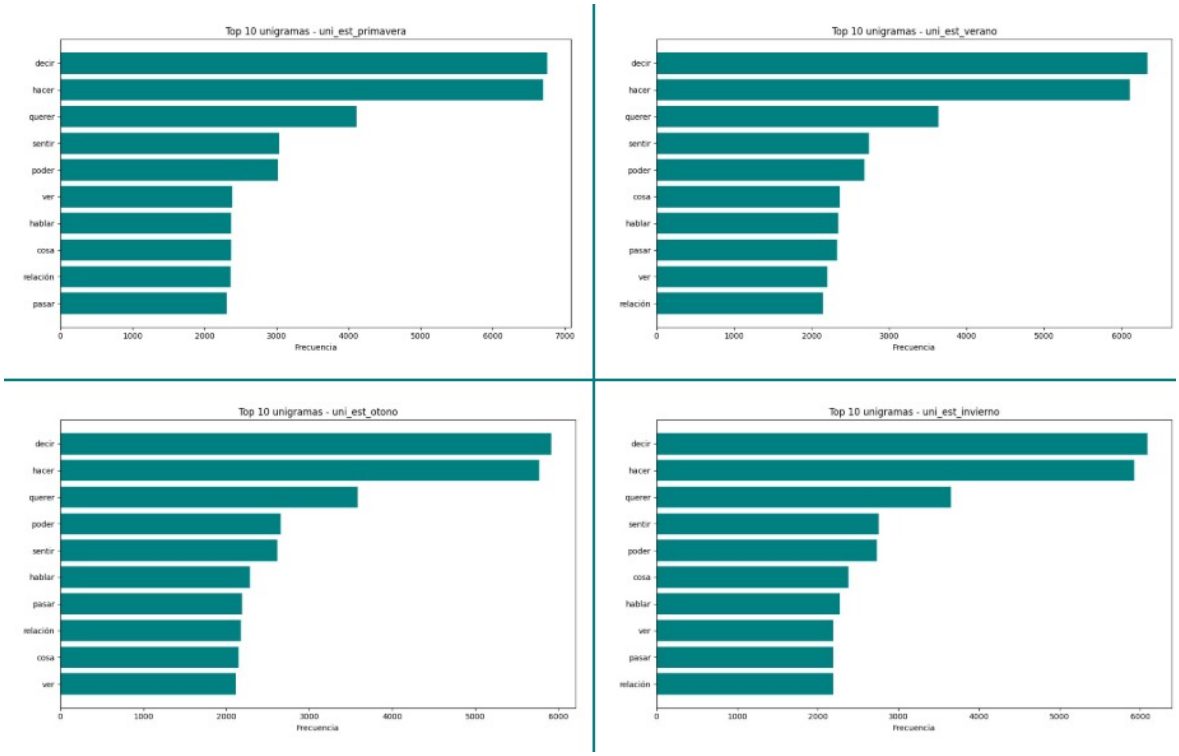

Figura 4.5: Frecuencia de los top 10 unigramas por estación. Elaboración propia

Los cinco bigramas predominantes en otoño son: hacer sentir (0.0800, 257), decir querer (0.0676, 257), sentir mal (0.0669, 239), poder hacer (0.0590, 201) y cada vez (0.0534, 180). “Hacer sentir” continua siendo el bigrama más relevante, por lo que podemos empezar a sospechar que no nos dará mucha información. En otoño, el tono en las publicaciones parece ser más reflexivo, introspectivo y evaluativo. El bigrama “decir querer” vuelve a aparecer, surgiendo nuevos bigramas de reconocimiento de malestares (“sentir mal”) y cuestionamiento de capacidades (“poder hacer”).

Por último, los cinco bigramas predominantes en verano son: hacer sentir (0.0827, 287), decir querer (0.0732, 283), poder hacer (0.0594, 180), primero vez (0.0592, 212) y sentir mal (0.0578, 215). “Hacer sentir” alcanza su máxima relevancia en invierno, indicando gran atención a cómo las acciones afectan emociones. Los bigramas reflejan una combinación de búsqueda de confort y reafirmación afectiva (“hacer sentir”, “decir querer”), reflexión sobre capacidad de acción (“poder hacer”), y atención a experiencias nuevas (“primero vez”) y malestares (“sentir mal”).

En general, los trigramas ofrecen mayor riqueza semántica, pero también aparecen con menor frecuencia. Siguiendo en esta línea, tiene sentido realizar un análisis con menor peso de frecuencia absoluta y mayor ponderación de TF-IDF. Durante la primavera los cinco trigramas con mayor relevancia han sido : “hacer sentir mal” (0.0312, 50), “hacer uno semana” (0.0161, 26), “querer tener relación” (0.0119, 22), “hacer cosa bien” (0.0107, 21), “decir querer hacer” (0.0118, 20). Indican introspección sobre malestares emocionales, referencias

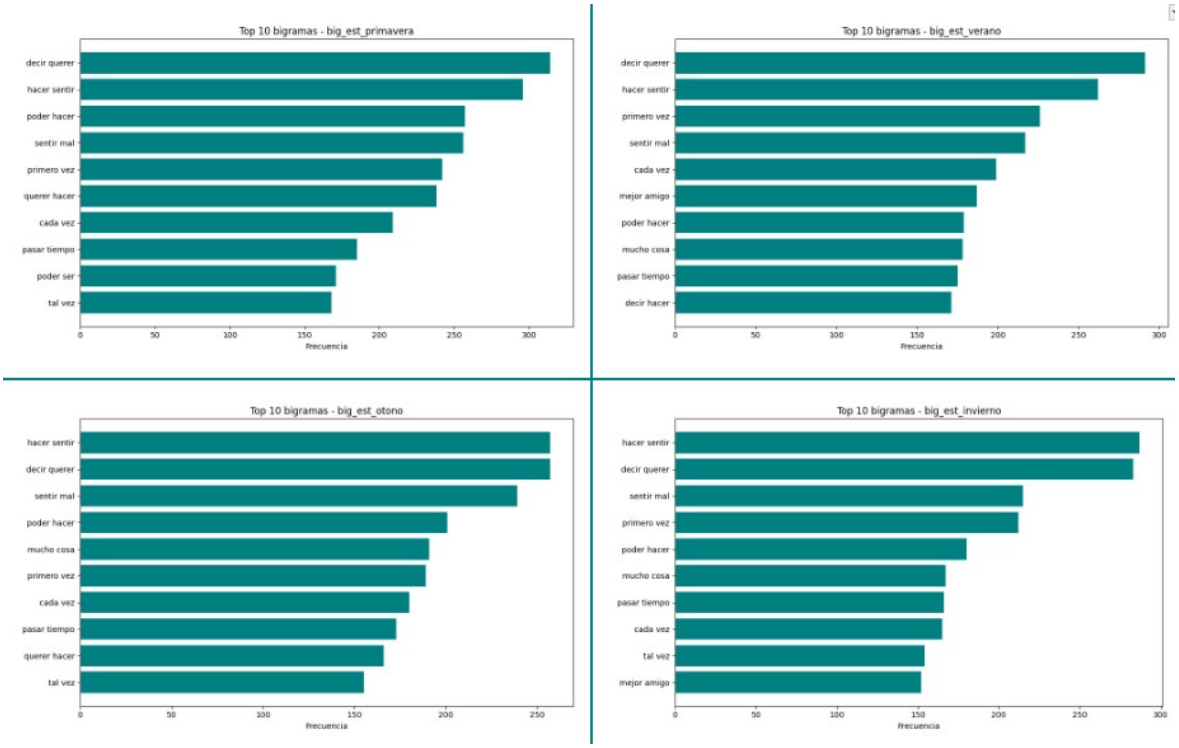

Figura 4.6: Frecuencia de los top 10 bigramas por estación. Elaboración propia

a eventos recientes, deseos de vínculo y atención a comportamientos adecuados al iniciar o revitalizar relaciones.

En verano, los top trigramas son “hacer sentir mal” (0.0383, 53), “hacer uno semana” (0.0181, 32), “querer tener relación” (0.0115, 17), “hacer mucho cosa” (0.0112, 17), “sentir mal decir” (0.0086, 17). Podemos confirmar de manera mucho más segura en comparación a los bigramas, que la percepción del amor en verano se ve influenciada por la exploración de nuevas experiencias y gestión de expectativas veraniegas.

Por otra parte, en otoño predominan “hacer sentir mal” (0.0318, 49), “hacer uno semana” (0.0211, 33), “poder dejar pensar” (0.0088, 17), “pasar mucho cosa” (0.0094, 16), “seguir ser amigo” (0.0086, 16). Efectivamente, los trigramas reflejan el tono introspectivo que hemos apreciado en los bigramas. Se centra en experiencias pasadas, transiciones relacionales y procesamiento de eventos.

Por último, en invierno los trigramas principales han sido “hacer sentir mal” (0.0305, 55), “hacer uno semana” (0.0175, 29), “poder dejar pensar” (0.0123, 21), “seguir ser amigo” (0.0096, 20), “pasar tiempo junto” (0.0103, 19). Podríamos justificar las intensificaciones emocionales de esta estación por las festividades y tal vez por el deseo de cercanía en época de cierre de año. Posiblemente “seguir ser amigo” indica que las conversaciones giran en torno a rupturas amorosas.

Como podemos ver los trigramas complementan los análisis realizados sobre los bigramas. Los trigramas son más ricos semánticamente y aportan información más clara. La com-

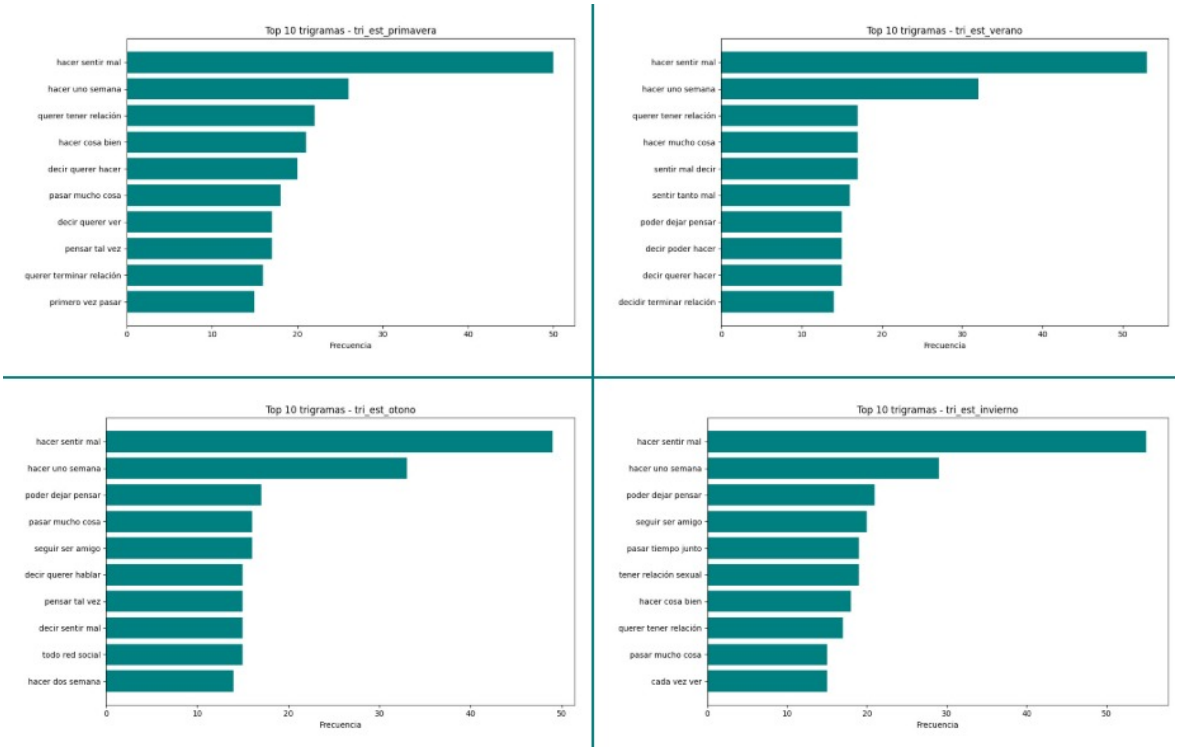

Figura 4.7: Frecuencia de los top 10 trigramas por estación. Elaboración propia

binación entre frecuencia absoluta y puntuación TF-IDF ha sido útil para detectar los textos que aportan más información en cada periodo. Este análisis previo aporta una buena base a partir de la cual podemos realizar el modelado de tópicos y el análisis de sentimiento.

4.4. Modelado de tópicos

El modelado de tópicos se ha aplicado para presentar cada una de las estaciones con sus principales temáticas de conversación en las publicaciones y comparar entre ellas. Siguiendo la estructura de procedimiento de la metodología, encontramos el número óptimo de tópicos para cada estación e identificamos qué temas relacionados con el amor emergen con mayor frecuencia y cómo cambian en función del contexto temporal. Metodológicamente, el uso de LDA aquí se alinea con Xu et al. (2024) y Göçen et al. (2024): la selección de número de tópicos basada en coherencia y la combinación con análisis de sentimiento.

Para este análisis se utilizó el modelo BERTopic que permite agrupar textos similares en clusters temáticos, aplicado independientemente a los textos de cada estación. Tras separar los datos por estación se exploraron los diferentes valores de tópicos con la metodología explicada en el capítulo anterior. La selección del número de tópicos es guiada por la coherencia temática observada en los términos clave. La Figura 4.8 resume el número óptimo de tópicos detectados para cada estación. Se han elegido 7 para primavera, 10 para verano, 3 para otoño y 9 para invierno.

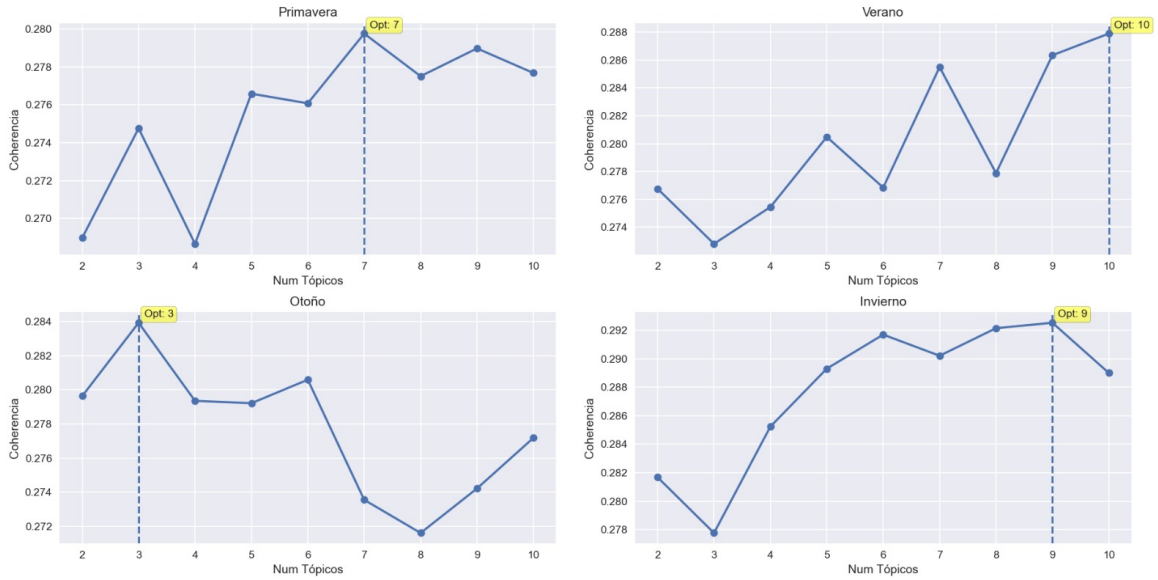

Figura 4.8: Número óptimo de tópicos por estación. Elaboración propia

En los resultados observamos que durante la primavera los temas son sobre comunicación y expresión emocional (“decir”, “querer”, “hablar”, “sentir”, “relación”) y presencia de otra persona (“él”, “amigo”, “ver”). La interpretación podría ser que durante la primavera el amor se percibe como una experiencia emocional interna en desarrollo en la que hay que hablar y expresar los sentimientos, lo que podría indicar una etapa de apertura y reflexión emocional. Esto confirma la teoría que establecen Wormley, Schaller y Varum (2023) al indicar que las temperaturas moderadas de primavera fomentan la sociabilidad. Los autores también destacan que la apertura emocional durante la primavera está alineada con la presencia de vegetación que disminuye el estrés.

Durante el verano los tópicos tornan más alrededor actividades y experiencias (“foto”, “ver”, “pasar”), relaciones sociales (“amigo”, “novio”, “novia”) e incluso sentimiento de deseo (“yo”, “querer”, “poder”). Como es lógico, el verano es la época social más activa y se relaciona con tener vivencias. Los vínculos sociales parecen tener el foco. Esto coincide con Wood, Varela, Bollen, Rocha y Gonçalves-Sá (2018), quienes encuentran que durante festividades culturales (muchas de las cuales caen en verano) hay picos de búsquedas relacionadas con el sexo, sugiriendo que las personas tienden a centrar su atención en la dimensión relacional y vivencial. Además, muestran que estos picos preceden aumentos en natalidad. El verano impulsa la conexión íntima y la búsqueda de experiencias compartidas. Está asociado con los social y vivencial, convirtiéndose en una estación activa y de exposición.

Por otra parte, el otoño parece caracterizarse más por ser reflexivo y nostálgico tras la intensidad del verano (“vida”, “relación”, “sentir”). Está dedicado a la introspección (“pensar”, “soledad”, “historia”) y a rupturas y recuerdos (“ex”, “pasado”, “problema”). Carrillo (2023) advierte sobre las expectativas idealizadas del amor que tanta frustración causan al enfrentar-

se a la realidad. Como el otoño es una temporada reflexiva, se da la toma de conciencia sobre mitos románticos. En otoño concentramos los tópicos en tan solo 3, lo cual tiene sentido al ser una época de introspección que se centra en pocos conceptos clave. Además, coincidimos con la recomendación de Göçen et al. (2024) de evaluar coherencia para evitar tópicos demasiado fragmentados o demasiado generales. Al tener menos temas podemos saber que el corpus otoñal discute de forma más dirigida.

Por último, en invierno parece predominar la necesidad de calor emocional y físico ("yo", "sentir", "frío"; "abrazo", "dormir", "cama"). En la misma línea la vulnerabilidad y la inseguridad emocional incrementan ("confianza", "preguntar", "extrañar"). Como confirman Wormley et al. (2023) la escasez de luz y las bajas temperatura incrementan estados de ánimo negativos y la necesidad de apoyo afectivo. También hablan del trastorno afectivo estacional, el cual da sentido a las dudas y expresiones de inseguridad. Desde la perspectiva de Carrillo (2023), en el invierno, al buscar refugio, surgen tensiones entre expectativas de "amor (romántico) como salvación" y los desafíos emocionales reales.

Resumiendo, se han identificado diferencias estacionales claras en el tono, la intención y la forma de experimentar y hablar del amor. La Figura 4.9 muestra un resúmen comparativo entre los tópicos principales de cada estación. Los meses más introspectivos son invierno y otoño, que se diferencian por que otoño es más llevado por la melancolía y la nostalgia mientras que en invierno se busca más una conexión emocional donde apoyarse. Primavera y verano, sin embargo, se caracterizan por la aventura y la energía, siendo primavera la etapa más romántica y emocional mientras que verano busca más intensidad e intimidad.

Estación	Enfoque emocional dominante	Tipo de conexión	Tópicos clave
Primavera	Apertura emocional, inicio de vínculos	Reflexiva y verbal	"decir", "querer", "relación"
Verano	Experiencias compartidas	Social y activa	"amigo", "foto", "ver", "yo"
Otoño	Nostalgia y balance	Introspectiva y melancólica	"ex", "vida", "sentir"
Invierno	Búsqueda de intimidad y refugio	Íntima y emocional	"abrazo", "frío", "extrañar"

Figura 4.9: Comparación de tópicos entre estaciones. Elaboración propia

4.5. Análisis de sentimiento

El último paso del análisis consiste en un análisis de sentimiento, en el que se incluirán análisis descriptivos, comparaciones entre estaciones, tendencias temporales, análisis de extremos y características lingüísticas. Tal como se observa en estudios similares de análisis

de sentimiento en redes sociales, la mayoría de las publicaciones relacionadas con el amor tienden a expresar emociones moderadas o equilibradas, sin inclinaciones extremas positivas o negativas (ver Figura 4.10). Se observa una asimetría leve hacia valores negativos, resaltando la diferencia entre positivos y negativos en invierno (ver Figura 4.11). Se confirma, una vez más, la teoría de Wormley, Schaller y Varum (2023) sobre las emociones negativas en invierno.

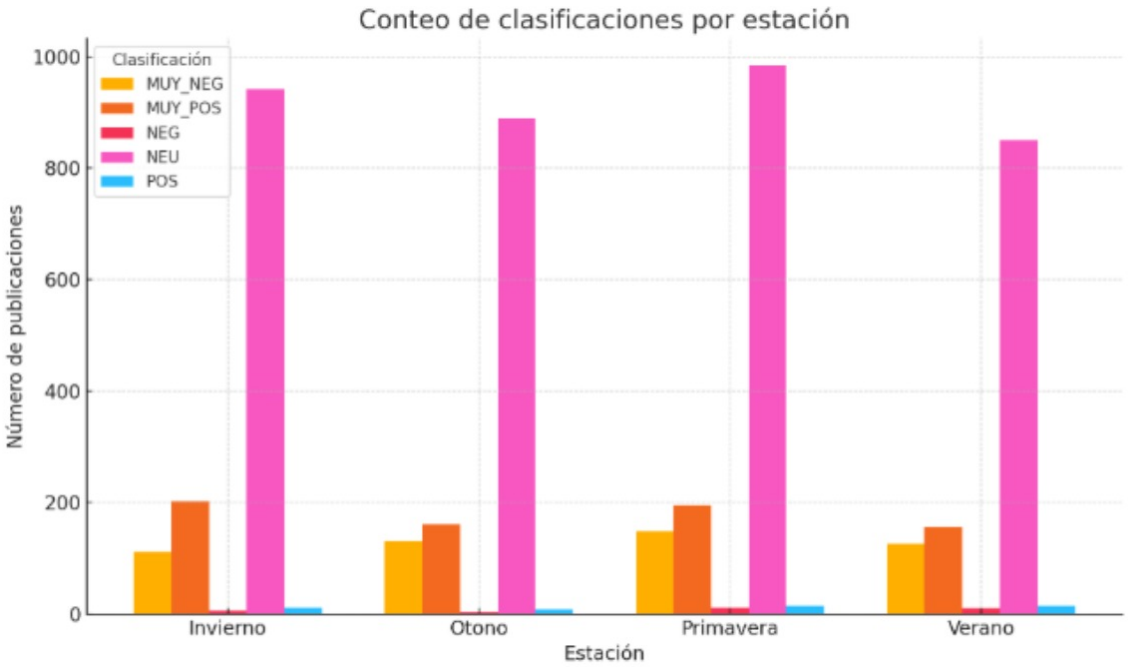

Figura 4.10: Conteo de clasificaciones por estación. Elaboración propia

Las cuartillas y rangos intercuartílicos muestran ligeras variaciones estacionales, pero ninguna estación presenta una dispersión notablemente distinta de las demás, indicando una relativa estabilidad estacional en la polaridad emocional. Se percibe una ligera tendencia a valores más negativos en verano e invierno en comparación con primavera, aunque las diferencias centrales no son drásticas.

Para cada estación se graficó la distribución de valores de final (ver Figura . Todas ellas exhiben un pico principal alrededor de cero con colas levemente pronunciadas hacia valores negativos en verano e invierno en comparación con primavera y otoño. El análisis de sentimiento se ve limitado por una falta de diferenciación entre los sentimientos expresados en las diferentes estaciones.

Como conclusión del análisis de sentimiento se puede decir que predomina el sentimiento neutro y no hay inclinaciones extremas. Sí se puede apreciar una inclinación leve hacia lo negativo aunque la mediana cercana a cero sugiere neutralidad. El sentimiento negativo se observa más en verano y en inverno. En verano se puede explicar con experiencias negativas de desamor y el calor extremo. En invierno entre que los días son más cortos y es una época más introspectiva, se podría explicar que se compartieran más pensamientos negativos.

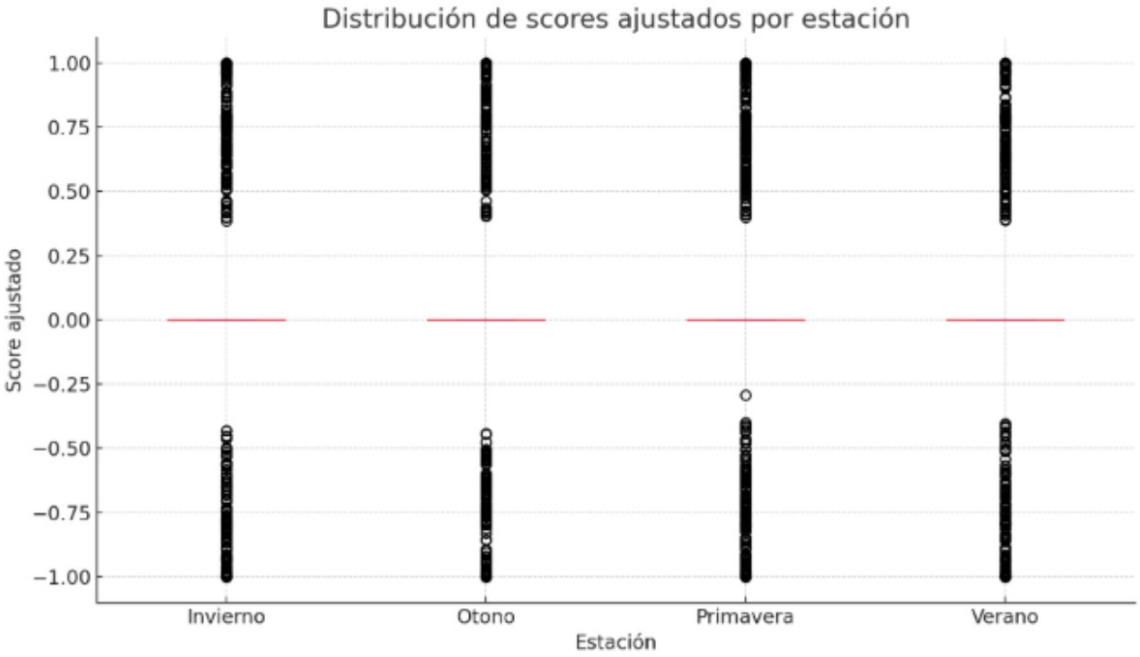

Figura 4.11: Histograma global de scores ajustados. Elaboración propia

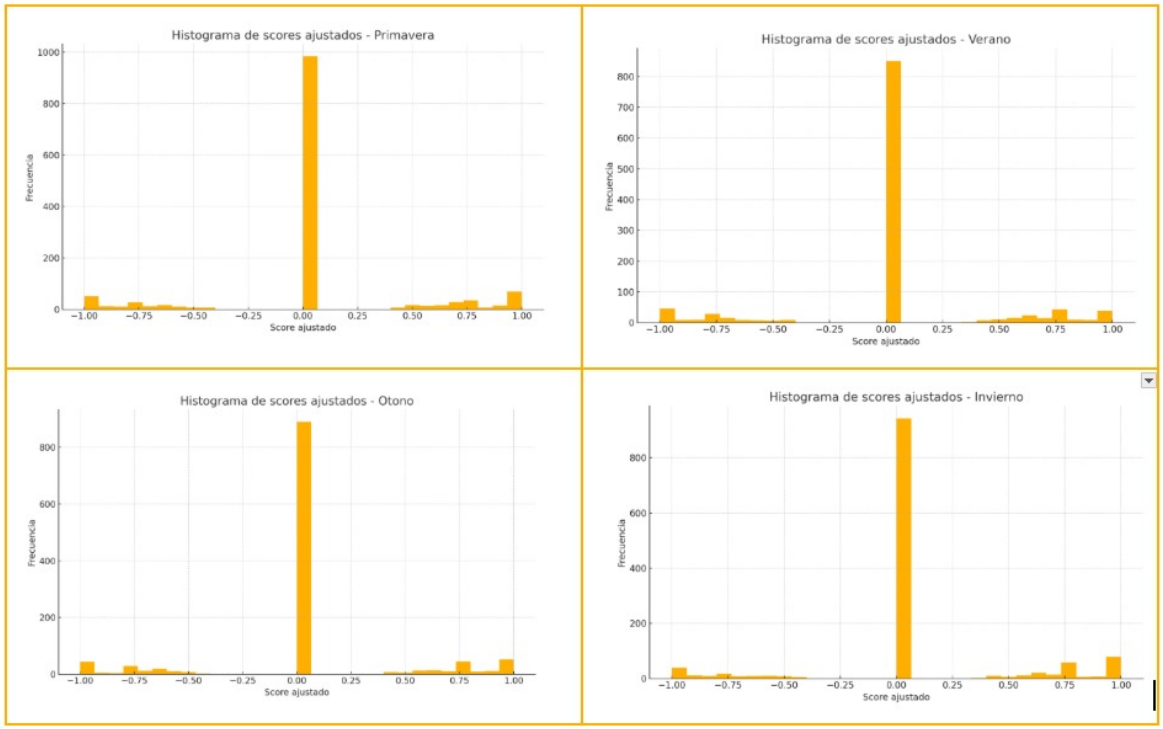

Figura 4.12: Distribución de valores de final. Elaboración propia

Raramente se han empleado puntuaciones enfáticas y repeticiones, parece que las exageraciones no dominan la conversación. Los elementos lingüísticos destacados son las negaciones y los conectores, indicando que los usuarios frecuentemente usan estructuras complejas para matizar sus expresiones sobre el amor.

La escasa presencia de “muy positivo” sugiere que las expresiones de amor extremadamente positivas no son comunes en Reddit. Sin embargo, no podemos descartar que haya picos de éstas, puesto que también tiene sentido que se compartan más en entornos privados y a personas fuera de la plataforma.

Capítulo 5

Conclusiones

De este estudio podemos concluir que es posible que existan cambios en las percepciones y expectativas en el amor dependiendo del contexto temporal. No obstante, existen ciertas limitaciones a este estudio, por lo que no podemos afirmarlo de manera segura. La limitación más grande de este estudio ha sido que lo más probable es que la mayoría de publicaciones provengan de un público más jóven y experimentado en las redes sociales, lo que supone un problema de representatividad y sesga nuestro estudio. Existe la posibilidad de que la percepción del amor no solo varíe a lo largo del año sino también entre generaciones. Además, Reddit es conocida por ser una red social especialmente habitada por el género masculino, lo cual supone otro problema de representatividad.

Las combinaciones léxicas más relevantes (n-gramas) se mantuvieron bastante estables entre estaciones, lo que sugiere que la forma en la que los usuarios hablan del amor es bastante constante. Sin embargo, las variaciones sutiles en trigramas y ciertos términos permiten detectar matices específicos en cada periodo. Mientras que la primavera se relaciona con apertura emocional y reflexión, el verano resalta por experiencias sociales y deseos, el otoño por introspección y nostalgia, y el invierno por vulnerabilidad y búsqueda de refugio afectivo. Esto refuerza la hipótesis de que factores ecológicos y sociales estacionales influyen en la forma en que las personas experimentan y verbalizan sus emociones amorosas. Se pudieron apreciar temáticas que luego consolidamos con el uso de un modelado de tópicos. Los n-gramas y el modelado de tópicos han estado extremadamente relacionados en este estudio, reflejando similitudes y apoyándose mutuamente. No obstante, el análisis de sentimiento no ha sido de tanta utilidad.

El modelado de tópicos ha resultado más útil que en el análisis de sentimiento para sacar conclusiones sobre las variaciones en la percepción del amor durante las diferentes estaciones del año. Al haber tantas observaciones clasificadas con sentimiento neutro y tan pocas polarizadas, no se ha considerado útil generalizar el sentimiento de cada época. El análisis de sentimiento reveló que la mayoría de publicaciones presentan sentimientos neutros, con una ligera inclinación hacia lo negativo, especialmente en invierno y verano. La baja frecuencia

de sentimientos muy positivos sugiere que las redes sociales, en especial Reddit, no son un espacio habitual para compartir expresiones de afecto altamente idealizadas o entusiastas. Podría decirse que lo negativo toma más peso que lo positivo en esta red social, ya que las conversaciones en las que se ha hallado sentimiento eran prácticamente todas negativas. No parece haber interés en compartir experiencias positivas. Parece que Reddit (en términos del amor romántico) se emplea más como plataforma de apoyo en momentos de necesidad emocional. El análisis de sentimiento también muestra cómo las estructuras lingüísticas, especialmente las expresiones con conectores de contraste y negaciones, revelan que los usuarios no sólo sienten amor, sino que lo cuestionan, lo resignifican y lo problematizan. Se evidencia un amor conversado, en diálogo interno constante, más que un amor impulsivo o simplemente instintivo. Además, se observó una presencia frecuente de conectores de contraste y negaciones, lo que indica que los usuarios tienden a matizar sus emociones de forma elaborada y no polarizada.

Como futura línea de investigación se plantea continuar este estudio fuera de las redes sociales, puesto que se han hallado pocas expresiones "muy positivas" en la plataforma Reddit. Parece que los foros de conversación online se polarizan hacia cuestiones negativas y problemas para buscar ayuda en situaciones difíciles. También sería interesante investigar cómo la pandemia ha modificado la percepción del amor a través de la redes sociales, especialmente las nuevas formas de vinculación digital emergentes en periodos de crisis global.

En definitiva, este trabajo ha demostrado que el amor no solo se vive, también se escribe, se analiza y se transforma. Las estaciones cambian, pero la necesidad de entender el amor permanece constante; y hoy, ese entendimiento también pasa por los algoritmos y las palabras compartidas en la red.

Declaración de Uso de Herramientas de Inteligencia Artificial Generativa en Trabajos Fin de Grado

ADVERTENCIA: Desde la Universidad consideramos que *ChatGPT* u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Leire Reneses Rodríguez, estudiante de E6-Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "Análisis de Percepciones y Sentimientos sobre la Concepción del Amor a través de Reddit", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa *ChatGPT* u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. Brainstorming de ideas de investigación: Utilizado para idear y esbozar posibles áreas de investigación.
2. Metodólogo: Para descubrir métodos aplicables a problemas específicos de investigación.
3. Interpretador de código: Para realizar análisis de datos preliminares.
4. Corrector de estilo literario y de lenguaje: Para mejorar la calidad lingüística y estilística del texto.
5. Sintetizador y divulgador de libros complicados: Para resumir y comprender literatura compleja.
6. Revisor: Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
7. Traductor: Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han

dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado *ChatGPT* u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 18 de Junio de 2025

Firma: Leire Reneses (50636006B)

Capítulo 6

Referencias

Álvarez, P. C., & De Baranda Andújar, C. S. (2018). *Percepción del amor romántico en adolescentes y papel de los medios de comunicación*. Documents - Universidad Complutense de Madrid. https://produccioncientifica.ucm.es/documentos/6381690f18a84b178fea98a6

Blanco, R., Parra, Á., Salado, V., & Díez, M. (2024). *Vista de Creencias sobre el amor y bienestar durante la adultez emergente*. https://www.apuntesdepsicologia.es/index.php/revista/article/view/1568/1388

Carrillo L, V. G. (2025). *Para entender el amor romántico*. https://www.researchgate.net/publication/374200108_Para_entender_el_amor_romantico

Cifuentes Quintero, J. A. & Cátedra Santalucía de Analytics for Education. (2024, 29 abril). *Introducción al Modelado de Tópicos 1* [Vídeo]. YouTube. https://www.youtube.com/watch?v=13WNXj45yZU

Davidson, C. (2023). *Use of Reddit for Social Science Research: A Review of Current Use, Exploration of Potential Sampling Error, and Practical Demonstration Using Reddit to Study Post-Pandemic Teacher Resignation*. ScholarWorks. https://scholarworks.wmich.edu/dissertations/3936/

Feldhege, J., Moessner, M., & Bauer, S. (2019). *Who says what? Content and participation characteristics in an online depression community*. https://pubmed.ncbi.nlm.nih.gov/31780138/

Gallego Granero, E., & Fernández Piedra, D. (2023). *Percepciones del Amor en Población Adulta-jóven Madrileña*. https://rua.ua.es/dspace/bitstream/10045/136442/6/OBETS_18_2_04.pdf

Göçen, A., Ibrahim, M. M., & Khan, A. U. I. (2024). Public attitudes toward higher education using sentiment analysis and topic modeling. *Discover Artificial Intelligence*, *4*(1). https://doi.org/10.1007/s44163-024-00195-4

Harmat, M. G., D Well, A., Overtree, C., & Kawamura, K. (2019). *Seasonal Variation of Depression and Other Moods: A Longitudinal Approach*. https://www.researchgate.net/publication/12379021_Seasonal_Variation_of_Depression_and_Other_Moods_A_Longitudinal_Approach

Hohm, I., Wormley, A. S., Schaller, M., & Varnum, M. E. W. (2023). Homo temporus: Seasonal Cycles as a Fundamental Source of Variation in Human Psychology. *Perspectives On Psychological Science*. https://doi.org/10.1177/17456916231178695

Holman, L., Head, M. L., Lanfear, R., & Jennions, M. D. (2015). Evidence of Experimental Bias in the Life Sciences: Why We Need Blind Data Recording. *PLoS Biology*, *13*(7), e1002190. https://doi.org/10.1371/journal.pbio.1002190

K. Coffey, J., Shahvali, M., Kerstetter, D., & Aron, A. (2022). *Couples vacations and romantic passion and intimacy*. https://pure.buas.nl/files/32706748/Shahvali_couples_vacations_and_romantic_passion_and_intimacy.pdf

KS, R. (2024). *Analyzing Online Conversations on Reddit: A Study of Stress and Anxiety Through Topic Modeling and Sentiment Analysis*. https://www.researchgate.net/publication/384839617_Analyzing_Online_Conversations_

on_Reddit_A_Study_of_Stress_and_Anxiety_Through_Topic_Modeling_and_Sentiment_Analysis

Medvedev, A., & Lambiotte, R. (2019). *The Anatomy of Reddit: An Overview of Academic Research*. https://www.researchgate.net/publication/333085087_The_Anatomy_of_Reddit_An_Overview_of_Academic_Research

Merlyn Sacoto, M.-F., Moreta-Herrera, R., & Jayo, L. (2020). *Percepciones sobre amor, compromiso, fidelidad y pareja en jóvenes universitarios de Quito*. https://revistas.unlp.edu.ar/revpsi/article/download/9464/9156/32052

Prats, A. (2024, 28 enero). *¿Qué dice La Teoría Triangular del Amor de Sternberg? - AEPSIS*. AEPSIS. https://www.aepsis.com/que-dice-la-teoria-triangular-del-amor-de-sternberg/

Rama Kiran Garimella, V., Weber, I., & Dal Cin, S. (2014). *From "I Love You Babe" to "Leave Me Alone" - Romantic Relationship Breakups on Twitter*. https://www.researchgate.net/publication/265967108_From_I_Love_You_Babe_to_Leave_Me_Alone_-_Romantic_Relationship_Breakups_on_Twitter

Román Fernández, S. (2025). *¿Qué entendemos por amor? Un análisis sociológico de las percepciones del amor romántico entre los jóvenes de la sociedad moderna*. https://www.socyl.es/revistas/index.php/revista-socyl/article/download/47/23

Wang, H. M., Bulat, B., Fujimoto, S., & Frey, S. (2022, 11 junio). *Governing for Free: Rule Process Effects on Reddit Moderator Motivations*. arXiv.org. https://arxiv.org/abs/2206.05629

Wellings, K., Macdowall, W., & Goodrich, J. (1998). *Seasonal variations in sexual activity and their implications for sexual health promotion*.

https://www.researchgate.net/publication/12850059_Seasonal_variations_in_sexual_activity_and_their_implications_for_sexual_health_promotion

Wood, I. B., Varela, P. L., Bollen, J., Rocha, L. M., & Gonçalves-Sá, J. (2017). Human Sexual Cycles are Driven by Culture and Match Collective Moods. *Scientific Reports*, *7*(1). https://doi.org/10.1038/s41598-017-18262-5

Xu, Z., Fang, Q., Huang, Y., & Xie, M. (2024). The public attitude towards ChatGPT on reddit: A study based on unsupervised learning from sentiment analysis and topic modeling. *PLoS ONE*, *19*(5), e0302502. https://doi.org/10.1371/journal.pone.0302502

Anexo

Código de descarga de datos:

```
import time
import datetime
import pickle
import praw
import os
from collections import defaultdict

# Registro
def registrar(mensaje):
    print(f"[{datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')}]
{mensaje}")

# Cliente Reddit
def configurar_cliente_reddit(id_cliente, secreto_cliente,
agente_usuario):
    return praw.Reddit(client_id=id_cliente,
                       client_secret=secreto_cliente,
                       user_agent=agente_usuario)

# Barra de progreso
def barra_progreso(i, total, prefijo=''):
    porcentaje = int(100 * i / total)
    barra = '#' * (porcentaje // 2) + '-' * (50 - porcentaje // 2)
    print(f"{prefijo} |{barra}| {porcentaje}%", end='\r')
    if i == total: print()

# Palabras clave
def contiene_palabras_clave(texto, palabras_clave):
    texto = (texto or '').lower()
    return any(palabra.lower() in texto for palabra in palabras_clave)

# Punto de control
def guardar_punto_control(estado, archivo):
    with open(archivo, 'wb') as f:
        pickle.dump(estado, f)

def cargar_punto_control(archivo):
```

```
    return pickle.load(open(archivo, 'rb')) if os.path.exists(archivo)
else None

# Descarga de publicaciones
def extraer_publicaciones(reddit, nombre_subreddit, palabras_clave,
fecha_limite, objetivo):
    subreddit = reddit.subreddit(nombre_subreddit)
    recolectadas = {}
    extendidas = palabras_clave + [
        "pareja", "relación", "relaciones", "amor", "novio", "novia",
        "esposa", "marido", "matrimonio", "cita", "citas", "dating",
        "salir", "ligar", "enamorar", "crush", "tinder", "badoo",
        "soltera", "soltero", "ex", "ruptura", "separación",
        "boda", "casarse", "compromiso", "aniversario",
        "san valentín", "valentine", "corazón", "querer",
        "conocer", "busco", "quedada", "rollo", "ligue", "novia", "novio",
"pareja", 'sexo', 'relación', 'amor', 'ruptura', 'celos', 'enamorado',
'ligar'
    ]
    fases = [
        ('buscar', [('relevance','all'),('new','all'),('hot','month')]),
        ('listar', ['new','hot','rising','controversial']),
        ('cronológico', None),
        ('autores', None),
    ]
    for nombre_fase, configuracion in fases:
        if len(recolectadas) >= objetivo: break
        registrar(f"FASE {nombre_fase.upper()} - {len(recolectadas)}
publicaciones")
        if nombre_fase == 'buscar':
            for metodo, filtro_tiempo in configuracion:
                for palabra in extendidas:
                    for pub in subreddit.search(palabra, sort=metodo,
time_filter=filtro_tiempo, limit=10000):
                        if pub.created_utc >= fecha_limite.timestamp():
                            recolectadas[pub.id] = pub
        elif nombre_fase == 'listar':
            for metodo in configuracion:
                if metodo in ['controversial', 'top']:
```

```
                    publicaciones = getattr(subreddit,
metodo)(limit=10000, time_filter='all')
                else:
                    publicaciones = getattr(subreddit,
metodo)(limit=10000)
                for pub in publicaciones:
                    if pub.created_utc >= fecha_limite.timestamp() and
contiene_palabras_clave(pub.title + pub.selftext, extendidas):
                        recolectadas[pub.id] = pub
        elif nombre_fase == 'cronológico':
            despues = None
            while len(recolectadas) < objetivo:
                publicaciones = list(subreddit.new(limit=10000,
params={'after':despues} if despues else {}))
                if not publicaciones: break
                for pub in publicaciones:
                    if pub.created_utc >= fecha_limite.timestamp() and
contiene_palabras_clave(pub.title + pub.selftext, extendidas):
                        recolectadas[pub.id] = pub
                despues = publicaciones[-1].name
        else:  # autores
            autores_top = sorted(
                defaultdict(int, {pub.author.name: 1 for pub in
recolectadas.values() if pub.author}).items(),
                key=lambda x: -x[1])[:5]
            for autor, _ in autores_top:
                for pub in
reddit.redditor(autor).submissions.new(limit=10000):
                    if pub.subreddit.display_name == nombre_subreddit and
pub.created_utc >= fecha_limite.timestamp():
                        recolectadas[pub.id] = pub
    return list(recolectadas.values())

# Procesamiento y guardado final
def procesar_y_guardar(publicaciones, fecha_limite):
    datos = []
    for i, pub in enumerate(publicaciones, 1):
        barra_progreso(i, len(publicaciones), prefijo='Procesando')
        if pub.created_utc < fecha_limite.timestamp(): continue
        datos.append({
```

```
                    'Título': pub.title,
                    'Fecha': pub_date,
                    'URL': pub.url,
                    'Texto': pub.selftext,
                    'Subreddit': pub.subreddit.display_name,
                    'Autor': pub.author.name if pub.author else "Deleted",
                    'pub_ID': pub.id,
                    'Score': pub.score,
                    'Num_Comments': pub.num_comments,
                    'Created_UTC': pub.created_utc,
                    'Es_Video': pub.is_video,
                    'Es_Imagen': pub.url.endswith(('.jpg', '.jpeg',
'.png', '.gif')),
                    'Flair': pub.link_flair_text if hasattr(pub,
'link_flair_text') else None,
                    'Upvote_Ratio': getattr(pub, 'upvote_ratio', None),
                    'Es_Self': pub.is_self,
                    'Permalink': f"https://reddit.com{pub.permalink}"
        })
    df =
__import__('pandas').DataFrame(datos).drop_duplicates('ID_Publicación')
    ahora = datetime.datetime.now().strftime('%Y%m%d_%H%M')
    salida = f"reddit_{len(df)}publicaciones_{ahora}.csv"
    df.to_csv(salida, index=False)
    registrar(f"Guardado {salida} con {len(df)} publicaciones")

# Principal
def main():

    credenciales_reddit = {
        'id_cliente': 'LOdUzAgzlqoLbE_eTJBkgQ',
        'secreto_cliente': 'mzHmfsD9YEkAMobS1WL9BGKh3pYQgw',
        'agente_usuario': 'App_Love'
    }
    palabras_clave = ['novia', 'novio', 'pareja', 'sexo', 'relación',
'amor', 'ruptura', 'celos', 'enamorado', 'ligar']
    subreddit = 'relaciones'
    dias = 10000
    fecha_limite = datetime.datetime.now() - datetime.timedelta(days=dias)
    reddit = configurar_cliente_reddit(**credenciales_reddit)
```

```
    publicaciones = extraer_publicaciones(reddit, subreddit,
palabras_clave, fecha_limite, objetivo=10000)
    procesar_y_guardar(publicaciones, fecha_limite)

if __name__ == '__main__':
    main()
```

Código de preprocesamiento:

```
import os
import re
import pandas as pd
from collections import Counter
from langdetect import detect
from langdetect.lang_detect_exception import LangDetectException
import nltk
from nltk import download
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import spacy
import warnings

warnings.filterwarnings('ignore')

# Descargar recursos de NLTK si no están
for r in ['punkt', 'stopwords']:
    try:
        nltk.data.find(f"tokenizers/{r}" if r=='punkt' else
f"corpora/{r}")
    except:
        download(r)

# Modelo spaCy
nlp = spacy.load('es_core_news_sm')

# Stopwords
sw_esp = set(stopwords.words('spanish'))
sw_bas = set('si sí no más muy solo también ya http https www'.split())
sw_temp = set('ayer hoy mañana años meses días segundos'.split())
all_sw = sw_esp | sw_bas | sw_temp
```

```
# Claves de tema
rel_claves = ['relación','sexo','amor','novio','novia','boda']

MIN_LEN = 3
MAX_RATIO = 0.95

# Quita solo stopwords
def rm_stop(text):
    if pd.isna(text): return ''
    parts = re.findall(r"\b\w+\b|\W+", text)
    out = []
    for p in parts:
        if re.match(r"\w+", p) and p.lower() in all_sw:
            continue
        out.append(p)
    return ' '.join(''.join(out).split())

# Detecta español
def is_es(text):
    t = str(text)
    if len(t)<10: return False
    try:
        return detect(t)=='es'
    except:
        return False

# Filtra tema
def has_rel(text):
    t = str(text).lower()
    return any(k in t for k in rel_claves)

# Tokeniza y limpia/lematiza
def proc_text(text):
    toks = word_tokenize(str(text), 'spanish')
    doc = nlp(' '.join(toks))
    lem = []
    for tok in doc:
        w = tok.text.lower()
        if (re.match(r'https?://',w) or w in all_sw or len(w)<MIN_LEN
```

```
            or not re.match(r'^[a-záéíóúñü]+$',w)):
            continue
        lem.append(tok.lemma_)
    return lem

class Preproc:
    def __init__(self, in_file, out_file=None):
        self.in_file = in_file
        base = os.path.splitext(in_file)[0]
        self.out_file = out_file or base + '_out.xlsx'
        self.df = None

    def load(self):
        self.df = pd.read_excel(self.in_file)
        if 'Post_ID' not in self.df:
            self.df['Post_ID'] = self.df.index
        self.df.set_index('Post_ID', inplace=True)

    def dedup(self):
        self.df = self.df.drop_duplicates('Texto')

    def filter(self):
        self.df = self.df[self.df['Texto'].apply(is_es)]
        self.df = self.df[self.df['Texto'].apply(has_rel)]

    def tokens(self):
        self.df['Tokens'] = self.df['Texto'].apply(proc_text)
        self.df['NoStop'] = self.df['Texto'].apply(rm_stop)

    def clean(self):
        self.df = self.df[self.df['Tokens'].map(len)>0]
        total = len(self.df)
        all_t = [t for lst in self.df['Tokens'] for t in set(lst)]
        cnt = Counter(all_t)
        bad = {w for w,c in cnt.items() if c/total>MAX_RATIO or c<2}
        self.df['FinalTok'] = self.df['Tokens'].apply(lambda l: [w for w
in l if w not in bad])
        self.df['Texto_Final'] = self.df['FinalTok'].apply(lambda l: '
'.join(l))
        self.df = self.df[self.df['Texto_Final']!='']
```

```
    def save(self):
        self.df.to_excel(self.out_file)

    def run(self):
        self.load()
        self.dedup()
        self.filter()
        self.tokens()
        self.clean()
        self.save()

if __name__=='__main__':
    p = Preproc(r"C:\Users\archivos_unidos_limpio.xlsx")
    p.run()
```

Código de gráficos:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import re
from datetime import datetime
import numpy as np

plt.style.use('default')
sns.set_palette("husl")

def load_data(file_path):
    df = pd.read_excel(file_path)
    return df

# Preprocesamiento de texto
def preprocess_text(text):
    if pd.isna(text):
        return []

    text = str(text).lower()
```

```
    text = re.sub(r'[^a-záéíóúñü\s]', '', text)

    words = text.split()

    stop_words = {'de', 'la', 'el', 'en', 'y', 'a', 'que', 'es', 'se',
'no', 'te', 'lo', 'le',
                  'da', 'su', 'por', 'son', 'con', 'para', 'una', 'al',
'del', 'las', 'los',
                  'un', 'como', 'me', 'mi', 'si', 'ya', 'ni', 'más',
'muy', 'ser', 'han',
                  'and', 'the', 'to', 'of', 'in', 'for', 'is', 'on',
'that', 'by', 'this',
                  'with', 'i', 'you', 'it', 'not', 'or', 'be', 'are',
'from', 'at', 'as',
                  'your', 'all', 'any', 'can', 'had', 'her', 'was', 'one',
'our', 'out',
                  'day', 'get', 'has', 'him', 'his', 'how', 'man', 'new',
'now', 'old',
                  'see', 'two', 'way', 'who', 'boy', 'did', 'its', 'let',
'put', 'say',
                  'she', 'too', 'use'}

    words = [word for word in words if len(word) > 2 and word not in
stop_words]

    return words

import matplotlib.pyplot as plt
import pandas as pd

def create_timeline_plot(df, date_column='Fecha'):
    plt.figure(figsize=(12, 6))

    df[date_column] = pd.to_datetime(df[date_column])

    monthly_counts = df.groupby(df[date_column].dt.to_period('M')).size()
    periods = monthly_counts.index.to_timestamp()

    bars = plt.bar(range(len(monthly_counts)), monthly_counts.values,
                   color='steelblue', alpha=0.7, edgecolor='black')
```

```
    plt.title('Evolución de publicaciones por mes', fontsize=14,
fontweight='bold')
    plt.xlabel('Año', fontsize=12)
    plt.ylabel('Número de publicaciones', fontsize=12)

    x_labels = []
    prev_year = None
    for date in periods:
        year = date.year
        if year != prev_year:
            x_labels.append(str(year))
            prev_year = year
        else:
            x_labels.append("")

    plt.xticks(ticks=range(len(x_labels)), labels=x_labels, rotation=0,
ha='center')



    plt.grid(True, alpha=0.3, axis='y')
    plt.tight_layout()
    plt.show()


def create_score_histogram(df, upvote_ratio_column='Upvote_Ratio'):
    plt.figure(figsize=(10, 6))

    upvote_ratios = df[upvote_ratio_column].dropna()
    upvote_ratios = upvote_ratios[(upvote_ratios >= 0) & (upvote_ratios <=
1)]

    plt.hist(upvote_ratios, bins=30, alpha=0.7, color='skyblue',
edgecolor='black')
    plt.title('Distribución de Upvote Ratio (engagement))', fontsize=14,
fontweight='bold')
    plt.xlabel('Upvote Ratio', fontsize=12)
    plt.ylabel('Frecuencia', fontsize=12)
    plt.grid(True, alpha=0.3)
```

```
    mean_ratio = upvote_ratios.mean()
    median_ratio = upvote_ratios.median()
    plt.axvline(mean_ratio, color='red', linestyle='--', label=f'Media:
{mean_ratio:.3f}')
    plt.axvline(median_ratio, color='orange', linestyle='--',
label=f'Mediana: {median_ratio:.3f}')
    plt.legend()

    plt.tight_layout()
    plt.show()

def create_top_words_table(df, text_column='Texto', top_n=20):
    all_words = []

    for text in df[text_column].dropna():
        words = preprocess_text(text)
        all_words.extend(words)

    word_counts = Counter(all_words)
    top_words = word_counts.most_common(top_n)

    words_df = pd.DataFrame(top_words, columns=['Palabra', 'Frecuencia'])
    words_df['Ranking'] = range(1, len(words_df) + 1)
    words_df = words_df[['Ranking', 'Palabra', 'Frecuencia']]

    print("Top 15 palabras más frecuente")
    print("=" * 60)
    print(words_df.to_string(index=False))

    plt.figure(figsize=(10, 8))
    words = [item[0] for item in top_words[:15]]
    counts = [item[1] for item in top_words[:15]]

    plt.barh(range(len(words)), counts, color='lightcoral')
    plt.yticks(range(len(words)), words)
    plt.xlabel('Frecuencia', fontsize=12)
    plt.title('Top 15 palabras más frecuentes', fontsize=14,
fontweight='bold')
    plt.gca().invert_yaxis()
```

```
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    return words_df

def create_subreddit_bar_chart(df, subreddit_column='Subreddit',
top_n=15):
    plt.figure(figsize=(12, 8))

    subreddit_counts = df[subreddit_column].value_counts().head(top_n)

    bars = plt.bar(range(len(subreddit_counts)), subreddit_counts.values,
                   color='lightgreen', edgecolor='black')

    plt.title('Subreddits más frecuentes', fontsize=14, fontweight='bold')
    plt.xlabel('Subreddit', fontsize=12)
    plt.ylabel('Número de publicaciones', fontsize=12)
    plt.xticks(range(len(subreddit_counts)), subreddit_counts.index,
rotation=45, ha='right')
    plt.grid(True, alpha=0.3, axis='y')

    for bar, value in zip(bars, subreddit_counts.values):
        plt.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 0.1,
                 str(value), ha='center', va='bottom', fontsize=10)

    plt.tight_layout()
    plt.show()

def main():
    file_path = r"C:\UserS\archivos_unidos.xlsx"

    try:
        # Cargar datos
        print("Cargando datos...")
        df = pd.read_excel(file_path)
        print(f"Datos cargados: {len(df)} filas, {len(df.columns)}
columnas")
        print(f"Columnas disponibles: {list(df.columns)}")
```

```
        create_timeline_plot(df, 'Fecha')

        create_score_histogram(df, 'Upvote_Ratio')

        words_table = create_top_words_table(df, 'Texto')

        create_subreddit_bar_chart(df, 'Subreddit')


    except FileNotFoundError:
        print(f"Error: No se encontró el archivo {file_path}")

    except Exception as e:
        print(f"Error: {e}")

# Ejecutar análisis
if __name__ == "__main__":
    main()
```

Código n-gramas:

```
import pandas as pd
import numpy as np
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer
import warnings

warnings.filterwarnings('ignore')

class AnalizadorNGramasTiempo:
    def __init__(self, archivo_excel, col_texto):
        self.archivo = archivo_excel
        self.col_texto = col_texto
        self.df = None
        self.res_temporales = {}

        self.estaciones = {
            'primavera': [3,4,5],
            'verano':    [6,7,8],
            'otono':     [9,10,11],
```

```
            'invierno':  [12,1,2]
        }

    def cargar(self):
        self.df = pd.read_excel(self.archivo, engine='openpyxl')
        self.df['Fecha'] = pd.to_datetime(self.df['Fecha'],
errors='coerce')
        self.df.dropna(subset=['Fecha', self.col_texto], inplace=True)
        self.df['mes'] = self.df['Fecha'].dt.month
        self.df['dia'] = self.df['Fecha'].dt.day
        self.df['estacion'] = self.df['mes'].apply(self._estacion)

    def _estacion(self, m):
        for nom, meses in self.estaciones.items():
            if m in meses: return nom
        return 'otra'


    def _estad_ngramas(self, textos, n):
        lista = []
        for t in textos:
            w = t.split()
            for i in range(len(w)-n+1):
                lista.append(' '.join(w[i:i+n]))
        c = Counter(lista)
        frec = list(c.values())
        return c, {
            'total': len(lista),
            'unicos': len(c),
            'media': np.mean(frec) if frec else 0,
            'mediana': np.median(frec) if frec else 0
        }

    def _analiza(self, per_tipo, per_valor, tipo_ng, maxf=30, mindf=2):
        dfp = self.df[self.df[per_tipo]==per_valor]
        if dfp.empty: return pd.DataFrame()
        textos = dfp[self.col_texto].tolist()
        n = {'uni':1,'bi':2,'tri':3}[tipo_ng]
        cnt, stats = self._estad_ngramas(textos, n)
        tv = TfidfVectorizer(ngram_range=(n,n), max_features=maxf,
```

```
                            min_df=mindf,
token_pattern=r'\b[a-záéíóúñü]{3,}\b')
        try:
            X = tv.fit_transform(textos)
        except:
            return pd.DataFrame()
        fts = tv.get_feature_names_out()
        tf = np.mean(X.toarray(), axis=0)
        dfr = pd.DataFrame({tipo_ng:fts, 'tfidf':tf, 'freq':[cnt[g] for g
in fts]})
        if tipo_ng=='tri':
            dfr['score'] = dfr['tfidf']*0.7 +
(dfr['freq']/dfr['freq'].max())*0.3
            dfr.sort_values('score', ascending=False, inplace=True)
        else:
            dfr.sort_values('tfidf', ascending=False, inplace=True)
        dfr['periodo']=per_valor
        dfr['tipo']=per_tipo
        dfr['docs']=len(textos)
        return dfr

    def analisis_completo(self, maxf=30):
        for ng in ['uni','bi','tri']:
            self.res_temporales[ng] = {}
            for est in self.estaciones:
                self.res_temporales[ng][f'est_{est}'] =
self._analiza('estacion', est, ng, maxf)
        return self.res_temporales

    def guardar(self):
        base = self.archivo.replace('.xlsx','')
        salida = f"{base}_reporte_{self.col_texto}.xlsx"
        with pd.ExcelWriter(salida, engine='openpyxl') as w:
            for ng, datos in self.res_temporales.items():
                for per, df in datos.items():
                    if not df.empty:
                        df.to_excel(w, sheet_name=f"{ng}_{per}"[:31],
index=False)
        print("Guardado en", salida)
```

```
if __name__=='__main__':
    a1 =
AnalizadorNGramasTiempo(r"C:\Users\archivos_unidos_limpio_out.xlsx",
'Texto_Final')
    a1.cargar()
    a1.analisis_completo()
    a1.guardar()
```

Código gráficos:

```
import pandas as pd
import matplotlib.pyplot as plt
import os

def graficar_top_ngrams(excel_path, top_n=10,
salida_dir='graficos_ngrams'):
    os.makedirs(salida_dir, exist_ok=True)

    xls = pd.ExcelFile(excel_path, engine='openpyxl')
    hojas = xls.sheet_names

    for hoja in hojas:
        df = xls.parse(hoja)

        col_ngram = next((c for c in ['uni', 'bi', 'tri'] if c in
df.columns), None)
        if col_ngram is None or 'freq' not in df.columns:
            print(f"[!] Hoja '{hoja}' omitida: columnas necesarias no
encontradas.")
            continue

        df_top = df.nlargest(top_n, 'freq')

        plt.figure(figsize=(10, 6))
        plt.barh(df_top[col_ngram], df_top['freq'], color='teal')
        plt.xlabel('Frecuencia')
        plt.title(f'Top {top_n} {col_ngram}gramas - {hoja}')
        plt.gca().invert_yaxis()
        plt.tight_layout()
```

```
        nombre_archivo = f"{col_ngram}_{hoja[:20].replace('/', '_')}.png"
        ruta_salida = os.path.join(salida_dir, nombre_archivo)
        plt.savefig(ruta_salida)
        plt.close()
        print(f"Gráfico guardado: {ruta_salida}")

ruta_archivo =
r"C:\Users\archivos_unidos_limpio_out_reporte_FinalTxt.xlsx"
graficar_top_ngrams(ruta_archivo)
```

Código modelado de tópicos:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from openpyxl import Workbook
from openpyxl.utils.dataframe import dataframe_to_rows
import warnings
warnings.filterwarnings('ignore')
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import gensim
from gensim import corpora
from gensim.models import LdaModel, CoherenceModel

plt.style.use('seaborn-v0_8')

class LDAEstaciones:
    def __init__(self, ruta):
        self.ruta = ruta
        self.df = None
        self.xporEst = {}
        self.mod = {}
        self.coher = {}

    def cargar(self):
        self.df = pd.read_excel(self.ruta)
        self.df['Fecha'] = pd.to_datetime(self.df['Fecha'],
errors='coerce')
```

```
        self.df.dropna(subset=['Fecha'], inplace=True)
        self.df['mes'] = self.df['Fecha'].dt.month
        self.df['dia'] = self.df['Fecha'].dt.day
        self._separa()

    def _separa(self):
        estaciones = {
            'Primavera': [(3,1),(5,31)],
            'Verano':    [(6,1),(8,31)],
            'Otono':     [(9,1),(11,30)],
            'Invierno':  [(12,1),(2,28)]
        }
        for nom,((m0,d0),(m1,d1)) in estaciones.items():
            if nom!='Invierno':
                masq = (self.df['mes']>=m0)&(self.df['mes']<=m1)
            else:
                masq =
((self.df['mes']==12)&(self.df['dia']>=d0))|((self.df['mes']<=2)&(self.df[
'dia']<=d1))
            sub = self.df[masq].copy()
            sub['est'] = nom
            self.xporEst[nom] = sub

    def prep(self, datos, col='Texto_Final'):
        est = datos['est'].iloc[0]
        txts = datos[col].dropna().astype(str).tolist()
        docs = []
        for t in txts:
            if '[' in t:
                ws = t.strip('[]').replace("'","").split(',')
                docs.append([w.strip() for w in ws if w.strip()])
            else:
                docs.append(t.split())
        return [d for d in docs if len(d)>1]

    def busca_k(self, docs, est):
        dct = corpora.Dictionary(docs)
        dct.filter_extremes(no_below=2,no_above=0.8,keep_n=1000)
        c = [dct.doc2bow(d) for d in docs if dct.doc2bow(d)]
        ks, scrs, mods = [],[],[]
```

```
        for k in range(2, min(10,len(c)//2)+1):
            lda = LdaModel(corpus=c, id2word=dct, num_topics=k,
random_state=42,
                           passes=10, iterations=50, alpha='auto',
eta='auto')
            coh = CoherenceModel(model=lda, texts=docs, dictionary=dct,
coherence='c_v').get_coherence()
            ks.append(k); scrs.append(coh); mods.append((lda,dct,c))
        if not scrs: return
        bi = int(np.argmax(scrs))
        best_k = ks[bi]
        lda, dct, ccor = mods[bi]
        self.mod[est] =
{'m':lda,'d':dct,'c':ccor,'k':best_k,'coh':scrs[bi]}
        self.coher[est] = {'ks':ks,'scr':scrs}

    def info_top(self, est, n=10):
        m = self.mod.get(est)
        if not m: return []
        out=[]
        for i in range(m['k']):
            terms = m['m'].show_topic(i, topn=n)
            ws = [w for w,_ in terms]
            ps = [p for _,p in terms]
            out.append({
                'est':est,'top':i+1,
                'pal5':','.join(ws[:5]),'pal':','.join(ws),
                'prob':','.join(f"{p:.3f}" for p in ps),
                'coh':m['coh']
            })
        return out

    def guardar_excel(self, sal="res_lda_est.xlsx"):
        wb = Workbook()
        ws = wb.active; ws.title = "Resumen"
        # resumen
        for est,md in self.mod.items():
            ws.append([est, md['k'], md['coh'], len(md['d'])])
        # tópicos
        ws2 = wb.create_sheet("Topicos")
```

```
        for est in self.mod:
            for itm in self.info_top(est):
                ws2.append(list(itm.values()))
        # coherencias
        ws3 = wb.create_sheet("Coher")
        for est,co in self.coher.items():
            for k,s in zip(co['ks'],co['scr']):
                ws3.append([est,k,s])
        wb.save(sal)
        print("Guardado en", sal)

    def graficar(self):
        if not self.coher: return
        fig, axs = plt.subplots(2,2,figsize=(10,8))
        for ax,est in zip(axs.flatten(), self.coher):
            ax.plot(self.coher[est]['ks'], self.coher[est]['scr'],
marker='o')
            b = self.coher[est]['ks'][np.argmax(self.coher[est]['scr'])]
            ax.set_title(est); ax.axvline(b, ls='--'); ax.annotate(f"opt
{b}", (b, max(self.coher[est]['scr'])))
        plt.tight_layout()
        plt.show()

    def run(self):
        self.cargar()
        for est,df in self.xporEst.items():
            docs = self.prep(df)
            if docs:
                self.busca_k(docs, est)
        self.guardar_excel()
        self.graficar()

if __name__=='__main__':
    m = LDAEstaciones(r"C:\Users\archivos_unidos_limpio_out.xlsx")
    m.run()
```

Código análisis de sentimiento:

```
import pandas as pd
import numpy as np
```

```
from transformers import pipeline, AutoTokenizer,
AutoModelForSequenceClassification
from datetime import datetime
import calendar
import re
from collections import Counter
import warnings
warnings.filterwarnings('ignore')

ruta_entrada = r"C:\Users\archivos_unidos_limpio_out.xlsx"
df = pd.read_excel(ruta_entrada, sheet_name="Sheet1",
parse_dates=["Fecha"])

print(f"Datos cargados: {len(df)} registros")
print(f"Columnas disponibles: {df.columns.tolist()}")

columnas_necesarias = ['Fecha', 'Texto_Final']
faltantes = [col for col in columnas_necesarias if col not in df.columns]
if faltantes:
    print(f"Faltan columnas: {faltantes}")



periodos = {
    "Primavera":    {"inicio": {"mes": 3,  "dia": 1},  "fin": {"mes": 5,
"dia": 31}},
    "Verano":       {"inicio": {"mes": 6,  "dia": 1},  "fin": {"mes": 8,
"dia": 31}},
    "Otono":        {"inicio": {"mes": 9,  "dia": 1},  "fin": {"mes": 11,
"dia": 30}},
    "Invierno":     {"inicio": {"mes": 12, "dia": 1},  "fin": {"mes": 2,
"dia": 29}},
}

def asignar_estacion(fecha):
    if pd.isna(fecha):
        return "Sin_fecha"
    for clave, periodo in periodos.items():
        año_inicio = fecha.year
        if clave == "Invierno" and fecha.month in (1, 2):
            año_inicio -= 1
```

```
        mes_i = periodo["inicio"]["mes"]
        dia_i = periodo["inicio"]["dia"]
        dia_max_i = calendar.monthrange(año_inicio, mes_i)[1]
        inicio = fecha.replace(
            year=año_inicio,
            month=mes_i,
            day=min(dia_i, dia_max_i)
        )

        año_fin = año_inicio + 1 if periodo["inicio"]["mes"] >
periodo["fin"]["mes"] else año_inicio
        mes_f = periodo["fin"]["mes"]
        dia_f = periodo["fin"]["dia"]
        dia_max_f = calendar.monthrange(año_fin, mes_f)[1]
        fin = fecha.replace(
            year=año_fin,
            month=mes_f,
            day=min(dia_f, dia_max_f)
        )

        if inicio <= fecha <= fin:
            return clave
    return "Fuera_de_periodo"

df["estacion"] = df["Fecha"].apply(asignar_estacion)

negadores = {'no', 'nunca', 'jamás', 'tampoco', 'nada', 'nadie',
'ninguno', 'sin', 'ni'}
contrastivos = {'pero', 'sin embargo', 'aunque', 'no obstante', 'a pesar
de', 'por el contrario'}

def detectar_caracteristicas(texto):
    if pd.isna(texto) or texto == "":
        return {k: False for k in [
            'puntuacion','repeticion','negacion','contraste'
        ]}
    t = str(texto)
    minus = t.lower()
    palabras = minus.split()
```

```
    # Mayúsculas
    enf_may = any(c.isupper() for c in t)
    # Puntuación
    excl = t.count('!') + t.count('¡')
    inter = t.count('?') + t.count('¿')
    enf_pun = excl>=2 or inter>=2
    # Repetición letras
    rep = len(re.findall(r'(.)\1{2,}', t))>0
    # Negación
    neg = any(n in palabras for n in negadores)
    # Contraste
    con = any(c in minus for c in contrastivos)
    return {'mayusculas': enf_may, 'puntuacion': enf_pun,
            'repeticion': rep, 'negacion': neg, 'contraste': con}

print("Cargando modelos...")
try:
    modelo_beto = pipeline("sentiment-analysis",
                           model="finiteautomata/beto-sentiment-analysis",

tokenizer="finiteautomata/beto-sentiment-analysis",
                           device=-1, truncation=True, max_length=512)
    print("BETO cargado")
except Exception as e:
    print(f"Error BETO: {e}")
    exit(1)


    usar_ensemble = True


def calcular_score(res_beto, intensidad=1.0, neg=False, con=1.0):
    label_b = res_beto.get("label","").upper()
    sc_b = res_beto.get("score",0)
    base = sc_b if label_b.startswith("POS") else -sc_b if
label_b.startswith("NEG") else 0
    final = max(-1, min(1, base*intensidad*con * (-0.8 if neg else 1)))
    return base, final

def procesar(df, tam=16):
```

```
    resultados=[]
    for inicio in range(0,len(df),tam):
        fin = inicio+tam
        lote = df.iloc[inicio:fin]
        print(f"Lote {inicio//tam+1}")
        textos = lote['Texto_Final'].fillna("").tolist()
        validos = [i for i,t in enumerate(textos) if t.strip()]
        entr = [textos[i] for i in validos]
        res_b = modelo_beto(entr) if entr else []
        for i,t in enumerate(textos):
            if i in validos:
                idx = validos.index(i)
                rb = res_b[idx]
            else:
                rb = {'label':'NEUTRAL','score':0.5}
            carac = detectar_caracteristicas(t)
            t_excl = str(t).count('!') + str(t).count('¡')
            intensidad = 1 + t_excl*0.1 + (len(re.findall(r'(.)\1{2,}',
str(t)))*0.05)
            neg, con = carac['negacion'], 1.5 if carac['contraste'] else 1
            base, final = calcular_score(rb, intensidad, neg, con)

resultados.append({'indice':inicio+i,'base':base,'final':final,'intensidad
':intensidad,
                               'neg':neg,'con':con,**carac})
    return pd.DataFrame(resultados)

print("Iniciando análisis...")
res = procesar(df)
df_final = pd.concat([df.reset_index(drop=True),
res.reset_index(drop=True)], axis=1)

def clasificar(f):
    if f>=0.5: return 'MUY_POS'
    if f>=0.1: return 'POS'
    if f<=-0.5: return 'MUY_NEG'
    if f<=-0.1: return 'NEG'
    return 'NEU'

df_final['clasif'] = df_final['final'].apply(clasificar)
```

```
salida = "reddit_sentiment_analisis_profundo.xlsx"
with pd.ExcelWriter(salida) as writer:
    df_final.to_excel(writer, index=False, sheet_name="Sentimientos")
print(f"Guardado en {salida}")
```