Facultad de Ciencias Económicas y Empresariales

Aplicación de técnicas de *Machine Learning* en el análisis de sentencias sobre el delito de trata de menores

Autor: Lucía Agrasar González
Director: Lucía Barcos Redín

MADRID | Junio 2022

RESUMEN:

Los datos sobre el delito de trata de menores en España no son representativos con la realidad y se trata de un problema que ya ha planteado la necesidad de encontrar una solución. Desde ciertas instituciones, se ha remarcado la utilidad que tendrían las nuevas tecnologías para evaluar y entender cuáles son las razones que han llevado a esta situación. En esta línea, el número de sentencias que se dictan sobre el delito de tratas en los tribunales españoles es muy reducido. Por ello, en este trabajo, a través del uso de las técnicas de *Text Mining*, en general, y *Topic Modeling*, en concreto, se analizarán los patrones de este tipo de sentencias, comparándolas con la jurisprudencia existente sobre los delitos de prostitución y corrupción de menores e identificando así las posibles razones que pueden haber llevado a esta situación. Además, en el presente trabajo se realizará una revisión literaria de las herramientas de Machine Learning, analizando su utilidad en el mundo jurídico, los avances en este campo y el camino que falta por recorrer.

PALABRAS CLAVE: *Text Mining, Topic Modeling*, LDA, Trata de menores, *Machine Learning*, Derecho

ABSTRACT:

Data on the crime of child trafficking in Spain are not representative of reality and it is a problem that has already raised the need to find a solution. Some institutions have highlighted the usefulness of new technologies to evaluate and understand the reasons that have led to this situation. Additionally, the number of sentences handed down on the crime of trafficking in the Spanish courts is very low. Therefore, in this paper, using Text Mining techniques, in general, and Topic Modeling, in particular, the patterns of this type of sentences will be analyzed, comparing them with the existing jurisprudence on the crimes of prostitution and corruption of minors and thus identifying the possible reasons that may have led to this situation. In addition, a literature review of Machine Learning tools will be carried out, analyzing their usefulness in the legal world, the advances in this field and the road ahead.

KEY WORDS: *Text Mining, Topic Modeling, LDA, Child Trafficking, Machine Learning, Law*

Índice

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

UNODC: Oficina de las Naciones Unidas contra la Droga y el Delito

CITCO: Ministerio de Defensa de España

ICAT: Grupo Interinstitucional de coordinación contra la trata de personas

CP: Código Penal

ML: *Machine Learning*

IA: Inteligencia Artificial

TIC: Tecnología de la información y de la comunicación

NLP: Procesamiento del Lenguaje Natural

BOW: *Bag of Words*

LDA: *Latent Dirichlet Allocation*

PLSA: *Probabilistic Latent Semantic Analysis*

LSA: *Latent Semantic Analysis*

TF: *Term Frequency*

SVD: *Singular Value Descomposition*

IDF: *Inverse Document Frequency*

TDM: *Term-Document Matrix*

CAPITULO I: INTRODUCCIÓN

1. MOTIVACIÓN: PROBLEMÁTICA ACTUAL

En un reciente informe elaborado conjuntamente por el Instituto Universitario de Estudios sobre Migraciones de la Universidad Pontificia Comillas y el Comité de Unicef en España se constata que existe un problema de visibilidad del número de casos de menores que se encuentran en situación de trata, muy alejado de la realidad actual en España.

En el prólogo del informe, el director del Comité Español de Unicef, José María Vera, nos recuerda que *"*en el caso de la trata de seres humanos, ni los datos oficiales ni las palabras que los acompañan permiten conocer la historia completa*" (*Castaño et al., 2022, p.7). La dificultad en la identificación y detección es común para todo tipo de víctimas, explotadas de diversas formas: en el campo, a través de trabajos domésticos, la mendicidad o la explotación sexual. Y esa dificultad se hace más acusada, si cabe, en el caso de los niños, que generalmente son poco protagonistas en las estadísticas oficiales relativas a la transgresión de los derechos humanos.

En este sentido, el Comité de los Derechos del Niño había recomendado ya a España que pusiera el foco en mejorar la calidad y cantidad en la recogida de datos sobre infancia, especialmente de menores en situación de vulnerabilidad. Asimismo, indicó que se debía asegurar que esos datos se utilizarán para la elaboración y monitoreo de planes, programas y políticas contra la trata de estos menores.

El 18 de julio de 2018, UNICEF y el Grupo Interinstitucional de Coordinación contra la Trata (en inglés, ICAT) ya pusieron de manifiesto en un comunicado de prensa (Tidey, 2018) que aproximadamente un 28% de las víctimas identificadas de la trata en todo el mundo son niños.

Pero UNICEF y el ICAT -y esto es lo más preocupante- creían que el número de niños víctimas de la trata es incluso mayor de lo que sugieren esos datos:

> La realidad es que los niños no suelen identificarse como víctimas de la trata. Muy pocos lo reconocen por miedo a los traficantes, por el desconocimiento de otras

> alternativas posibles, por desconfianza de las autoridades, por temor al rechazo o la posibilidad de que los devuelvan sin ningún tipo de protección y con ayuda limitada (Tidey, 2018).

Por lo que se refiere a España, los datos y las estadísticas nos han ido narrando hasta ahora que la trata estaría protagonizada sobre todo por mujeres víctimas de explotación sexual, identificándolas con la prostitución; se ha ido sedimentando así una historia oficial que ignora a los menores, cuando estos son protagonistas de muchos de los titulares periodísticos. El Informe Global sobre Trata de Personas, elaborado por UNODC, ha estimado que el delito en menores de edad se ha triplicado en los últimos años mientras que los números en España no siguen la tendencia global.

Retomando el análisis del informe inicial, se expone que, para poder combatir esa realidad, resulta necesario conocer unas cifras fieles y un lenguaje común que no las enmascare, como sucede hasta ahora; cifras y lenguaje que manejen los mismos criterios para definir qué es lo que se considera o no como 'explotación', 'modos de captación' o 'víctimas'. Solo así, sin conceptos equívocos, podremos saber qué personas son verdaderas víctimas de este delito.

Por otro lado, además de la falta de consenso acerca de los conceptos que deben utilizarse, el informe plantea que no existe un organismo que aúne todos los datos de los actores implicados en el fenómeno, lo que dificulta el análisis global de esta lacra social.
Por último, otro de los grandes problemas se debe a que cada uno de los actores utiliza metodologías distintas para la recogida de datos. En este sentido, mientras que el CITCO centra su análisis en un estudio estadístico, el Ministerio Fiscal recoge aquellos datos necesarios para comenzar los procedimientos penales.

El presidente de UNICEF España plantea, por su parte, una nueva perspectiva a este problema al indicar que:

> Es necesario pensar nuevas formas de enfrentar viejos problemas y tenemos a la tecnología de nuestra parte. La tecnología, que tanto facilita la captación de las víctimas y que ha traído nuevas formas de explotarlas, puede ser también utilizada para un mejor registro y seguimiento de los casos de explotación, y un mayor acceso a sus derechos por parte de las víctimas (Castaño et al., 2022, p.7).

De esta forma se plantea si, a través del uso de nuevas tecnologías, se podrían encontrar datos más fidedignos sobre el problema planteado. Un buen tratamiento de los datos existentes, a través de la implementación de las herramientas de *Machine Learning* -de las que ya se está haciendo uso en la mayor parte de las áreas- puede ayudar a obtener una mayor visibilidad acerca de las cuestiones detrás de la ausencia de datos.

Ahondando aún más en el problema, encontramos también un número muy reducido de casos de trata en el sistema de justicia penal. Los datos de los que disponemos, no solo a nivel nacional, sino también a nivel internacional, son escasos e incompletos por lo que es necesario analizar cuál es el origen de esta situación (Bjelland 2017; Rusell, 2018).

Como ya hemos expuesto, se descarta la opción de que se trate de un delito que ha disminuido con el paso del tiempo, ya que las tendencias globales indican una situación muy distinta. Ahora bien, casos con titulares como "Tres detenidos en Málaga acusados de abuso sexual a tres chicas fugadas de centros de menores" (Europa Press, 2020) o "Redada policial contra la prostitución de menores en Mallorca con 17 detenidos" (El Mundo, 2021) no fueron conductas calificadas como trata de seres humanos, sino como delitos de corrupción y prostitución de menores, aunque por los hechos narrados pudieran indicar lo contrario.

Es así como esta situación nos hace pensar que uno de los problemas podría radicar en que nuestros tribunales están calificando conductas típicas de trata de menores (art. 177 *bis* CP) como otros tipos penales, como son los delitos de prostitución y corrupción de menores (artículos 183,187 y 188 CP).

2. OBJETIVOS

A través del uso de las herramientas de *Machine Learning*, en este trabajo se intentará analizar la problemática sobre la ausencia de datos del delito de trata de menores desde un punto de vista jurisprudencial. De esta forma, el objetivo consistirá en analizar y comparar las sentencias del delito de trata de menores con aquellos pronunciamientos sobre los delitos de prostitución y corrupción, identificando los patrones clave de cada uno y exponiendo en qué tipo de supuestos los jueces están optando por calificarlo bajo cada tipo de delito. Asimismo, se busca demostrar que el uso de los métodos de *Text*

Mining, y en concreto de *Topic Modeling*, es eficaz para este tipo de análisis y que su implementación en el mundo del Derecho ya es una necesidad ante la creciente tendencia de este tipo de problemas. Para ello, se realizará una revisión de las principales investigaciones y aplicaciones de estas técnicas en las distintas áreas jurídicas, con el fin de contextualizar y exponer los avances realizados.

3. METODOLOGÍA

Para poder alcanzar las conclusiones acerca del tema expuesto, es necesario seguir una metodología que nos proporcione las herramientas teóricas y prácticas para ello. Es así como la metodología utilizada para el desarrollo del presente trabajo comienza con una revisión de términos jurídicos que permitirán entender el análisis de los textos jurídicos. Posteriormente, se realiza una revisión de literatura, en concreto sobre las herramientas de *Machine Learning* y su aplicación en el mundo del Derecho, incidiendo en las técnicas de *Text Mining* y *Topic Modeling* que se implementan en la parte práctica. En esta última parte se ha procedido a una compilación de sentencias seleccionadas a través de motores de búsqueda del CENDOJ, indicando concretamente los delitos sobre los que debían tratar. Las fuentes que han servido para la elaboración del trabajo se han obtenido de *Google Scholar* y de *Scorpus* en su mayoría, así como de publicaciones en revistas y libros científicos. Por último, para la aplicación concreta de paquetes novedosos del entorno R se han utilizado los recursos propios de ayuda con los que cuenta la herramienta, así como publicaciones prácticas que han servido a modo ejemplificativo.

4. ESTRUCTURA

Finalmente, se ha seguido una estructura inductiva en la elaboración de este trabajo, comenzando con un estudio general de los métodos de *Machine Learning*, para finalizar exponiendo las técnicas propias de *Text Mining* y su aplicación directa a las sentencias sobre trata. Por ello, en el capítulo 2 y 3 se realiza un estudio general sobre la estructura de textos jurídicos y sobre las técnicas de *Machine Learning* empleadas en el mundo legal, respectivamente. Posteriormente, en el capítulo IV se expone de una forma teórica cuáles son las técnicas y algoritmos concretos que permitirán realizar el análisis posterior, explicando de forma detallada los conceptos relativos al *Text Mining* y al *Topic Modeling*. Finalmente, en el capítulo V se implementan dichas técnicas sobre la base de datos creada

con las sentencias seleccionadas y se exponen los resultados analizados, resumiendo en último lugar las conclusiones obtenidas a través de gráficos y tablas resumen.

CAPÍTULO II: CUESTIONES LEGALES PREVIAS

Si bien el trabajo se enfocará inicialmente en una revisión literaria de las técnicas de *Machine Learning* y posteriormente en la aplicación práctica en las sentencias de trata de menores, es conveniente entender algunos conceptos jurídicos previos que nos servirán para entender los resultados que obtengamos.

Por un lado, al analizar sentencias de tres tipos penales, es conveniente diferenciar cuáles son las conductas típicas que se penan. Seguidamente, se analizará cuál es la estructura propia de una sentencia con el fin de comprender qué partes aportan información relevante para el análisis concreto de este trabajo.

1. TIPIFICACIÓN DE LOS DELITOS PENALES

En primer lugar, el delito de trata es un delito que se tipifica en nuestro ordenamiento jurídico en el artículo 177 *bis* del Código Penal. Consiste en el proceso conducente a una situación de explotación ya sea con finalidades de explotación sexual (incluyendo la pornografía), laboral, para la realización de actividades delictivas, para la extracción de órganos o para la celebración de matrimonios forzados. En este delito, el consentimiento dado por la víctima es irrelevante cuando se demuestra que se ha realizado mediando medios ilícitos. Hasta la aprobación de la Ley Orgánica 2/2010, este delito estaba castigado por la vía del delito de tráfico de personas migradas, o bien, cuando se trataba de menores, se solía calificar como prostitución o corrupción de menores (Del Moral, 2020).

Si analizamos más en detalle el artículo y ponemos el foco en la trata de menores podemos empezar a extraer conclusiones. En primer lugar, este tipo penal está castigado con entre cinco y ocho años de pena de prisión que, como ya veremos, es un periodo superior al del resto. Por otro lado, el legislador ha querido penar la conducta típica del proceso completo de traslado de la víctima para su explotación (captura, transporte, traslado, acogimiento…). Por su parte, enumera en su primer apartado una serie de medios de

explotación, entre los que incluye: la violencia, intimidación o engaño y el abuso de situación de superioridad, de necesidad o de vulneración.

Ahora bien, el apartado 2 del artículo 177 *bis* analizado se centra en la trata de menores y especifica que tendrá la consideración de trata de seres humanos cualesquiera de las acciones indicadas anteriormente, sin exigirse que concurra ninguno de los medios enunciados, cuando se realice respecto de menores de edad con fines de explotación. Es decir, no es necesario que exista ni violencia, ni intimidación, etc., para que se califique como delito de trata de menores.

Por su parte, los delitos de prostitución y corrupción de menores vienen recogidos bajo el mismo título en el Capítulo V (artículos 188 a 190) del CP. Concretamente, el delito de prostitución de menores, consagrado en su literalidad en el artículo 188 CP, se castiga con pena de prisión de dos a cinco años. Este artículo indica que será penado "el que induzca, promueva, favorezca o facilite la prostitución de un menor de edad (…) o se lucre con ello, o explote de algún otro modo a un menor o a una persona con discapacidad para estos fines". De esta forma, se castiga al que ayude de cualquier forma y haga posible la práctica de la prostitución. En este caso, si el menor fuera menor de 16 la pena sería de cuatro a 8 años.

Seguidamente, el artículo 189 CP castiga con pena de uno a cinco años al que capturare o utilizase a los menores para espectáculos exhibicionistas o pornográficos, así como el que produjera, vendiera, distribuyera pornografía infantil. Ahora bien, si el menor fuera menor de 16 años, la pena impuesta aumentaría hasta el rango de cinco a nueve años. Cabe mencionar que en el artículo 189 bis se hace mención concreta al delito relacionado con difusión de este contenido por medios como Internet, dispositivos móviles u otros aparatos tecnológicos.

Finalmente, bajo el concepto de "corrupción de menores", los jueces también han venido calificando el artículo 183 del CP que es un delito que trató de dar respuesta a un vacío legal en el ámbito penal sobre los delitos contra la libertad sexual, ya que no existía un artículo concreto para los menores y se recurría a los tipos agravados. De esta forma, cuando se trata de delitos sexuales contra menores, el bien jurídico protegido no es la

libertad sexual, sino la indemnidad sexual y al encontrarse en una mayor situación de vulnerabilidad, el legislador decidió darle su protección particular. De esta forma, se condena con pena de prisión, de dos a 6 años, al que realizara actos de carácter sexual (abusos sexuales) a menores de 16.

2. ESTRUCTURA DE LAS SENTENCIAS

Al utilizar las sentencias como bases de datos de nuestro análisis, debemos entender la estructura propia de este tipo de documentos.

En primer lugar, todas las sentencias cuentan con un encabezamiento, donde se indica entre otros aspectos el lugar donde se conoce el asunto, el juez (o ponente en caso de ser un órgano colegiado), el número de la sentencia que permite identificarla, la fecha en la que se dicta, el nombre de las partes intervinientes o el tipo de delitos que se imputan. Esta información no nos será de relevancia para este análisis, aunque sí que sería relevante desde otros enfoques: *sentiment analysis* (relacionándolo con los jueces) y *topic modeling* (categorizar las sentencias según fecha, tipos de delitos, etc.). Es por ello por lo que muchos de los términos que aparecen en el encabezamiento los trataremos como si fueran palabras vacías o carentes de contenido (*stopwords* en adelante) y las eliminaremos de nuestros textos.

Posteriormente, en las sentencias nos encontramos con los antecedentes de hecho, donde se exponen las pretensiones de las partes, los hechos acaecidos, las pruebas propuestas y los hechos probados. Los hechos probados podrían considerarse la parte más importante de la resolución, de forma que debe constar con gran precisión y claridad la valoración de las pruebas. De esta forma, servirán como base de nuestro análisis y nos permitirán encontrar patrones acerca de las circunstancias vinculadas a la comisión de los diferentes delitos.

En tercer lugar, la sentencia recoge los fundamentos de Derecho. En este apartado el juez o tribunal aplica la ley vigente de tal forma que se motiva la decisión final sobre los hechos anteriormente relatados. Esta información también nos será de gran utilidad para

poder entender qué razonamiento legal han seguido los jueces para calificar la conducta por un tipo diferente al delito de trata de menores.

Finalmente, el documento concluye con el fallo, que es la decisión final que resuelve las peticiones de las partes y que determina si el acusado queda absuelto o condenado, indicando el tipo de delito por el que se le condena. En este sentido, no quedan dudas de que el fallo será vital para delimitar y agrupar las distintas sentencias en base al tipo penal: trata de menores, corrupción de menores o prostitución de menores.

CAPITULO III: TÉCNICAS DE *MACHINE LEARNING* EN EL MUNDO DEL DERECHO

1. INTELIGENCIA ARTIFICIAL, *BIG DATA* Y *MACHINE LEARNING*

El uso de las nuevas Tecnologías de la Información y la Comunicación (TIC), y en concreto el de las técnicas de Inteligencia Artificial, se viene aplicando durante este último siglo en un gran abanico de áreas: gestión financiera, estudios médicos, ámbito industrial, etc… y el Derecho no se ha quedado al margen. Aunque pudiera parecer que su aplicación en el mundo jurídico es menos palpable, debido a las reticencias ligadas a la tradicional forma de ejercer el derecho y a la desestructuración de los datos jurídicos, el impacto que está teniendo es realmente cuantificable.

En primer lugar, y para comenzar con el contexto, podemos definir la Inteligencia Artificial como aquellos sistemas que manifiestan un comportamiento inteligente y que tienen la capacidad de tomar decisiones autónomamente con el fin de lograr ciertos objetivos específicos (Borges, 2020). En cuanto a su funcionamiento, la Inteligencia Artificial tiene que ver con "el procesamiento de la información para resolver problemas y tomar decisiones a partir de máquinas que operan a través de los llamados algoritmos inteligentes" (Corvalán, 2018, p.299). De esta forma, la IA utiliza información procedente de otras máquinas o generada por las personas, con el objetivo de crear modelos que produzcan resultados en forma de recomendaciones, predicciones o decisiones.

Pues bien, para poder llevar a cabo este proceso, es necesario que exista una secuencia de instrucciones, conocida como la estructura algorítmica (Nava, 2017, p.24). Aunque han

sido diversas las definiciones que se han otorgado al concepto de algoritmo, la Agencia de los Derechos Fundamentales de la Unión Europea, ha señalado que "es aquella secuencia de comandos que permite a una computadora tomar entradas y producir salidas y usarlos puede acelerar los procesos y producir resultados más consistentes" (Eguíluz, 2020, p. 329). Es así como los algoritmos tienen como objetivo encontrar correlaciones entre los datos y para lograrlo operan sobre bases de datos que estén ordenadas de forma comprensible para su procesamiento.

Un concepto distinto, pero también a la orden del día es el *Big Data*. Este nuevo fenómeno, cuya definición se da a conocer en el trabajo de Mayer-Schönberger y Cukier (2013), es una ciencia que se basa en aplicar herramientas matemáticas a grandes bancos de datos para inferir probabilidades. De esta forma, al trabajar con métodos estadísticos, cuanto mayor es la base de datos con la que se trabaja, mejor es el resultado o predicción que se obtiene. Por ello, aunque se trata de conceptos distintos, están íntimamente relacionados. De esta forma, la Inteligencia Artificial necesita datos para construir sus procesos y el hecho de que la base del análisis esté configurada por grandes cantidades de datos permite que los resultados que se extraigan sean más precisos.

Pues bien, la IA se apoya en algoritmos avanzados o de aprendizaje automatizado, que quedan recogidos bajo el concepto de técnicas de *Machine Learning*. Esta disciplina, puede definirse como el proceso de entrenamiento de un modelo computacional para cumplir una tarea con datos (Nay, 2018) y está enfocado en el autoaprendizaje a través del estudio del ambiente que los rodea. Este aprendizaje puede ser supervisado o no supervisado. En el primero se combinan variables predictivas y de resultado, que permiten entrenar un modelo y este a su vez predecir nuevos resultados con nuevas observaciones. En función del resultado que se obtenga podemos encontrar modelos de regresión (el resultado es numérico) o modelos de clasificación (el resultado es categórico). Por su parte, en el aprendizaje no supervisado no encontramos un valor correcto que se deba replicar durante el entrenamiento de los modelos y lo que se busca es encontrar patrones existentes en los datos.

Por otro lado, los datos con los que se trabajan pueden provenir de diferentes fuentes y dependiendo de su origen habrá que darles un tratamiento diferenciado. Así, atendiendo a la clasificación de Joyanes (2013), nos encontramos con tres categorías:

- Datos estructurados: los datos se muestran en un formato definido con campos fijos. Ejemplos: hojas de cálculo, archivos, bases de datos relacionales o SQL…
- Datos semiestructurados: no existe un formato específico, pero se pueden encontrar marcadores que intentan clasificar ciertos elementos. Por ejemplo: textos con etiquetas XML y HTML.
- Datos no estructurados: encontramos la mayoría de los datos dentro de esta categoría. Se trata de datos de carácter indefinido, que están almacenados en documentos sin estructura fija y que, si bien pueden ser entendibles para los humanos, no lo son para los ordenadores. Por ejemplo: correos electrónicos, archivos PDF, vídeos, audios…

2. HERRAMIENTAS DE *MACHINE LEARNING* EN EL DERECHO

Retomando el foco sobre el análisis de datos de base jurídica, hay que entender que los procesos de razonamiento legal y de toma de decisiones dependen de la información que se recoge en documentos no estructurados (sentencias, contratos, normativa, etc.). Estos textos legales son redactados en su mayoría por personas, que -aunque son expertas en la materia- tienen una capacidad de procesamiento limitada. La producción de esta información se está incrementando a un ritmo mayor del que se puede procesar por lo que las herramientas de *Machine Learning* están dando solución a este problema.

Parte de la doctrina se ha cuestionado en los últimos años: "¿en qué medida las técnicas de *Machine Learning* podrían ser aplicadas a la práctica de la ley?" (Harry, 2014). Este sector argumenta que, si bien dichas herramientas pueden ayudar a reducir tiempos, las decisiones judiciales no pueden ser reemplazados por algoritmos por completo. Ahora bien, otra parte de la literatura jurídica, como Kleinberg et al., (2017), grandes defensores del uso de estas técnicas en el mundo jurídico ya han utilizan algoritmos de *Machine Learning* en la toma de decisiones penales en Estados Unidos. De hecho, en este caso, gracias al uso de estas herramientas se pudo concluir que ciertas decisiones judiciales, como es la puesta en libertad preventiva del acusado hasta el juicio, dependían altamente de factores subjetivos y no tanto de parámetros objetivos como el score de riesgo criminal.

Por otro lado, los datos legales, que han crecido a una rapidez desmesurada con la incorporación de las bases de datos *online*, presentan dificultades de procesamiento para

las herramientas de *Machine Learning*. Por un lado, como ya se ha comentado, los textos presentan un formato desestructurado ilegible para los sistemas computacionales. Por otro lado, al ser textos escritos por personas y no por máquinas, se añaden las ambigüedades propias del lenguaje natural humano.

En esta línea, y de forma genérica, surge el *Data Mining* por la necesidad de explotar y dar valor a la gran cantidad de datos que hoy en día están almacenados en las empresas, instituciones públicas, en bases de datos personales, etc. De esta forma, aunque tradicionalmente se utilizó la estadística para el estudio de estos datos, con el gran volumen y variedad existentes se hace necesario desarrollar nuevas disciplinas.

En este sentido, la 'minería de datos' o *Data Mining*, que según Molina (2000) se refiere a aquel proceso de extracción de conocimiento de las bases de datos, tiene como principal objetivo encontrar patrones o modelos a partir de la información obtenida y decidir si son útiles o aportan valor.

Concretamente, como disciplina propia, nos encontramos con la minería de datos o *Text mining*. Esta modalidad busca extraer conocimientos útiles como relaciones, patrones y tendencias de datos no estructurados o semiestructurados como, por ejemplo, documentos de texto (Feldman y Sanger, 2006). En este sentido, el proceso consiste en la transformación de textos a través de la utilización de métodos estadísticos para recoger la información en una matriz documento-término organizada que incluye dos dimensiones: los términos y los documentos (Moro et al., 2019).

Antes de continuar, debe quedar claro que, aunque el *Data Mining* y el *Text Mining* son procesos analíticamente complementarios, difieren en el tipo de datos que manejan. De esta forma, como señalan Feldam y Sanger (2006):

> Las diferencias entre *Data Mining* y *Text Mining* se pueden evidenciar porque, en la primera, los datos se guardan en formatos estructurados, se centran en depurar estos datos y de esta manera en crear un gran número de uniones de tablas. En contraste, en el *Text Mining* el procesamiento se enfoca en reconocer y extraer características representativas para documentos en lenguaje natural. (p.6)

Pues bien, el *Text Mining* comprende un abanico de técnicas, entre las que encontramos el Procesamiento de Lenguaje Natural (NLP). Se trata de un "conjunto de técnicas computacionales que analizan y representan de forma natural lo que ocurre en los textos en distintos niveles del lenguaje" (Liddy, 2001, p.1). De esta forma, el NLP tiene como objetivo transformar el lenguaje natural en un lenguaje formal que permita a los ordenadores procesar la información recogida en estos textos.

Dentro de la disciplina del *Text Mining*, ya en 2018 se comenzó a hablar del concepto de *Legal Data Mining*. Durante el "*International Workshop on Legal Data Analytics – LeDAM*", se delimitó como una nueva subtarea dentro del ámbito del *Text Mining*, aplicado al análisis de textos legales, como jurisprudencia, contratos, patentes, recursos, doctrina académica, etc. En este contexto se comienza a conocer el concepto de Jurimetría (o *Jurimetrics*) que, a través de métodos cuantitativos, permite predecir decisiones de los tribunales, identificar patrones en los textos y, en general, convertir las decisiones judiciales en procesos más predecibles (Armonas et al., 2017). La Jurimetría, que ya llevaba existiendo desde hacía varios siglos y se definía como la investigación científica de los problemas legales (Loevinger, 1971), ha comenzado a utilizarse hace poco en España. En este sentido, la editorial jurídica Wolters Kluwer ha desarrollado una herramienta bajo ese nombre que asesora sobre cuál es la mejor estrategia para el caso que se analiza, aportando información intuitiva al profesional a través del análisis de más de 10 millones de resoluciones jurídicas.

Pues bien, del concepto de Jurimetría ha surgido la expresión *"Legal informatics"* que según Michael Genesereth (2020), profesor de la Universidad de Stanford, se define como aquella rama que se ocupa de la representación de las normas de forma computable y que estudia los sistemas que son capaces de realizar cálculos jurídicos de utilidad. Concretamente, dentro de esta rama encontramos tres subcategorías que han sido aprobadas por la mayor parte de expertos en el campo:

1. Informática Jurídica de Gestión (IJG): se encarga de utilizar herramientas computacionales para mejorar los procesos judiciales, ayudando en las tareas cotidianas de peritos, jueces, abogados, etc. Se basa en prestar asistencia en ciertas tareas como son el almacenamiento de datos, el procesamiento de textos jurídicos, el envío de comunicaciones, la elaboración de agendas de trabajo, etc.

2. Informática Jurídica Documental (IJD): se encarga de analizar, resumir y clasificar documentos legales para la posterior toma de decisiones, así como para la recuperación de documentos en amplios repositorios jurídicos. Por ello, entre las tareas incluidas bajo este concepto, encontramos el resumen y clasificación de sentencias, la determinación de líneas jurisprudenciales, la búsqueda selectiva en bases jurídicas, etc. En esta rama se utilizan herramientas de minería de datos, el procesamiento del lenguaje natural, análisis de sentimientos, el *Topic Modeling* o métodos de clasificación supervisada y no supervisada.

3. Informática para la Ayuda de Toma de Decisiones Jurídicas (IATDJ): se centra en ayudar a predecir decisiones utilizando modelos estadísticos de *Machine Learning* que simulan el razonamiento jurídico. Algunas de las técnicas más utilizadas se han centrado en modelos basados en redes neuronales.

En cuanto a los usos concretos de las herramientas de *Machine Learning* en las tres áreas comentadas anteriormente, observamos que muchas de las técnicas pueden utilizarse con distintos fines. A continuación, expondremos algunos de los trabajos e investigaciones más relevantes llevados a cabo en el área del Derecho.

En primer lugar, dentro de las tareas propias que se engloban bajo el concepto de Informática Jurídica de Gestión, el *Topic Modeling* -técnica propia del *Text Mining* y que desarrollaremos más a fondo en el siguiente capítulo- es una herramienta que se utiliza para encontrar conceptos semánticos o temas presentes en documentos. En concreto, el Latent Dirichlet Allocation es el algoritmo más utilizado dentro del *Topic Modeling*. Pues bien, en el ámbito jurídico, el algoritmo LDA se ha implementado para estudiar la agenda de Tribunales como el Tribunal Supremo de Estados Unidos (Livermore et al., 2016), el Tribunal de Justicia de la Unión Europea (Lampach y Dyevre, 2018) o el Tribunal Superior de Australia (Carter et al., 2016).

Por otro lado, en estos últimos años se ha incrementado el uso del *Text Mining* para el resumen de textos legales. Este proceso, recogido bajo el concepto de "*Document Summarization*" se basa en dos enfoques: uno abstractivo y otro extractivo. En las técnicas de resumen abstractivo se tiene en cuenta todo el texto y se reformula de una manera más simple, en muchas ocasiones utilizando palabras y frases distintas a la

original. Por su parte, las técnicas de resumen extractivo se encargan de elegir aquellas partes del texto que se incorporarán al resumen. Las primeras son más difíciles que las segundas, y por ello requieren de algoritmos más complejos (Anand y Wagh, 2022).

La investigación sobre las técnicas de resumen se ha incrementado recientemente con las nuevas técnicas de *Machine Learning*. En concreto, se ha venido utilizando un enfoque para agrupar sentencias basadas en temas obtenidos a través del algoritmo LDA y, de esta forma, encontrar el resumen de cada documento utilizando los mismos temas. Un enfoque más simple consiste en limitarse a utilizar las técnicas más básicas del *Text Mining* en el que las puntuaciones TF-IDF de las palabras de una frase se vayan sumando y se normalizan por la longitud de la frase con el fin de averiguar la puntuación de importancia de cada término.

En este sentido, los buscadores de jurisprudencia (Vlex, Aranzadi, CENDOJ), utilizan tecnologías de resumen de sentencias, de forma que extraen y muestran los conceptos clave de una sentencia y a través de la aplicación del *Machine Learning* sugieren casos similares. De esta forma, el usuario puede determinar si tiene interés por una sentencia en concreto y facilita la búsqueda posterior de sentencias parecidas. Como se comentará posteriormente, se ha hecho uso de estos motores de búsqueda para la configuración de nuestra base de datos.

Por su parte, otra de las técnicas utilizadas es el *Sentiment Analysis* para la identificación de líneas jurisprudenciales. Cuando un Tribunal toma una decisión existen factores psicológicos que pueden condicionar el fallo final. Aunque la ley debe prevalecer y los jueces deben apoyarse en ella, es inevitable que no se vean influenciados por sus valores personales o emocionales de forma que pueden derivar en opiniones contrapuestas dentro de los órganos colegiados, como se refleja en los votos particulares. De esta forma, existen algoritmos propios del *Text Mining* que permiten identificar los sentimientos ocultos en los textos. Muchos de los análisis que han utilizado estas herramientas de *Sentiment Analysis* se han centrado en determinar los sentimientos (positivos o negativos) de los jueces hacia una serie de grupos sociales objetivo (colectivos según su color de piel, ideas políticas, origen, etc.). Este es el caso de estudios como el de Elliott Ash, Daniel L- Chen y Sergio Galletta (2022), que analiza las opiniones de los Tribunales de Apelación de

Estados Unidos y otros que aplican estas herramientas para analizar las decisiones del Tribunal Supremo de Brazil (Sila et al., 2018).

Finalmente, uno de los objetivos que ha tenido la implementación del *Machine Learning* en el mundo del Derecho es la predicción de sentencias jurídicas. En este sentido, y en el contexto de la aparición de la Jurimetría, surge el concepto de "Justicia predictiva", entendida como *"*la posibilidad de prever el resultado de un juicio a través de algunos cálculos, en particular predecir la probable sentencia relativa a un caso específico, con el auxilio de algoritmos" (Viola, 2018).

La primera vez que se hizo uso del término fue en 2013 en Estados Unidos por la Corte Suprema de Winsconsin, en donde se utilizó el algoritmo COMPAS para calcular la probabilidad de que una persona reincidiese en la comisión de un delito y asesoró sobre el tipo de supervisión que debería recibir el condenado a prisión.

En casos más recientes (Katz et al., 2017), se ha utilizado como información para el análisis la base de datos de la Corte Suprema de Estados Unidos y se han aplicado algoritmos de aprendizaje supervisado como el *random forest*, máquinas de soporte vectorial y redes neuronales en la predicción de fallos de sentencias.

También cabe mencionar el ensayo que realizaron conjuntamente la Universidad de Pennsylvania, la Universidad College London y la Universidad de Sheffield (Aletras et al., 2016) en donde predijeron el resultado del fallo de la Corte Europea de los Derechos Humano un 79% de las veces.

Finalmente, en estos últimos años ha surgido el debate sobre la posibilidad de introducir "jueces robots" en los procesos judiciales. En concreto, países como Estonia y Estados Unidos ya son pioneros en esta práctica, habiendo logrado predicciones con una eficacia en un 85% de los casos. Ahora bien, una de las grandes amenazas radica en los sesgos algorítmicos, ya que la IA no interpreta los resultados, sino los patrones que se repiten. Esto podría llegar a perjudicar a ciertos grupos minoritarios o a comunidades más pobres, aumentando aún más las desigualdades en los procesos judiciales.

CAPITULO IV: DETALLE DE LAS TÉCNICAS EMPLEADAS

1. TEXT MINING

1.1 Definición y conceptos generales

Como ya hemos comentado, los textos son datos no estructurados que presentan un contenido solo legible por humanos y que además pueden incluir errores gramaticales, abreviaturas o una incorrecta utilización de las reglas de puntuación. Por ello es necesario seguir un proceso que nos permita transformar los cuerpos de los textos en datos que puedan ser procesados por los algoritmos de *Machine Learning*.

En primer lugar, cabe delimitar los conceptos básicos inherentes al proceso de *Text Mining*. Por un lado, un documento es una forma de dato no estructurado que hace referencia a la agrupación de un conjunto de términos, sin tener en consideración su longitud (puede ser una oración, un *report*, un comentario en Twitter o una sentencia). A la colección de documentos se le da el nombre de corpus. Por su parte, estos documentos están formados por *tokens* o términos individuales, que en la mayor parte de las ocasiones serán las propias palabras del texto.

En este sentido, se conoce como *tokenizacion* al proceso de descomposición de un texto en palabras, símbolos, frases, etc., es decir, en un conjunto de términos o *tokens*. De esta forma, la lista de *tokens* servirá como base de análisis para el procesamiento futuro y para la implementación de técnicas concretas de *Text Mining* futuras. No obstante, uno de los grandes problemas es que es muy común tener documentos con muchos *tokens*, lo que dificulta el análisis posterior. Es por ello por lo que se implementan técnicas de preprocesamiento de los documentos con el objetivo de reducir el número de términos, eliminando aquellas palabras vacías de contenido para el análisis.

Para entender mejor este proceso, se expone un ejemplo a lo largo de este apartado. De esta forma, partimos de la creación de un corpus que está formado por dos documentos, que en este caso son frases:

- Documento 1: Las menores ejercían la prostitución
- Documento 2: Cometió el delito de prostitución y el delito de corrupción de menores

Para comenzar nuestro análisis, debemos llevar a cabo el proceso de *tokenizacion* ya comentado. Una vez delimitados los *tokens*, se procede a la fase de preprocesamiento.

- Documento 1 (5 tokens): “las”, “menores”, “ejercían”, “la”, “prostitución”
- Documento 2 (12 tokens): “cometió”, “el”, “delito”, “de”, “prostitución”, “y”, “el”, “delito”, “de”, “corrupción”,” de”, “menores”

1.2 Técnicas de pre-procesamiento de datos

En cuanto a las técnicas de limpieza de datos no siempre se aplican inmediatamente tras el proceso de *tokenizacion* de los documentos, ya que pueden realizarse sobre el propio corpus o en pasos posteriores, una vez creada la matriz de frecuencias expuesta posteriormente.

1.2.1 Eliminación de stopwords

En este caso, una vez que hemos dividido el texto en tokens, a priori puede observarse que no todas las palabras aportan la misma información. De esta forma, aquellos términos que aparecen con gran frecuencia, pero que no aportan mucho valor ya que suelen utilizarse para dar coherencia al texto reciben el nombre de *stopwords.* En este sentido, las *stopwords* suelen ser las palabras más comunes en cualquier idioma incluyendo artículos, preposiciones, pronombres, conjunciones, etc. Cuando eliminamos estos términos, ponemos el foco en la información realmente importante del texto, reduciendo el tamaño del *dataset* y el tiempo de procesamiento posterior de los datos. Los programas (como R studio o Python) suelen incluir su propio vocabulario de *stopwords* para cada idioma, pero el usuario también puede crear una lista propia, como veremos en la parte práctica.

De esta forma, aplicando las técnicas de *stopwords* a nuestro ejemplo podríamos reducir la longitud del vocabulario de cada documento:

- Documento 1 (~~5~~ **3 tokens**): ~~“las”,~~ “menores”, “ejercían”, ~~“la”,~~ “prostitución”
- Documento 2 (~~10~~ **6 tokens**): “cometió”, ~~“el”,~~ “delito”, ~~“de”,~~ “prostitución”, ~~“y”, “el”,~~ “delito”, ~~“de”,~~ “corrupción”, “menores”.

Dentro de esta técnica, encontramos 4 tipos de métodos distintos (Vijayarani y Ilamathi, 2015):

- El método clásico: se basa en la eliminación de palabras vacías obtenidas de listas precompiladas.

- Métodos basados en la ley de Zipf (métodos Z): estos métodos incluyen la eliminación de las palabras más frecuentes y eliminan las palabras que solo aparecen una vez. También consideran eliminar las palabras con baja frecuencia inversa en el documento. Todos estos conceptos se explicarán en el siguiente punto.

- Método de la información mutua (MI): es un método supervisado que funciona computando la información común entre un término dado y una clase de documento (positivo, negativo, etc.), proporcionando una sugerencia sobre cuanta información puede aportar el término sobre una clase determinada. De esta forma, una información mutua baja sugiere que el término tiene un alto poder de discriminación y que por lo tanto debe eliminarse.

- Muestreo aleatorio basado en términos (TBRS): este método fue propuesto por primera vez por Lo et al., (2005) con el fin de detectar de forma manual las *stopwords* en los documentos web. Este método funciona iterando sobre cadenas de datos que han sido seleccionados de forma aleatoria. A continuación, clasifica los términos en cada cadena en base a sus valores de formato utilizando la medida de divergencia de Kullback-Leibler, donde Px(t) es la frecuencia de términos normalizada de un término 't' en un subconjunto y P(t) es la frecuencia, también normalizada, de 't' en toda la colección. De esta forma, la lista definitiva de *stopwords* se configura con las palabras menos informativas de todas las cadenas.

1.2.2 Steeming

Se trata del proceso que busca identificar la raíz/stem de varias palabras. En este sentido, se trata de eliminar los prefijos y sufijos de los tokens, de forma que se logra también

reducir el volumen de términos. No obstante, se suelen cometer además dos errores al utilizar este tipo de algoritmos:

- *Overstemming*: este error se comete cuando dos palabras con diferentes *stems* se reducen a una misma raíz. De esta forma, se pueden llegar a resultados sin sentido o a una misma raíz para dos palabras con significados muy dispares. También recibe el nombre de un falso positivo.

  Ejemplo: ejercían (ejerc-ían), ejército (ejérc-ito) → = raíz (ejerc)

- *Understemming*: este error se comete cuando no se logra llegar a la raíz que comparten. También se conoce como falso negativo.

  Ejemplo: delito (del-ito), delincuente (delincu-ente) → ≠ raíz

Pues bien, como refleja la figura 1, normalmente los algoritmos de *steeming* suelen clasificarse en tres grupos: métodos de truncamiento, métodos estadísticos y métodos mixtos (Jivani, 2011).

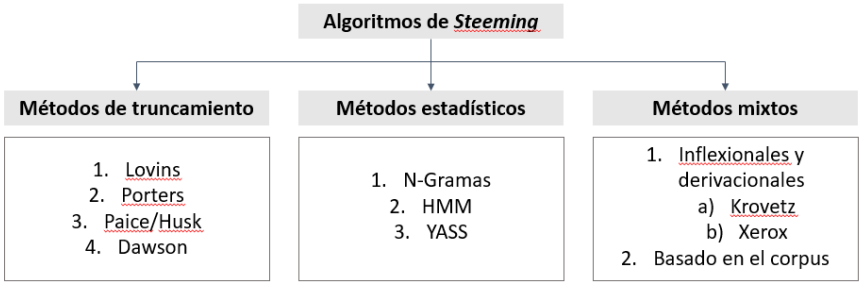

Figura 1. Algoritmos de Steeming. Fuente: elaboración propia a partir de Jivani, (2011)

- Métodos de truncamiento (eliminación de afijos): como su nombre indica, esta técnica está relacionada con la eliminación de sufijos y prefijos (comúnmente conocidos como afijos). Entre estos métodos nos encontramos con el *Porters Steemmer*, uno de los algoritmos más populares. Se basa en la idea de que los sufijos en inglés (aproximadamente 1200) se forman a partir de grupos de sufijos más pequeños y simples. Se trata de un proceso de 5 pasos en el que se van aplicando condiciones, de forma que, si una regla es aceptada, el sufijo se elimina y al finalizar el quinto paso se devuelve la raíz resultante. Porter diseñó este

marco, denominado "*Snowball*", con el fin de que los programadores pudieran desarrollar *stemmers* para otros conjuntos de caracteres y lenguajes.

- Métodos estadísticos: estos mecanismos también eliminan los sufijos y prefijos, pero aplicando distintos métodos estadísticos. Entre ellos encontramos el N-grama, el HMM *Stemmer* y el YASS *Stemmer*

- Métodos mixtos: estos métodos son una combinación de los anteriores, pero en este caso, los *stemmers* se aplican sobre todo el corpus.

1.2.3 Lematización

Se trata de un proceso que consiste en encontrar el lema de la forma flexionada correspondiente. Existen dos normalizaciones lingüísticas según las flexiones: nominales (género, número, etc.) o verbales (tiempo, modo, etc.). De esta forma, el lema constituye la forma más simple de una palabra: en singular para sustantivos, masculino singular para adjetivos e infinitivo para verbos. La lematización utiliza el análisis morfológico para determinar el tipo de palabra y aplica reglas en consecuencia para obtener la base. En este sentido, se aplica el *Part of Speech (PS) tagging* para asignar a cada *token* su categoría gramátical correcta (adjetivo, sustantivo, verbo, conjunción, etc.)

Ejemplo: [menores → menor (sustantivo); cometió → cometer (verbo)]

1.2.4 Otras técnicas

En general, existen otros métodos de pre-procesamiento de textos que incluyen la eliminación signos de puntuación (.,""), de caracteres especiales (@), de números ("2.1") o de caracteres duplicados ("aaaa") o de letras aisladas ("a"). Asimismo, también pueden enfocarse en la transformación de mayúsculas en minúsculas o en el procesamiento de acrónimos (C.A.T -> ¿cat?) o de cuestiones idiomáticas (color <->colour), entre otros.

1.3 *Bag of words*

Para el entender el siguiente concepto no hay que perder de vista cuál es el objetivo principal del *Text Mining*. Estas técnicas se utilizan para transformar un conjunto de documentos (cada uno con sus propias peculiaridades) en unos datos entendibles para los algoritmos de *Machine Learning*. De esta forma, el *"Bag of Words"* se entiende como aquel método que trata a cada documento del corpus como un vector cuya longitud es igual al vocabulario del corpus y que ignora gramática, orden de las palabras, estructura y puntuación.

Existen varios tipos de representaciones del *Bag of Words* y expondremos tres de ellas a continuación. Para un mejor entendimiento de cómo funciona cada uno, utilizaremos el corpus ya comentado tras la eliminación de *stopwords:*

- Documento 1: "menores", "ejercían", "prostitución"
- Documento 2: "cometió", "delito", "prostitución", "delito", "corrupción", "menores"

De esta forma, la lista de vocabulario estaría formado por 9 palabras, 3 de ellas repetidas: **"menores",** "ejercían", **"prostitución", "cometió", "delito", "prostitución"," delito"** "corrupción", **"menores",** y por ello la longitud del vector es 6 ya que entre ambos textos existen ese número de términos distintos.

1.3.1 Modelo binario

Un análisis básico basado en el Modelo binario pasaría por entender que cada palabra es un *token*, de forma que se asignaría un 1 o un 0 en función de si ese *token* está presente en un documento (véase en Figura 2). En otras palabras, cada elemento de un vector codificado con este método refleja la presencia o ausencia del token en el texto.

	menores	**ejercían**	**prostitución**	**cometió**	**delito**	**corrupción**
D1	1	1	1	0	0	0
D2	1	0	1	1	1	1

Figura 2. Representación del modelo binario. Fuente: elaboración propia

- Documento 1 [1,1,1,0,0,0]

- Documento 2 [1,0,1,1,1,1] (aunque la palabra delito aparece dos veces en el documento, con este sistema de representación solo se marca con un "1")

1.3.2 Term Frequency

Este tipo de representación, comúnmente utilizada, consiste en realizar un recuento de palabras (frecuencia) de forma que nos permita diferenciar el número de veces que aparece una palabra en concreto, indicándolo de forma numérica. Por tanto, la frecuencia mide la prevalencia de un token en un documento.

	menores	**ejercían**	**prostitución**	**cometió**	**delito**	**corrupción**
D1	1	1	1	0	0	0
D2	1	0	1	1	**2**	1

Figura 3. Representación del modelo TF. Fuente: elaboración propia

- Documento 1 [1,1,1,0,0,0]

- Documento 2 [1,0,1,1,2,1] (en este caso la palabra delito se marca con un "2" ya que aparece dos veces en el texto

1.3.3 TF-IDF

No obstante, puede ser relevante estudiar lo común que es el término en todo el corpus. En este sentido, el TF-IDF constituye otra representación del *Bag of Words* y se puede definir como la estadística numérica que pretende incorporar la importancia de una palabra en un documento de un corpus.

El TF-IDF, como indica su nombre, está compuesta por las dos frecuencias siguientes:

1. TF (*Term frequency*): como ya se ha explicado representa la relevancia que tiene un término t dentro de un documento d. Para normalizar el TF, se divide entre la longitud del documento: $tf\ (t,d) = \frac{f(t,d)}{longitud\ (d)}$ El numerador representa el número de veces que el término 't' aparece en el documento 'd'. Por lo tanto, cada documento y término tendrá su propio valor TF.

   Siguiendo el mismo ejemplo anterior, calcularemos las frecuencias: TF para la palabra "delito" (número de veces que se repite la palabra "delito" en el documento 2) /(número de términos del documento 2) = 2/6

   A modo de resumen se calculan las frecuencias para todos los términos:

	Doc 1	**Doc 2**	**TF (Doc 1)**	**TF (Doc 2)**
Menores	1	1	1/3	1/6
Ejercían	1	0	1/3	0/6
Prostitución	1	1	1/3	1/6
Cometió	0	1	0/3	1/6
Delito	0	2	0/3	2/6
corrupción	0	1	0/3	1/6

Figura 4. Ejemplo del cálculo peso TF (d,f). Fuente: elaboración propia

2. *Inverse Document Frequency* (IDF): esta ecuación mide cómo de común es un término dentro del corpus. De esta forma, si un término aparece en pocos documentos, será más significativo en los documentos en los que aparezca.

   Existen variaciones en cuanto a la fórmula de cálculo del IDF, pero esta es la que se utiliza en el programa R studio y es la que implementaremos en el siguiente apartado.

$$idf\ (t) = \log(\frac{n}{n'})$$

   El numerador mide el número de documentos que tiene el corpus y el denominador el número de documentos donde aparece el término 't'. Siguiendo con nuestro ejemplo, mostramos una tabla resumen con las frecuencias inversas:

	Doc 1	Doc 2	IDF
Menores	1	1	Log(2/2)=0
Ejercían	1	0	Log(2/1)=0.30
Prostitución	1	1	Log(2/2)=0
Cometió	0	1	Log(2/1)=0.30
Delito	0	2	Log (2/1)=0.30
corrupción	0	1	Log (2/1)=0.30

Figura 5. Ejemplo del cálculo peso IDF. Fuente: elaboración propia

A modo de explicación, un IDF que aparece en la mayoría de los documentos tendrá un IDF bajo. Aunque se han eliminado, lo normal es que las *stopwords,* al ser términos muy comunes, tengan un valor bajo. Por el contrario, un IDF alto supone que existe poca coincidencia y por lo tanto que ese término tiene una mayor importancia.

Como último paso se pueden computar ambas frecuencias TF-IDF para cada palabra en el corpus. Aquellas que obtengan un score más alto serán más importantes y aquellos con valores bajos menos: $Tf - idf\ (t,d) = tf\ (t,d)\ x\ idf\ (t)$

A continuación, en la Figura 6 se muestra a modo de ejemplo el cálculo de los pesos TF-IDF:

	Doc 1	Doc 2	IDF	TD-IDF(D.1)	TD-IDF(D. 2)
Menores	1	1	Log(2/2)=0	0	0
Ejercían	1	0	Log(2/1)=0.30	0.03	0
Prostitución	1	1	Log(2/2)=0	0	0
Cometió	0	1	Log(2/1)=0.30	0	0.05
Delito	0	2	Log (2/1)=0.30	0	0.1
corrupción	0	1	Log (2/1)=0.30	0	0.05

Figura 6. Ejemplo del cálculo de pesos TF-IDF. Fuente: elaboración propia

Palabras como “menores” o “prostitución” son poco relevantes porque aparecen en ambos documentos. Otras como “cometió” o “ejercían” son más relevantes porque solo aparecen

una vez en un texto. Finalmente, la palabra delito, que aparece solo en el documento 2, se repite además 2 veces.

3. *TOPIC MODELING* COMO HERRAMIENTA DE *TEXT MINING*

2.1 Introducción al *Topic Modeling*

El *Topic Modeling* se define como un conjunto de algoritmos encuadrados dentro del *Data Mining*, concretamente del *Text Mining*, y tiene como objetivo principal encontrar temas ocultos dentro del corpus. De esta forma, Blei (2012) explica que "el *Topic Modeling* toma una colección de textos como entrada, descubre un conjunto de *topics* (temas recurrentes que se discuten o comentan en la colección de documentos) y el grado en que cada documento exhibe esos temas". Este investigador es uno de los más importantes en el área con aportaciones continuas desde 2003 hasta 2017 y con más de 200 publicaciones sobre el tema.

En los últimos años, estas técnicas han ido ganando popularidad debido a la proliferación de los textos en formato digital, disponibles online para su consulta. Además, estos algoritmos no solo son capaces de extraer temas de documentos, sino también de canciones o incluso de imágenes (Laitonjam et al., 2015).

Existen varias maneras de clasificar los algoritmos de *Topic Modeling*, pero en el presente trabajo se utilizará la clasificación de Kherwa et al., (2019), que diferencia entre modelos probabilísticos y no probabilísticos:

- No probabilísticos: estos modelos se basan en la factorización de matrices algebraicas y surgieron en los años 90 con el concepto del *Latent Semantic Analysis and Non-Negative Matrix Factorization*. Ambos algoritmos funcionan en base al *Bag of Words*, donde el corpus se convierte en un documento matriz y el orden de las palabras no se tiene en cuenta.

- Probabilísticos: este modelo surgió para mejorar los modelos algebraicos, como *Latent Semantic Analysis*, añadiendo la probabilidad y utilizando enfoques de modelos generativos.

El siguiente nivel de clasificación se centra en diferenciar entre modelos supervisados o no supervisados. De esta forma, tanto el PLSA como el LDA (modelos probabilísticos) eran algoritmos no supervisados, pero posteriormente muchos investigadores han trabajado en el LDA con modelos supervisados.

Finalmente, el último nivel se basa en considerar la secuencia de palabras durante el proceso de *Topic Modeling*. Hasta el 2006, todos los modelos de *Topic Modeling* se centraban en el *Bag of Words* pero Hanna Wallach (2006) introdujo la importancia de incorporar secuencias de palabras a través de *n-grams* y el modelo "*Hierarchical Dirichlet Bigram*" con resultados más precisos. No obstante, aunque este último enfoque sigue siendo desarrollado, por ahora la mayoría siguen basándose en el *Bag of Words*.

Pues bien, analizando más en detalle los algoritmos más importantes, en primer lugar, encontramos el *Latent Semantic Analysis* (LSA), que se trata de un método basado en la descomposición de valores simples (SVD). En 1997 Laundauer and Dumais propusieron esta técnica centrada en una hipótesis distributiva que afirma que los términos con un significado similar también se encuentran muy cerca en su uso contextual. En este sentido, utiliza la representación vectorial del texto para calcular la similitud entre los textos en *clusters* semánticos. Este algoritmo ha sido de gran uso en el área del *Text Mining*, sobre todo en la calificación automática de ensayos, la recuperación de información, el análisis de redes sociales o en el resumen de textos.

En segundo lugar, nos encontramos con el *Probabilistic Latent Semantic Analysis* (PLSA), también conocido como *aspect model* y se trata de una técnica de reducción de dimensionalidad. Basada en el *Bag of Words*, se utiliza para detectar la co-ocurrencia semántica de palabras utilizando la probabilidad en el corpus. Este método fue introducido por Jan Puzicha y Thomas Hoffman en 1999 con el fin de superar algunas desventajas que presentaba el algoritmo anterior. El objetivo principal fue identificar y distinguir entre los diferentes contextos de las palabras sin tener que acudir a un

diccionario. De esta forma, cada palabra se genera a partir de un único tema y diferentes palabras dentro de un mismo documento pueden ser generadas por diferentes *topics.* No obstante, es un modelo ciertamente incompleto, ya que cada documento se representa como una lista de números que indican la distribución de los tópicos, pero no existe un modelo probabilístico generativo de estos números a priori.

Por último, la aparición del *Latent Dirichlet Allocation* (LDA) surge por la necesidad de mejorar los modelos que capturan la intercambiabilidad de las palabras y los documentos, a partir de los modelos más antiguos de PLSA y LSA. Este algoritmo, que es el más utilizado en el *Topic Modeling* y que servirá de base para nuestro análisis práctico, será desarrollado con más detalle en el siguiente punto.

El algoritmo LDA, que fue introducido por Blei et al., (2003), sentó un precedente en el análisis del *Topic Modeling* ya que la mayor parte de los estudios posteriores lo tomaron como punto de partida. Un año más tarde, Rosen et al., (2004) publicaron el primer trabajo donde se incluían metadatos al algoritmo LDA. En 2005, McCallum et al., adaptaron este algoritmo para corpus con origen en cartas, emails y posteriormente en 2007, a redes sociales.

2.2 Aplicación del algoritmo LDA

El algoritmo LDA es uno de los algoritmos más simples, pero a la vez más utilizados. Se trata una técnica de estadística Bayesiana de inferencia donde un conjunto de obversaciones se explica por un conjunto de variables latentes (no observadas), mostrando la similitud entre ciertos datos. Por ello, el reto de las técnicas de *Machine Learning* se centra en desvelar cual es la estructura implícita/latente de temas detrás del corpus, tomando como referencia los documentos observables. Los modelos bayesianos sirven para identificar como se deben modificar las probabilidades subjetivas cuando aparece información nueva.

Uno de los avances que realiza Blei et al., (2012) con este nuevo algoritmo es que remarca que en el modelo anterior de PLSA Hoffman no tenía en cuenta el peso de los documentos. De esta forma, permite equilibrar los resultados, generando una tensión entre las palabras del vocabulario que pertenecen a cada *topic* versus el *topic* al que pertenecen las palabras de documento.

Pues bien, este algoritmo pertenece al campo del Modelado probabilístico, de forma que necesita de un modelo generador que cree valores de forma aleatoria a partir de datos observables. De este modo, se crea una función de densidad de probabilidad formada por datos observados y parámetros ocultos y cuyo objetivo radica en desvelar cual es la probabilidad condicional de esos valores latentes. Por ello, cada documento del corpus está representado por una distribución de probabilidad sobre un número finito de topics y estos a su vez también están representados como una distribución de probabilidad sobre un vocabulario concreto.

Para explicarlo de una forma más intuitiva, pensemos en un corpus con documentos de tres áreas temáticas muy distintas. De esta forma, si queremos modelarlo, el tipo de distribución que buscamos será una que pondere mucho un tema específico y que en cambio no de mucho peso al resto:

- Documento 1: **90**% topic A, 5% topic B, 5% topic C
- Documento 2: 5% topic A, **90**% topic B, 5% topic C
- Documento 3: 5% topic A, 5% topic B, **90**% topic C

A su vez, cada uno de los *topics* tienen una distribución concreta de vocabulario: ***Topic A*** (30% palabra 1, 20% palabra 2, 10% palabra 3, etc.)

Pues bien, para la aplicación correcta del modelo basado en el algoritmo LDA hay que seguir una serie de pasos que se explicarán a continuación.

En primer lugar, debemos entender que el modelo LDA establece una serie de hiperparámetros, que son características de los documentos y *topics* obtenidos de la distribución de Dirichlet. (Blei y Lafferty, 2009):

- η: escalar utilizado en la distribución de Dirichlet que indica la distribución de cada topic en función de las palabras/términos

- $\vec{\alpha}$: vector que se utiliza como parámetro de la distribución de Dirichlet que define la proporción de los temas de cada documento

Pues bien, una vez explicados estos conceptos, el proceso de generación de cada documento se describe con los siguientes pasos (Blei y Lafferty, 2009):

1. **Selección un número de *topics* (*k*):** existen diferentes métodos de calcularlo utilizando distintas medidas (*perplexity*, *topic coherence*, etc.). En concreto en este trabajo se utilizarán las medidas de *topic coherence*, que puntúan un único tema midiendo el grado de similitud semántica entre las palabras con mayor puntuación. De esta forma, estas medidas permiten distinguir entre los temas que son semánticamente interpretables y los temas que son artefactos de inferencia estadística (Kapadia, 2019). Estas medidas pueden ser intrínsecas se utiliza un mismo *dataset* para generar después el modelo, o extrínsecas, que indican que número de *topics* es coherente con referencias externas (Rodrigo, 2016). Una de las medidas más utilizadas es la denominada "*UCI measure*", que, al tratarse de una medida intrínseca, permite averiguar el número de k óptimos utilizando el mismo corpus que para el modelo predictivo.

2. Para cada **topic,** definir cuál es la distribución de las palabras:$\vec{\boldsymbol{\beta}}$**k ~ Dir(η).** Se trata de un valor que es seleccionado aleatoriamente de una distribución de probabilidad de Dirichlet, con parámetro η.

3. Para cada **documento d:**

    a. Definir un vector de proporciones del topic: $\vec{\boldsymbol{\theta}}$**d ~ Dir(**$\vec{\boldsymbol{\alpha}}$**:).** Se genera a través de una distribución de probabilidad de Dirichlet con parámetro $\vec{\boldsymbol{\alpha}}$**.**

    b. Para cada término i:

        i. Escoger el topic para la palabra wd: i ~ **Mult(θd), zd,n ∈** {1, ..., K} (distribución multinomial)

        ii. Escoger la palabra basado en la distribución de la palabra en el tópico: i ~ **Mult(φzd,i), wd,i ∈ {1, ..., V }** (distribución multinomial)

Con la notación de placas (véase Figura 7), muy utilizada para presentar modelos gráficos probabilísticos, se pueden entender de forma más precisa cuales son las dependencias entre variables. De esta manera, la placa exterior representa los documentos y la interior las posiciones de palabras repetidas en un documento. La secuencia sombreada (W) indica que estas son las únicas variables observables y que el resto son variables latentes.

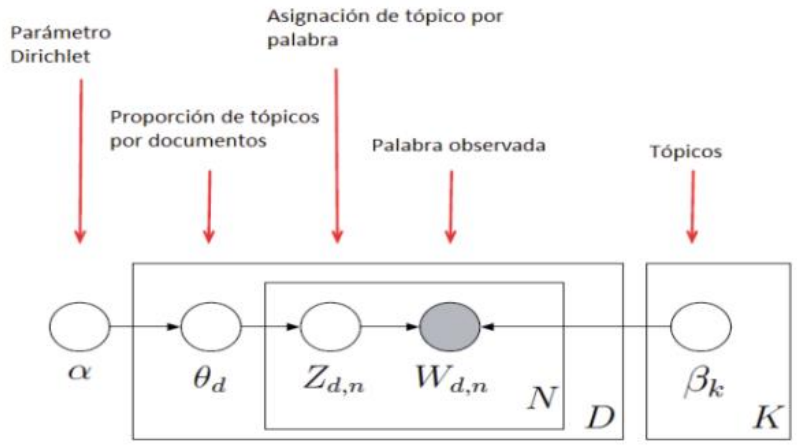

Figura 7. Modelo del algoritmo LDA. Fuente: elaboración propia a partir de Adytana y Nathan 2020

Si analizamos la nomenclatura, podemos entender la realización entre las distintas variables:

- **D:** número especifico de documentos
- **N:** cantidad de palabras existentes en el documento
- ***K:*** número de topics que se seleccionan a priori, por el programador o por un algoritmo
- **βk**: son los topics K de la estructura de topics ocultos
- **θd:k:** lo definimos como la proporción del tópico k en el documento d.
- **Wd,n** : las palabras observadas para el documento d
- **Zd,n**: representa la asignación de topics ocutlos (**βk**) por palabras del documento Wd,n

Pues bien, este modelo asume que en el corpus podemos encontrar dos tipos de estructuras: una oculta de topics y otra de variables observadas. En el proceso generativo, se define una distribución de probabilidad conjunta formada por las variables observadas

y latentes. De esta forma, se computa la distribución condicional de aquellas variables latentes teniendo en cuenta las variables observadas (distribución posterior):

$$p(\theta, z, d \mid \alpha, \beta) = \prod_{i=1}^{N} p(w_i \mid z_i, \beta) \cdot p(z_i \mid \theta_d) \cdot p(\theta_d \mid \alpha)$$

$$p(d \mid \alpha, \beta) = \int p(\theta_i \mid \alpha) \cdot \left[\prod_{i=1}^{N} \sum_{z_i \in Z} p(w_i \mid z_i, \beta) \cdot p(z_i \mid \theta_i) \right] d\theta_i$$

Ahora bien, aunque el algoritmo LDA es aparentemente simple, computacionalmente la inferencia exacta es complicada debido al total de posibles distribuciones. Por ello, suelen utilizarse algoritmos de inferencia aproximada, como el Gibss Sampling. El funcionamiento que esconde consiste en la obtención de secuencia de ciertas observaciones que son aproximaciones de una distribución de probabilidad multivariable. Esta distribución después es utilizada como método de aproximación de la distribución conjunta.

CAPÍTULO V: APLICACIÓN PRÁCTICA

En este último capítulo se expondrá la parte práctica del trabajo y, en concreto, la aplicación de las técnicas de *Machine Learning* para analizar sentencias sobre el delito de trata de menores y su comparativa con el delito de prostitución y corrupción de menores. Para este análisis se utilizará el programa *R studio*, que cuenta con multitud de paquetes que nos permitirán analizar las herramientas necesarias para la obtención de conclusiones de valor.

Por ello, se comenzará detallando cuál es el *Data Set* y cuáles son las sentencias que lo configurarán. Posteriormente, se aplicarán las técnicas de *Text Mining* y, por último, el trabajo se centrará en la implementación del algoritmo LDA ya explicado, con el fin de comprobar en qué forma el *Topic Modeling* puede servir para identificar los *topics* detrás de cada tipo de sentencia.

En una primera iteración, analizaremos y compararemos por separado los resultados del delito de trata y, por otro lado, el conjunto de sentencias de prostitución y corrupción.

Como segundo paso, analizaremos de forma conjunta todas las sentencias, creando un único corpus; primero, para observar la similitud entre los documentos y, posteriormente, para aplicar las técnicas de *Topic Modeling*. Para la explicación de los pasos generales aplicables a todo el análisis, utilizaremos el corpus de Trata de menores a modo de ejemplo, aunque aplicará de la misma manera para la creación del resto de conjunto de documentos.

1. ANÁLISIS GENERAL: *TEXT MINING*

1.1 Preparación inicial de los datos

1.1.1 Obtención de los datos

Como hemos comentado en apartados *supra*, el corpus está formado por distintos documentos. Para este análisis, la base de datos que utilizaremos para la creación del corpus vendrá configurada por 12 sentencias sobre el delito de trata de menores y 12 sobre los delitos de prostitución y corrupción de menores (6 de cada uno). Para la obtención de estas sentencias, se ha acudido a la base de datos del Consejo del Poder Judicial (2022), en la que -a través de sus buscadores propios- hemos podido obtener con precisión las sentencias de cada tipo en formato PDF.

1.1.2 Instalación y carga de paquetes

Antes de comenzar el análisis, es necesario instalar y cargar una serie de paquetes en el entorno R para que, posteriormente, puedan realizarse los análisis correspondientes de los datos contenidos en los textos. A continuación, se muestra concretamente la lista de los paquetes instalados para el análisis práctico del trabajo:

- **Ggplot2:** permite crear diferentes gráficos y formatearlos a través de las varias opciones que incluye el paquete

- **Quanteda:** este paquete permite realizar análisis cuantitativos de textos, desde el procesamiento hasta la creación del corpus, y desde las matrices, hasta la construcción de las gráficas que muestran los resultados. Incluye también los

paquetes quanteda.textstats () y quanteda.textplots() que también utilizaremos a lo largo de nuestro análisis práctico.

- **Readtext:** paquete que permite leer datos del texto casi en cualquier formato (texto, csv, Excel, PDF, etc.)

- **TopicModels:** paquete concreto para la implementación de las técnicas de *Topic Modeling*, concretamente del algoritmo LDA

- **Tidytext:** paquete concreto para la implementación de técnicas de *Text Mining*, que proporciona funciones que ayudan trabajar con textos y con otros paquetes de minería de datos existentes

- **Dplyr:** paquete que reúne un conjunto de funciones que permiten la manipulación de marcos de datos de una manera fácil y rápida

- **UTF8:** paquete que permite almacenar caracteres en un formato estándar

- **Udpipe:** este paquete se utiliza para el procesamiento del lenguaje natural y permite realizar procesos de *tokenización*, de etiquetado de partes de oraciones o la lematización de los textos

- **Text2vec:** paquete creado para la vectorización de textos y el modelado (LDA, LSA) y también se utiliza para el cálculo de las medidas de coherencia en R

1.1.3 Importación de los datos

Para importar los datos al entorno de R, y debido a la diversidad de formatos en los que hoy en día se encuentran los textos, R cuenta con varios tipos de lectores de textos concretos para archivos .doc, .pdf, .xml, etc. No obstante, el paquete *readtext* permite leer el tipo de texto que se le indique y en nuestro caso, ya que las sentencias se descargan en formato pdf desde la base de datos del CENDOJ, habrá que indicárselo.

Pues bien, como primer paso, aplicando la sentencia *list.files(pattern = "pdf$")*, logramos listar los documentos en formato pdf. De esta forma, se seleccionan los documentos presentes en el directorio de trabajo y se crea un vector formado por el número de documentos introducidos (12 para cada uno de los conjuntos). Finalmente, tras listar los pdfs, los importamos a R utilizando la sentencia *"readtext"*.

```
trata_de_menores <- list.files(path = pdf_path, pattern = 'pdf$',  full.names = TRUE)
trata_de_menores <- readtext(trata_de_menores)
```

1.1.4 Creación del corpus

Para la creación de cada uno de los corpus, utilizamos la función *Corpus()*, procedente del paquete "quanteda". De esta forma, crearemos dos corpus distintos, formado cada uno por 12 documentos. A través de la función *summary* podemos analizar la estructura y el contenido de nuestros corpus, donde se indica el nombre de cada texto, el número de *tokens* y de frases por cada documento.

```
c_trata_de_menores<-corpus(trata_de_menores)
summary(c_trata_de_menores)
```

1.1.5 Proceso de tokenizacion y creación de la matriz TDF

En las versiones más antiguas de R, la función *dfm*, que proviene del paquete quanteda y que crea la matriz de frecuencias que después comentaremos, *tokenizaba* los documentos del corpus y los convertía a minúsculas. No obstante, las nuevas versiones han separado los procesos, indicando que se debe realizarse *a priori* el proceso de *tokenizacion* para posteriormente crear la matriz.

Por ello, aplicamos la función *tokens* al corpus creado previamente y comenzamos a limpiar el resto. En primer lugar, le indicamos que elimine los números que aparezcan en el texto con la función *remove number=TRUE* ya que en las sentencias aparecen muchos números que no aportan información relevante, como el de la misma sentencia, el del recurso, el del código de búsqueda de la base de datos del CENDOJ, etc.). Asimismo, le

indicamos que elimine cualquier tipo de signo de puntuación (comas, puntos, guiones, etc.) con *remove punt=TRUE*, ya que tampoco son relevantes en el análisis de las sentencias (previamente ya hemos realizado una primera limpieza de estos signos).

```
tokens1<-tokens(c_trata_de_menores, remove_numbers = TRUE, remove_punct = TRUE )
```

Una vez *tokenizados* nuestros documentos, podemos crear nuestra matriz TDM con la función propia de quanteda *dfm.* Utilizando la función *dim* podemos ver el número total de *tokens* que forman nuestra matriz inicialmente, habiendo detectado en el caso de las sentencias sobre trata de menores 9487 términos. Si aplicamos la función *topfeatures* nos muestra los 10 términos más repetidos que, en este caso, no aportan ninguna información al tratarse de *stopwords:* "de", "la", "que", "en", "el", "y", "a", "por", "las", "del". Por ello será necesario seguir realizando una limpieza de los datos en mayor profundidad, para así quedarnos solo con aquellos términos que sí nos aportan valor en el análisis

```
midfm1<-dfm(tokens1)

dim(midfm1)

topfeatures(midfm1)
```

1.1.6 Limpieza de los datos

En primer lugar, cabe mencionar que, como ya se ha comentado, hay funciones que se han ido quedando obsoletas con las nuevas versiones de R y este es el caso de *"remove" o "select",* que anteriormente se utilizaban para el preprocesamiento de los datos. De esta forma, utilizaremos las nuevas funciones de quanteda *dfm_remove* y *dfm_select,* donde le indicaremos qué tipo de limpieza queremos realizar.

Como primer paso, quitaremos todos los ordinales y las letras aisladas, ya que en las sentencias aparecen muchas palabras formadas por una o dos letras que rara vez aportan información de valor y aparecen como códigos asociados al documento.

Posteriormente, le indicaremos que elimine las *stopwords* que, como ya explicamos en puntos anteriores, son aquellas palabras que resultan comunes en un cierto lenguaje pero que no aportan información, como preposiciones, conjunciones, etc. En este caso, tendremos que indicarle que nuestro vocabulario es el español para que pueda acceder al mismo.

No obstante, si observamos los *topfeatures* después de esta iteración, encontramos muchos términos que tampoco aportan valor pero que en general no se reconocen como *stopwords*. Estos términos suelen ser aquellos que aparecen en el encabezado de las sentencias, que como se indicó no aportan utilidad para nuestro análisis, o bien palabras propias del lenguaje jurídico que repiten a lo largo de los textos. Por ello, creamos nuestra propia lista de *stopwords* y le indicamos que las elimine.

Como último paso, se realiza el proceso de *steeming* que, como ya se explicó, reduce las palabras a su raíz. Al finalizar todo el proceso de limpieza del texto, llegamos a reducir el número de términos de 9487 a 4346.

```
midfm1<-dfm_select(midfm1, pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-zA-Z]\\b" ), selection = "remove", valuetype = "regex")

midfm1<-dfm_remove(midfm1, pattern = stopwords(language=”spa”)

midfm1<-dfm_remove(midfm1, pattern = c("año","artículo","delito","sentencia", "num", "art", "tribunal",”delito”))

midfm <- dfm_wordstem(midfm1, language = "spa")
```

1.2 Proceso analítico

1.2.1 Gráficos de frecuencias

Con nuestros corpus creados y tras la incorporación de nuestras matrices TDM preprocesadas, podemos proceder al análisis de los resultados. Para ello, crearemos un gráfico con R para explicar de una manera más visual las conclusiones que se pueden obtener. Al estar utilizando el paquete *quanteda*, con las nuevas actualizaciones debemos llamar a la librería previamente mencionada: *quanteda.textstats*.

Utilizando la función *textstat_frequency* creamos una matriz de dimensiones 20X5 donde podemos observar cuáles son las 20 palabras más repetidas, su frecuencia exacta en el total de documentos, su ranking, la frecuencia de documentos en el que sale y su grupo correspondiente. Resulta sobre todo interesante observar la columna de *"docfreq"* ya que a través de *ggplot* podemos crear un gráfico que nos muestre la información sobre las frecuencias:

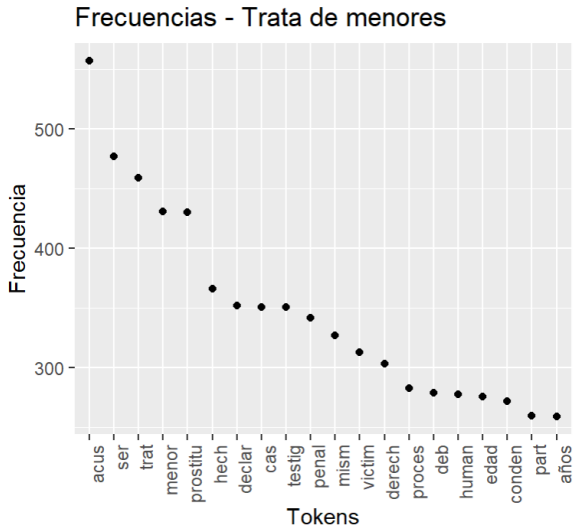

Figura 8. Gráfico de frecuencias sobre trata de menores. Fuente: elaboración propia

Pues bien, como primeras conclusiones sobre el delito de trata de menores podemos observar que las palabras que más se repiten son "acusado", "ser", "trata", "menor" y "prostitución". Aunque en un primer momento podría parecer que este análisis no aporta mucha información, si analizamos las palabras podemos extraer conclusiones relevantes. Por un lado, una alta frecuencia en la palabra "acusado" nos indica que en este tipo de sentencias se pone mucho énfasis en la parte del delincuente, a lo mejor enfocándose mucho en el perfil de este. Además, una alta repetición en la palabra menor podría indicar que en este tipo de sentencias se da una gran importancia al hecho de que las víctimas sean menores. Por su parte, que exista una alta repetición de la palabra "trata" indica que los motores de búsqueda utilizados para la creación de nuestra base de datos han funcionado correctamente.

Por otro lado, los dos términos que *a priori* nos pueden aportar más información son el de "prostitución" y "testigo". Por un lado, y como ya explicamos en el primer punto, el

delito de trata de menores puede tener distintas finalidades: de explotaciones sexual, laboral, para la realización de actividades delictivas, para la extracción de órganos o para la celebración de matrimonios forzosos. No obstante, resalta la gran incidencia de este término, relacionado claramente con actividades con fines de explotación sexual. Como primer enfoque, podríamos entender que el reducido número de sentencias sobre trata de menores se debe en parte a que no se están instruyendo casos en los que se utilizan a los menores para finalidades distintas a la prostitución. De hecho, a través de la tabla de frecuencia mencionada, podemos observar que dicho término aparece en 9 de los 12 documentos. Por otro lado, también nos puede indicar que en la mayoría de las sentencias existe un concurso medial entre el delito de trata y el de prostitución. Esto quiere decir, que el delito de trata se comete como medio para la consecución del delito de prostitución y que no se castiga doblemente.

Con relación a la palabra "testigo", también llama la atención su gran frecuencia, lo que pueda indicar que en este tipo de delitos es muy común que haya testigos y que su testimonio tenga gran peso en el fallo. Habrá que seguir investigando para ver qué información adicional nos puede aportar.

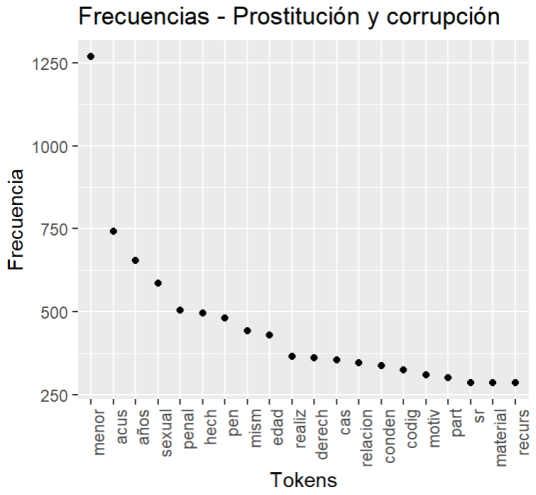

Figura 9. Gráfico de frecuencias sobre prostitución y corrupción de menores. Fuente: elaboración propia

Si observamos las palabras más frecuentes de las sentencias de prostitución y corrupción de menores, vemos que algunas de ellas coinciden con el delito de trata como "acusado", que igualmente se repite con una alta frecuencia. No obstante, también podemos delimitar ciertas diferencias. En primer lugar, la palabra "menor" es el término más repetido, "años" aparece dentro de los cinco términos más repetidos y "edad" dentro de los diez. Esto

quiere decir que, en este tipo de casos, se le da gran importancia a la edad de la víctima, ya no solo especificando que se trata de un menor, si no también cuál es su edad concreta. Esto se puede deber a que, en estos delitos, cuando el menor tiene menos de 16, al delincuente se le impone una pena mayor. De esta forma, a lo mejor en estos casos, cuándo el menor no ha cumplido los 16 años, se opta por calificarlos por este tipo de delitos porque el marco punible es mayor: en el delito de trata se puede llegar a imponer hasta 8 años y en el de corrupción (pornografía) hasta 9.

Otro de los puntos a destacar es que no aparece la palabra "testigo", que en el caso anterior tanto se repetía. Esto puede indicar que, en aquellos casos en los que la trata de menores se estuviera produciendo sin ningún tipo de constancia externa, podría abogarse por una explotación sexual en la que el menor mantiene un vínculo más privado con el acusado. Por último, también destaca que no se repiten con gran frecuencia las palabras "prostitución" o "corrupción" y quedan recogidas bajo el carácter de "sexual". Esto puede deberse a que bajo estos conceptos se pueden producir delitos sexuales de pornografía infantil o exhibicionismo de menores, por lo que podría concluirse que solo en aquellos casos claros en los que existe prostitución se estaría calificando como concurso medial con el delito de trata de menores. En el resto de los casos, se podrían estar calificando como delitos de prostitución o de corrupción de menores en general.

1.2.2 N-Gramas: bigramas

Un análisis algo más concreto consistiría en analizar los N-gramas. Se denomina N-grama a una secuencia de n elementos y en concreto bigrama a la secuencia de 2 términos. De esta forma, podemos observar por cada corpus cuáles son las secuencias verbo-sustantivo, sustantivo-sustantivo, sustantivo-adjetivo, etc, que más se repiten. Posteriormente, al igual que hemos realizado en el punto anterior con los términos individuales, crearemos una gráfica que muestre los resultados, esta vez en forma nube de palabras.

En primer lugar, para crear bigramas, debemos realizarlo en nuestro proceso de *tokenización*, indicándole a R que debe separar cada término en secuencias de dos. En este caso, le indicaremos -como en pasos anteriores- que elimine los signos de puntuación, los números, las letras aisladas y las *stopwords*. No obstante, mantendremos las palabras de la lista creada de *stopwords* propias ya que, de forma individual, puede que no nos

aporten valor; pero puede que, de forma conjunta, sí. Asimismo, no realizaremos el proceso de *steeming* ya que carecería de sentido para este análisis. Por último, le indicaremos que queremos extraer los *tokens* en formato bigrama con la función *tokens_ngrams(n=2)*, y crearemos posteriormente nuestra matriz TDM.

Para la creación de la nube de palabras, utilizamos la función *textplot_wordcloud* propia del paquete de quanteda y le indicaremos que represente la nueva matriz formada por los bigramas, cogiendo solo aquellos que se repiten más de 20 veces.

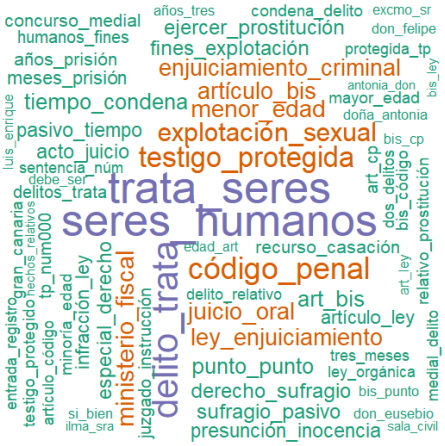

Figura 10. Nube de palabras de la frecuencia de bigramas de trata de menores. Fuente: elaboración propia

Pues bien, de este segundo análisis observamos bigramas propios del delito de trata de menores (“trata_seres”, “seres_humanos”, “artículo_bis”) que no aportan mucha información. En cambio, nos encontramos con “explotación sexual”, volviendo a incidir en el hecho de que es el objetivo mayoritario por el que se comete este delito. Por otro lado, y con relación a lo ya comentado, destaca el bigrama “testigo protegida”, lo que puede indicar que en muchos de estos procesos gente cercana a la víctima es testigo de lo que está ocurriendo (fijándonos en el género podríamos decir que madres o amigas). También podemos concluir que muchos de estos testigos quedan protegidos bajo la ley cuando deciden hablar con las autoridades y contar los delitos que se están cometiendo.

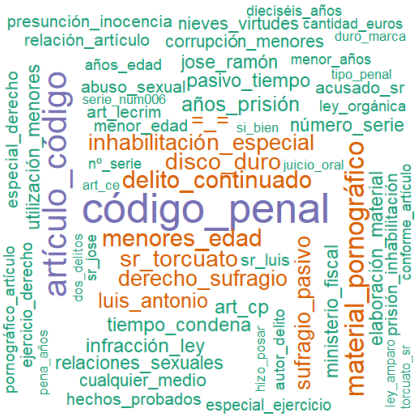

Figura 11. Nube de palabras de frecuencia de bigramas de prostitución y corrupción. Fuente: elaboración propia

Si analizamos el bigrama de los delitos de prostitución y corrupción, las conclusiones son bastantes claras. En primer lugar, reafirma lo ya comentado en el análisis previo de los términos individuales y es que, en estas sentencias, los casos se centran sobre todo en delitos relacionados con la distribución de material pornográfico, no poniendo el foco solo en casos de prostitución. De hecho, no es casualidad que uno de los bigramas que más se repita sea el de "disco duro", ya que en bajo el concepto de corrupción de menores, el artículo 189 pena la producción, venta y distribución de ese material y el artículo 189 bis pena la difusión por Internet o elementos electrónicos de cualquier información que favorezca a la consecución de delitos sexuales contra menores.

Por otro lado, llama la atención que uno de los bigramas más repetidos sea "delito continuado". Si analizamos la definición jurídica del término, extraemos que se trata de aquellos casos en los que existen diversas acciones materiales referidas a un delito semejante y dicha pluralidad de acciones se trata como si se hubiera producido un único hecho. Pues bien, con esta lectura se podría pensar que algunos de los casos de trata de menores con fines de explotación sexual se estarían tratando como delitos continuados, no distinguiendo que uno realmente es un medio de comisión del otro.

Finalmente, hay una serie de bigramas que en este caso no aportan gran información, como son "derecho de sufragio", "inhabilitación especial" o "sufragio pasivo", ya que se trata de castigos asociados a las penas de estos dos delitos.

1.2.3 Lematización

Otro de los análisis que podemos realizar se centra en observar las frecuencias por tipo de palabra. Para ello, realizaremos con R un proceso de lematización, que se define como aquel proceso que, dada una forma flexionada (plural, género femenino, conjunción, etc.), logra encontrar el lema correspondiente. Para realizarlo, partiremos de nuestros dos corpus, ya que -lo explicaremos más adelante- no queremos trabajar sobre la matriz TDM procesada.

Pues bien, para llevar a cabo este análisis utilizaremos el paquete *udpipe* y, a través de las funciones *udpipe_download* y *udpipe_load_model,* le indicaremos al programa que queremos descargarnos un modelo de lematización en español y que lo cargue en nuestro directorio. En caso de que esta opción no funcione correctamente, existe la posibilidad de descargar directamente un modelo concreto, obtenido a través de la consulta *list.files(getwd()).*

Como siguiente paso, y haciendo uso de la función *as_utf8,* convertimos nuestro corpus en un objeto de caracteres UTF-8 válido. Como se ha indicado, trabajaremos sobre el corpus original, que mantiene signos de puntuación, números, *stopwords* y no se ha reducido a su raíz haciendo *steeming*. Esto se debe a que, para que R pueda realizar con mayor precisión el proceso de lematización, debe entender qué tipo de palabra es (verbo, sustantivo, etc.) y mantener el texto con la coherencia inicial le resultará más fácil. A través de la función *udpipe_annotate,* anotamos el resultado de aplicar el modelo cargado a nuestro corpus y posteriormente lo convertimos en *dataframe.*

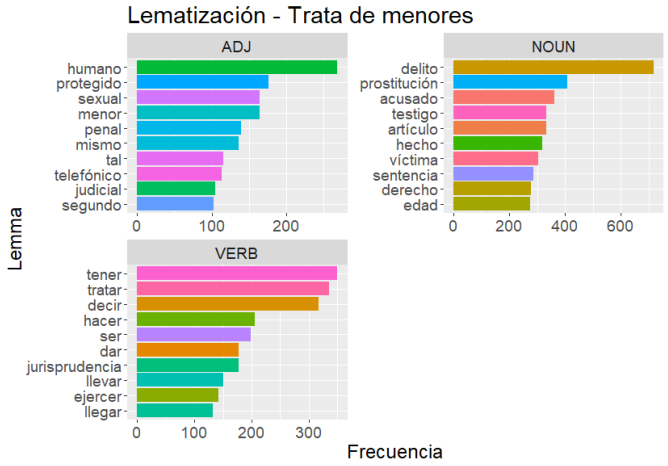

Figura 12. Gráfico de frecuencias por categorías gramaticales de trata de menores. Fuente: elaboración propia

Si observamos los resultados mostrados en la Figura 12 para el delito de trata de menores, podemos concluir que el análisis ha sido bastante preciso. Siempre existen limitaciones dentro de los paquetes y a veces R no detecta con total exactitud, como ha ocurrido con la palabra "jurisprudencia" al calificarla como verbo. No obstante, si ponemos el foco en el resto del gráfico, volvemos a encontrarnos con sustantivos ya analizados, como "prostitución" o "testigo". Esta vez, resulta más interesante poner el foco en los adjetivos y en los verbos. Por un lado, encontramos el adjetivo "protegido", que por análisis previos sabemos que en su gran mayoría se encontrará en género femenino pero que también podría hacer referencia, no solo al testigo, sino también al menor, que muchas veces, tras sufrir un delito de este tipo, queda protegido bajo la ley mediante órdenes de alejamiento. También se suele dar el caso de que los menores queden protegidos por las instituciones públicas que ejercen de forma temporal la patria potestad que recae sobre sus familias.

Por otro lado, el adjetivo "telefónico" es la primera vez que aparece y esto puede ser una indicación de que muchas veces se atraen a los menores por medios tecnológicos como son los dispositivos móviles.

En cuanto a los verbos, la palabra "decir" puede también relacionarse con el análisis anterior y es que muchas veces los delincuentes atraen a sus víctimas a través de promesas que realizan por teléfono, ya que todo lo que queda por escrito tiene mayor valor como prueba. Cuando aparece la palabra "ejercer", nos vuelve a indicar que la mayor parte de los casos de trata tienen como finalidad que la menor termine ejerciendo la prostitución.

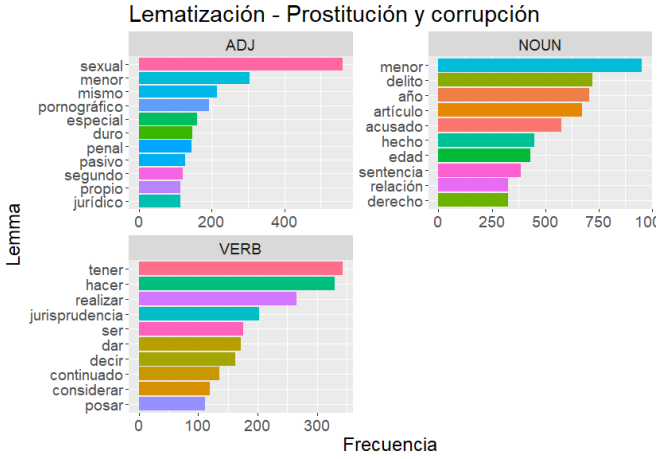

Figura 13. Gráfico de frecuencias por categorías gramaticales de prostitución y corrupción Fuente: elaboración propia

En el caso de la Figura 13, vemos que vuelve a aparecer el término “pornográfico”, y “sexual”, esta vez ya definidos como adjetivos. En la parte de los sustantivos, destacan los temas relacionados con la edad -ya comentados-, así como con la palabra “relación”, que puede indicarnos conclusiones parecidas. En este tipo de delitos, la pena se gradúa en función de la condición del menor por lo que si el acusado tiene una relación de superioridad, ya sea por cuestión de edad o discapacidad de la víctima, la pena impuesta será superior. Por otro lado, también puede referirse a la relación que mantenga el acusado con la víctima de carácter sexual, que muchas veces deriva en el delito de prostitución o corrupción del menor. Por último, de los verbos podemos destacar el término “posar”, que no había aparecido previamente pero que también se relaciona con los delitos de pornográfica infantil, donde el acusado gana dinero con la venta de fotos de las menores.

1.2.4 Comparativa entre documentos: pesos TF-IDF y similitud de coseno

Para el cálculo de similitudes entre los documentos, como es lógico, debemos partir de un corpus que cuente con todas las sentencias, con el fin de poder comparar la cercanía entre unas y otras. Por ello, para ser más eficientes, partiremos de los dos corpus originarios y creamos un tercero *(c_conjunto)* mediante la unión de los dos *(c_conjunto<-c_trata_de_menores+c_prost_corrup).* Al igual que realizamos en pasos anteriores, limpiaremos el corpus, realizaremos el mismo proceso de *tokenización* y crearemos nuestra matriz TDM, aplicando las mismas funciones de limpieza de datos que anteriormente.

No obstante, a pesar de haber creado una matriz de frecuencias, para el cálculo de similitudes elaboraremos una matriz con los pesos TF-IDF. Como ya explicamos en el punto 1.3.3 (Capítulo IV), se trata de otra representación del *Bag of Words* y permite saber cómo de relevante es una palabra para un documento concreto en una colección, más allá de su frecuencia en todo el corpus. Esta medida suele utilizarse en los procesos enfocados a determinar la similitud de los documentos y en nuestro caso utilizaremos la similitud de coseno como métrica de cálculo. Esta medida de similitud mide el coseno de un ángulo entre dos vectores y cada documento se representa como un vector cuya dirección está determinada por un conjunto de valores TF-IDF en el espacio. Hay otras medidas, como la euclídea, que no funcionan bien en estos casos ya que cada documento

tiene a su vez muchos términos, por lo que se trabaja en un espacio de dimensiones muy altas.

Pues bien, para su análisis, transformamos el tipo de pesos de nuestra matriz con la función *dfm_tfidf*. Como siguiente paso, se calcula la similitud entre documentos, haciendo uso de la función *textstat_simil* y le indicamos que el método utilizado es el del coseno. Ambas funciones forman parte del paquete de quanteda, lo que facilita y reduce el tiempo de procesamiento.

Como siguiente paso, y para poder extraer conclusiones de este análisis, crearemos un dendograma clasificatorio, donde R se encargará de agrupar por *clusters* las sentencias. Para ello, utilizamos la función *hclust* que realiza *clustering* jerárquico, procesando las distancias introducidas con el cálculo de similitud.

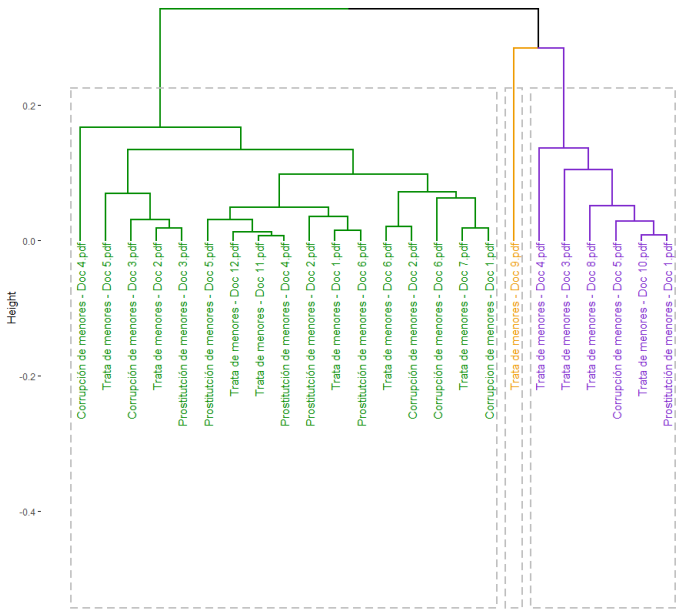

Figura 14. Dendograma clasificatorio según similitudes de coseno

Realizando un análisis más detenido de las similitudes y del dendrograma podemos observar que hay tres *clusters* diferenciados. El tamaño de cada uno difiere ya que el *cluster* 2 solo está formado por 1 sentencia, el 3 por 6 y el 1 por el resto de las 17 sentencias. En el segundo *cluster* vemos que solo encontramos una sentencia del delito de tratas. Un análisis más profundo podría consistir en analizar en concreto dicha sentencia para identificar cuáles son los factores que le han llevado a esta clasificación.

Una de las razones podría ser que fuera una de las sentencias en las que la finalidad de la trata no fuera la explotación sexual, sino cualquiera de las otras mencionadas. Por otro lado, el *cluster* 2 está formado por una mayoría de sentencias de trata, lo que puede indicar que estos documentos ponen mayor foco en "la captura, transporte, traslado" de la víctima en vez de en la finalidad con la que se comete el delito. Por último, las sentencias que conforman el primer *cluster* son la su mayoría de prostitución y corrupción de menores, aunque también aparecen en este grupo sentencias de trata. Estas últimas pueden ser aquellas en las que se condena por un concurso medial con delitos de prostitución o corrupción de menores.

2. ANÁLISIS GENERAL: *TOPIC MODELING*

2.1 Aclaración sobre la base de datos

En el punto anterior ya se ha explicado detalladamente cuál es el proceso de preparación de los datos con las técnicas de *Text Mining*, por lo que no incidiremos más en esa parte. No obstante, para la implementación de las herramientas de *Topic Modeling* creemos conveniente realizar unos apuntes previos.

Por un lado, volveremos a utilizar la base de datos general, es decir, analizaremos todas las sentencias en un mismo corpus por lo que utilizaremos el corpus que creamos para el análisis anterior (*c_conjunto).* No obstante, el análisis de textos a través de *Topic Modeling* necesita una gran base de datos sobre la que trabajar para que pueda identificar correctamente los distintos *topics* y las palabras que lo forman. En este caso, nuestro corpus conjunto está formado por 24 sentencias que no serán suficientes para que el algoritmo funcione adecuadamente. Por ello, con el fin de aumentar nuestro corpus, dividiremos los documentos por párrafos con la función *corpus_reshape*, también del paquete quanteda. De esta forma, se observa que nuestro corpus pasa de estar formado por 24 elementos a 1482 documentos, que en ese caso son párrafos.

Pues bien, una vez que contamos con nuestro nuevo corpus, aplicaremos el mismo proceso de limpieza de textos y crearemos nuestra matriz TDF con la misma función *dfm*,

que utilizaremos como base para crear nuestro modelo de *Topic Modeling*. Como penúltimo paso, realizaremos *pruning* con la función *dfm_trim a* nuestra matriz, de forma que solo nos quedemos con aquellos términos que aparecen más de 20 veces. Finalmente, transformaremos nuestra matriz de documentos-términos a una matriz de documentos-temas, utilizando la expresión: *dtm=convert(midfm4, to="topicmodels")*

2.2 Determinación del número de *k topics*

Como ya se ha explicado en puntos anteriores, la determinación de la k en el modelo, es decir, del número *topics* no es tarea sencilla. Si bien en este caso podríamos decidir indicarlo de forma manual, hemos considerado que un análisis más preciso requería el entrenamiento de un algoritmo.

Es por ello por lo que hemos recurrido al concepto de coherencia y en concreto al paquete *text2vec*, que nos permitirá realizar todo el proceso de cálculo en R. Al tratarse de un paquete nuevo, siempre es conveniente hacer uso de la ayuda que proporciona R y que indica cuáles son los argumentos que deben introducirse. Pues bien, para determinar el número de k a través del cálculo de la coherencia, primero debemos crear un bucle que nos permita ir midiendo el desempeño de distintos números de *topics* en el modelo. Para ello debemos construir nuestro modelo LDA propio del paquete *topicmodeling*, indicando que la k=i y que el método de cálculo utilizado es el de Gibbs. De esta forma, el modelo irá calculando la coherencia con las diferentes k y las irá almacenando en un *dataframe* creado, calculando la media para cada iteración (empezaremos por dos *topics* hasta un máximo de 10, yendo de 2 en 2). De esta forma, seleccionaremos aquella k que obtenga como resultado un mayor valor en la media de la coherencia, ya que ese será el modelo más adecuado para nuestro análisis.

En concreto, para el cálculo de la coherencia, debemos entender qué argumentos debemos introducir y qué métrica implementaremos:

- X: se trata de una matriz formada por los *top* términos de cada *topic*. Para ello, hacemos uso de la función *terms* y le indicamos que obtenga solo los 10 términos

más repetidos y los guarde en *topicTerms,* previamente transformándolo en matriz.

- Tcm (*term-co-occurence matrix*): sirve de base para calcular las métricas de coherencia (*tcm=crossprod(sign(as.matrix(dtm)))*

- Metrics: este paquete cuenta con varias métricas entre las que encontramos "mean_logratio", "mean_pmi", "mean_npmi", "mean_difference", "mean_npmi_cosim", "mean_npmi_cosim2". En nuestro caso, compararemos las métricas "mean_pmi" y "mean_npmi_cosine" que, según uno de los creadores del paquete, son con las que mejores resultados se obtienen (Bickel, 2019).

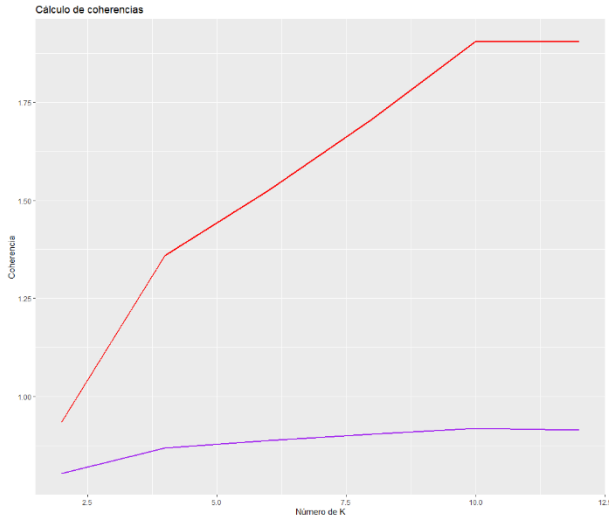

Figura 15. Comparativa de los valores de coherencias. Fuente: elaboración propia

Una vez que hemos puesto en marcha nuestros bucles, observamos que la métrica con la que se obtiene una coherencia más alta es con "mean_pmi" (línea roja). En parte se puede deber a que se trata de una métrica que más semejanza tiene con la del método "UCI", comúnmente empleada en otros programas (como Python) y que -como se ha explicado- se trata de una métrica de coherencia interna. De esta forma, al haber introducido un corpus interno y no haber entrenado el modelo con información ajena, puede que sea esta la razón de su mejor *performance*. En cuanto al número de k, seleccionamos 10 ya que, como indican ambas métricas, en este caso es el número óptimo y a partir de ese valor la coherencia comienza a decrecer.

2.3 Aplicación del modelo y análisis de resultados

Una vez que hemos decidido el número de k, debemos volver a entrenar el modelo LDA, esta vez indicándole que k=10

```
topicModel<- LDA(dtm, method="Gibbs", k=10, control=list(alpha=0.1))
```

El siguiente paso consiste en comparar la frecuencia de cada palabra con respecto a cada *topic* generado y para ello utilizamos la función *tidy* del paquete *tidytext*, generando una matriz que recogerá dichas frecuencias (beta). Esta matriz nos permitirá posteriormente visualizar mediante una gráfica qué palabras tienen mayor presencia en cada *topic*.

```
c_topics<- tidy(topicModel, matrix = "beta")
```

Para crear esta gráfica, primero le indicaremos que agrupe las palabras por *topics*, en base al valor de beta que tengan asignado. Hemos indicado que muestre las 10 palabras más comunes, es decir, con mayor beta y lo recogemos bajo el objeto *c-topics_term*. Como último paso, creamos nuestras gráficas de barras para los 10 *topics* generados entre todas las sentencias.

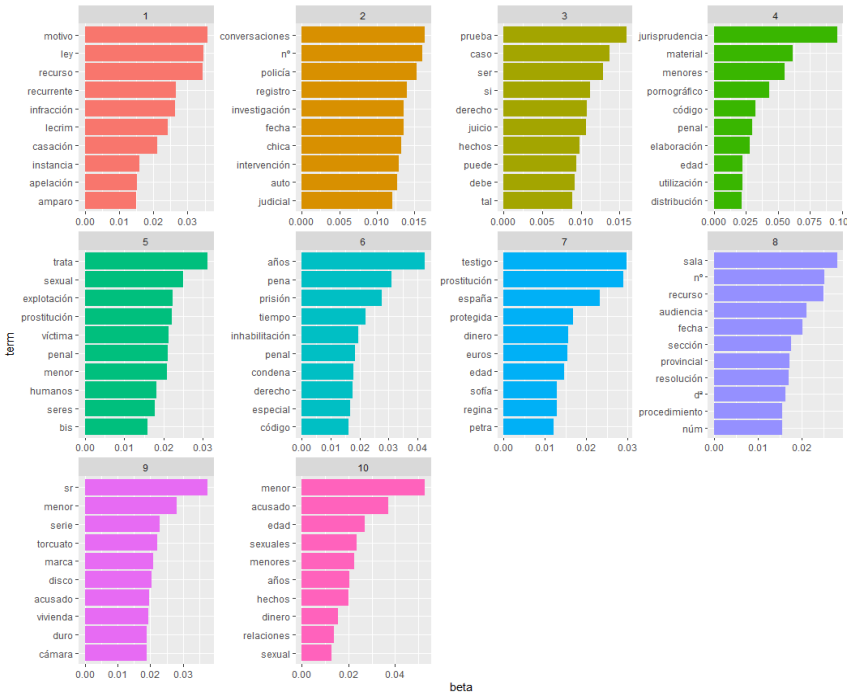

Figura 16. Resultados del modelo LDA. Fuente: elaboración propia

Si analizamos en concreto cada uno de los *topics* generados podemos seguir sacando conclusiones. Así, podemos diferenciar entre *topics* comunes a todas las sentencias y *topics* concretos de cada delito. En la primera categoría encontramos los siguientes *topics:*

- ***Topic* 1:** trata sobre partes materiales del proceso judicial, comunes a todas las sentencias. Los términos que conforman este *topic* son puramente jurídicos como "recurso", "infracción", "recurrente", etc.

- ***Topic* 2:** destacan palabras como "conversaciones", "policía" o "registro", relacionadas con el proceso de recolección de pruebas previa al proceso. Este *topic* también aparecerá en la mayoría de las sentencias, en la parte de antecedentes de hecho.

- ***Topic* 3:** en este *topic* destacan palabras como "prueba", "juicio", "hechos" por lo que podría ser un tema relacionado con las pruebas que se presentan y se enjuician durante el proceso y que se enuncian también en los antecedentes de hecho de la mayoría de las sentencias.

- ***Topic* 6:** las palabras que conforman este *topic* son "años", "pena", prisión", "inhabilitación", "condena", lo que hace referencia a las penas que se imponen por la comisión de los delitos y que suelen incluirse en el fallo.

- ***Topic* 8:** en línea con el *topic* 1, este *topic* recoge palabras que identifican un determinado proceso judicial, indicando la tipología del caso y del tribunal, entre otros: "nº", "sala", "fecha", "sección", etc.

Por otro lado, hay algunos *topics* que pueden identificarse con ciertas tipologías de delitos:

- ***Topic* 4:** bajo este *topic* encontramos palabras como "menores", "pornografía", "distribucion" o "material", indicando que son patrones propios de sentencias que tratan sobre la comercialización de pornografía infantil. Bajo este *topic* también aparece la palabra "edad", que como hemos comentado, solía aparecer en las

sentencias de corrupción de menores para enfatizar las penas superiores en función de la edad de la víctima.

- ***Topic* 5:** en este *topic* encontramos la palabra "trata" con mayor representatividad, pero seguida de los términos "sexual", "explotación" y "prostitución", lo que indica que se trata de sentencias sobre trata de menores, pero solo aquellas que tienen como finalidad la explotación sexual del menor.

- ***Topic* 7:** este *topic* es uno de los más interesantes ya que, por un lado, quedan recogidas palabras como "prostitución" y "testigo"; sin embargo, no se relaciona con el delito de trata, aunque por análisis anteriores, sabemos que este *topic* sería propio de estas sentencias. Además, aparecen nombres propios femeninos y palabras relacionadas con el dinero, lo que indica que las testigos en su mayoría eran mujeres y que podían haber sido extorsionadas económicamente.

- ***Topic* 9:** bajo este *topic* encontramos palabras como "sr", "marca", "disco", "vivienda", "acusado", "duro", "cámara". Observamos que se trata de un *topic* enfocado hacia el perfil del delincuente y a sus actividades delictivitas, en este caso relacionadas con temas de pornografía infantil a través de dispositivos electrónicos.

- ***Topic* 10:** encontramos palabras comunes a los tres tipos de delito como "menor", "sexual", "relaciones", "acusado", etc., por lo que podría decirse que es un *topic* susceptible de ser encontrado en cualesquiera de las sentencias analizadas. No obstante, podemos destacar la palabra "dinero", más común en los delitos de prostitución y corrupción de menores donde la finalidad suele estar enfocada a la obtención de beneficios económicos.

Como último paso calculamos los valores gamma que nos indican el peso de cada *topic* en cada documento. Para obtener estos valores se vuelve a hacer uso de la función *tidy()*, pero en esta ocasión indicando que utilice la probabilidad gamma. Como ya comentamos, nuestros documentos son párrafos de las sentencias y al tener más de 1000 es difícil poder recoger las probabilidades de todos de una forma visual. Por ello, agrupamos por *topics*

y obtendremos de cada uno aquel documento que tiene mayor valor gama (véase Figura 16), es decir, el documento que tiene un mayor peo de cada uno de los *topics*.

```
corp_doc_filter<-corp_doc_filter%>%group_by(topic)%>% slice(which.max(gamma))
```

Documento	Topic	Gamma
Corrupción de menores – Doc 5	1	0.996
Corrupción de menores – Doc 6	2	0.997
Corrupción de menores – Doc 5	3	0.987
Corrupción de menores – Doc 4	4	0.984
Trata de menores – Doc 8	5	0.993
Corrupción de menores – Doc 4	6	0.994
Trata de menores – Doc 2	7	0.886
Trata de menores – Doc 12	8	0.997
Trata de menores – Doc 3	9	0.993
Corrupción de menores – Doc 1	10	0.997

Figura 17. Tabla resumen de valores gamma más altos por cada topic. Fuente: elaboración propia

Como ejercicio final, podemos analizar cuál ha sido el documento con mayor peso para cada *topic.* Por un lado, los *topics* que hemos considerado que podrían ser característicos de todas las sentencias (*topics* 1,2,3,6 y 8) encuentran sus valores gamma máximos en distintas categorías de sentencias. En cambio, si analizamos el resto las conclusiones pueden tener mayor relevancia.

El *topic* con mayor peso para uno de los documentos de corrupción de menores fue el *topic* 4, que habíamos delimitado bajo el tema de pornografía infantil. El documento al que mayor peso se le ha asignado para el *topic* 5 es sobre trata de menores, al igual que habíamos concluido previamente. Avanzando con el análisis, y como bien habíamos remarcado, el *topic* 7 es un *topic* propio de trata, con mayor peso en el documento 2 sobre trata de menores. En cuanto al *topic* 9, en principio podríamos plantearnos que está más enfocado a delitos de corrupción de menores, pero también cabe recordar que una de las palabras con mayor frecuencia en los delitos de trata era “acusado”, por lo que su mayor peso se asigne a un documento sobre tratas podría encajar. Finalmente, el último de los *topics* encuentra su mayor peso en una sentencia sobre corrupción de menores, afirmando el enfoque planteando en nuestro análisis.

CONCLUSIONES

La realidad muchas veces se aleja de los datos y en cuestión del delito de trata de menores es un hecho que preocupa. Asombra en distintos organismos que exista tal dificultad en recabar información acerca del número de casos de trata de menores en España y su falta de visibilidad dificulta la implementación de medidas destinadas a su erradicación. En concreto, en el sistema judicial penal español, el número de sentencias dictadas por este tipo de delitos es insignificante si se compara con otros delitos similares como son el de prostitución o corrupción de menores. Como muchos abogan, gracias al desarrollo de nuevas tecnologías, este tipo de cuestiones pueden tratarse y analizarse, con el fin de entender cuál es el foco real del problema.

En primer lugar, tras la elaboración de este trabajo, se puede concluir que el mundo avanza rápido y que el uso de las herramientas de Inteligencia Artificial aún más. Concretamente, las técnicas de *Machine Learning* se están utilizado en una gran variedad de ramas de estudio, en las que ya se han logrado grandes avances. El mundo jurídico no se está quedando atrás y, aunque en un inicio tuvo un arranque más lento, ha empezado a incorporar este tipo de herramientas para le mejora de la eficiencia, eficacia y precisión de los procesos y decisiones jurídicas. Como se ha expuesto, las técnicas de *Machine Learning* ya no solo han servido para resumir textos, sino que ahora se implementan en la gestión de los tribunales, en la creación de motores de búsqueda o en la predicción de sentencias, planteando una próxima incorporación de jueces robot en los procesos.

En concreto, las técnicas de *Text Mining*, enfocadas al análisis de datos no estructurados como son los textos jurídicos y las sentencias particularmente, han ido evolucionando y cada vez son más numerosas las investigaciones que se centran en desarrollar nuevos enfoques que permitan seguir descifrando los datos ocultos en este tipo de documentos. En los últimos años, ha proliferado concretamente el uso de las técnicas de *Topic Modeling,* enfocadas hacia la identificación de *topics* dentro de un conjunto de documentos. En esta área queda aún mucho recorrido, sobre todo en su implementación en programas como R Studio; sin embargo, se ha podido constatar que la implementación del modelo LDA ha obtenido igualmente buenos resultados. En este sentido, una novedosa línea de investigación podría estar enfocada a desarrollar nuevos paquetes en

este entorno o simplemente crear guías de ayuda para el usuario en los procesos de implementación de estas herramientas.

Por último, y centrándonos concretamente en el objetivo final de este trabajo, a través del análisis práctico expuesto hemos podido extraer varias conclusiones acerca de las causas que puede haber detrás del recudido número de sentencias sobre trata de menores. Si bien es cierto que el análisis no ha dado un solución exacta y directa al problema, sí que ha permitido distinguir ciertas líneas argumentales de las que se podría realizar análisis más exhaustivos en el futuro:

- Como se ha comentado en la primera parte del trabajo, el delito de trata de menores tradicionalmente se penaba como de prostitución o corrupción, por lo que puede existir cierta reticencia aún por parte de los jueces

- Los delitos de prostitución y corrupción tienen marcos similares al de trata, pero la máxima pena aumenta cuando involucran a menores de 16. Se ha podido comprobar realizando un análisis de dichos textos y puede ser una de las razones por las que se prefiere penar por este tipo de delitos

- En las sentencias de trata de menores aparece con alta frecuencia el término "testigo", no siendo el caso del resto. Esto puede ser un indicativo de que -cuando no exista esta figura- se entendería que no existe como tal un proceso preconcebido de captura y traslado de la víctima y se pena por otro tipo de delitos

- Tras el análisis de similitudes, se ha podido observar que existen ciertas sentencias de tratas que se asemejan mas a los delitos de prostitución y corrupción porque no ponen tanto el foco en la primera parte del delito: la captura, transporte o traslado de los menores

- Finalmente, bajo mi punto de vista la conclusión más relevante es que, si bien el delito de trata puede cometerse con distintas finalidades (ya expuestas a lo largo del trabajo), tras el análisis se ha observado que prevalece con gran diferencia la finalidad de la explotación sexual. Esto puede indicar que no se están

identificando el resto de los supuestos de trata o que no se están instruyendo causas penales por ello.

Para finalizar este trabajo, cabe remarcar que hay ciertas limitaciones del trabajo que podrían solucionarse con análisis más profundos. Por un lado, la identificación de las sentencias de trata de menores en las bases de datos jurisprudenciales no ha sido tarea sencilla y si bien los motores de búsqueda son muy eficaces, hay otras herramientas que podrían servir de gran ayuda (por ejemplo, técnicas de *web scraping* para localizar palabras concretas dentro de las bases de datos). Por otro lado, uno de los grandes problemas expuestos es que no llegan a los tribunales españoles casos de trata de menores cuya finalidad no sea la explotación sexual lo que dificulta poder tener una imagen completa del problema. De esta forma, un análisis complementario podría centrarse en la búsqueda de sentencias de Tribunales Internacionales o en el Derecho Comparado, incluyendo en la base de datos objeto de análisis jurisprudencia de otros países.

BIBLIOGRAFÍA

Adyatama, A., Nathan, J., (2020). Topic Modeling with Latent Dririchlet Allocation (LDA). *Algoritma Technical Blog.* Obtenida el 01/06/2022 de https://algotech.netlify.app/blog/topic-modeling-lda/

Anand, D., Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University – Computer and Information Sciences*, 34(5), 2141-2150

Armonas, B., Buck, P, Miana, V. (2017). Challenges when suing Jurimetrics in Brazil – A survey of courts. *Future Internet*, 9(4), 1-68

Ash, E., Chent, D., Galletta, S. (2021). Measuring Judicial Sentiment: Methods and Application to US Circuit Courts. *Economica.*

Benoit, K., Obeng, A., Watanabe, K., Matsuo, A., Nulty, P., Muller, S., Benoit, K. (2014). readtext: Important and Handling for Plain and Formatted Text files. *Github.* Obtenido el 10/06/2022 de https://github.com/quanteda/readtext/issues

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30), 774

Bickel, M. (2019). Reflecting trends in the academic landscape of sustainable energy using probabilistic topic modeling. *Energy Sustain Doc*, 49(9)

Bjelland, H. F. (2017). Identifying human trafficking in Norway: a register-based study of cases, outcomes, and police practices. *European Journal of Criminology*, 14 (5), pp. 522-542

Blei, D. Topic Modeling and Digital Humanities (2012). *Journal of Digital Humanities*. Obtenida el 28/04/2022 de http://journalofdigitalhumanities.org/2-1/topic-modeling-and-digital-humanities-by-david-m-blei/

Blei, D., y Lafferty, J (2009). Topic Models. En A.Srivastava y M.Sahami (ed.), Text Mining. *Classification, Clusstering and Applications* (pp.1-24), Champman and Hall/CRC

Borges, R. (2020). El sesgo de la máquina en la toma de decisiones en el proceso penal. *Ius et Scientia*, 6(2), 54-71

Carter, D., Brown, J., Rahmani, A. (2016). Reading the High court at a distance: topic modelling the legal subject matter and judicial activity of the high court of Australia. *UNSW Law Journal*, 39(4)

Castaño, M., Barrio, C., Diez, I., Maffeis, G., Olaguibel, A. (2022). Qué sabemos y como lo contamos. Cultura de datos en la trata de seres humanos. *Instituto Universitario de estudios sobre Migraciones de la Universidad Pontificia Comillas.* Obtenida el 98/05/2022 de https://www.unicef.es/sites/unicef.es/files/comunicacion/que_sabemos_como_lo_contamos.pdf

Código Penal [CP]. Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal. Artículos 177,183,188, 189 (España)

Colom, E. (2021, 16 de septiembre). Redada policial contra la prostitución de menores en Mallorca con 17 detenidos. *El Mundo*. Obtenida el 16/05/2022 de https://www.elmundo.es/baleares/2021/09/16/61433d99e4d4d89f5a8b45fc.html

Corvalán, J. G., (2018). Inteligencia artificial: retos, desafíos y oportunidades – Prometea: la primera inteligencia artificial de Latinoamérica al servicio de la Justicia. *Revista de Investigações Constitucionais*, 5(1), 295-316

Del Moral, A. (2020). La trata de seres humanos en el ordenamiento jurídico español. *Lefebvre*. Obtenida el 24/04/2022 de https://elderecho.com/la-trata-de-seres-humanos-en-el-ordenamiento-juridico-espanol

Eguíluz, J. A. (2020). Desafíos y retos que plantean las decisiones automatizadas y los perfilados para los derechos fundamentales, *Estudios de Deusto*, 68(2), 325-367

Europa Press (2020, 14 de julio). Tres detenidos en Málaga acusados de abuso sexual a tres chicas fugadas de centros de menores. *Europa Press.* Obtenida el 16/05/2022 de https://www.europapress.es/andalucia/malaga-00356/noticia-tres-detenidos-malaga-presunto-caso-abuso-sexual-tres-menores-fugado-20200714112017.html

Feldman, R., & Sanger, J. (2006). Introduction to Text Mining. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data* (pp. 1-13). New York: Cambridge University Press

Genesereth, m. (2015). Computational Law: The Cop in the Bakseat. *Code X: The Stanford Center for Legal Informatics*. Obtenida el 25/04/2022 de http://logic.stanford.edu/publications/genesereth/complaw.pdf

Grün, B., Hornik, K. (2011). topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software*, 40(13), 1–30.

Harry, S. (2014). Machine Learning and Law. *Washington Law Review*, 89(1), 1-29

Joyanes, L. (2013). *Big Data. Análisis de grandes volúmenes de datos en organizaciones*. México D.F: Alfaomega Grupo Editor

Katz, D., Bommarito II, M., Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. *Plos one.* 12(4)

Kherva, P., y Bansal, P. (2019). Topic Modeling: a comprehensive review. *EAI Endorsed Transactions on Scalable Information Systems*, 7 (24), 1-16

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., Mullainathan, S. (2017). Human Decisions and Machine Predictions. *National Bureau of Economic Research*

Laitonjam, N., Padmansabhan, V., Pujari, A., Prasad, R. (2015). Topic Modeling for songs. *International Conference on Information Technology*, 130-135

Lampach, N., Dyevre, A. (2018). Issue attention on international courts: A text-mining approach. *SSRN Electronic Journal*

Liddy, E. (2001). Natural Language Processing. Nueva York: Marcel Decker, Inc.

Livermore, A., Riddell, A., Rockmore, D. (2016). A Topic Model Approach to Studyin Agenda Formtion for the U.S. Supreme Court. *SSRN Electronic Journal*

Lo, R. T.-W., He, B., Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*

Loevinger, L. (1971). Jurimetrics the next step forward. *Jurimetrics Journal*, 12 (1), 3-41

Mayer-Schönberger, V., Cukier, K. (2013). *Big Data: A revolution that will transform how we live, work and think.* John Murray Publishers Ltd

McCallm, A., Corrada, A., Wang, X. (2005). Topic and Role Discovery in Social Networks. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*

Molina, Luis Carlos. (2002). Torturando los Datos Hasta que Confiesen. Obtenida de https://www.businessintelligence.info/resources/assets/dss/molina-torturando-datos.pdf

Moro, S., Pires, G., Rita, P., Cortez, P. (2019). A text mining and topic modelling perspective of ethnic marketing research. *Journal of Business Research*, 103, 275-285

Navas, S. (2017). Derecho e Inteligencia Artificial desde el diseño. Aproximaciones. En: S. Navarro (coord.), *Inteligencia artificial Tecnología Derecho* (pp. 23-72). Madrid: Tirant lo Blanch.

Nay, J. (2018). Natural Language Processing and Machine Learning for Law and Policy Texts. *Legal Informatics. Cambridge University Press*

Rodrigo, A. (2016). Automatic Coherence Evaluation Applied to Topic Models. *U. Porto. Faculdade de ciencias universidade do Porto*

Rosen, M., Griffitths, T., Steyvers, M., Smyth, P. (2004). The Author-Topic Model for Authors and Documents. *Uia'04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 487-494

Russell, A. (2018). Human Trafficking: A Research Synthesis on Human-Trafficking Literature in Academic Journals from 2000–2014. *Journal of Human Trafficking*, 4(2), 114–136

Selivanoz, D., Bickel, M., Wang, Q. (2022). Text2vec: Modern Text Mining Framework for R. Obtenida el 10/06/2022 de http://text2vec.org/

Sila, M., Bicalho, G., De Paula, T., y Araujo, H (2018). Sentiment Classification over Brazilian Supreme Court decisions using Multi-Channel CNN. Obtenida el 25/04/2022 de https://vixra.org/pdf/1910.0568v1.pdf

Silge, J., Robinson, D. (2016). tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *JOSS*, 1(3)

Straka, M., Stakova, J. (2022). Udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the 'UDPipe' 'NLP' Toolkit. *ACL Technology*. Obtenida el 09/06/2022 de https://aclanthology.org/K17-3009.pdf

Tidey, C. (2018). Los niños representan casi una tercera parte de las víctimas identificadas de la trata en todo el mundo. Obtenida el 08/05/2022 de https://www.unicef.org/es/comunicados-prensa/los-ni%C3%B1os-representan-casi-una-tercera-parte-de-las-v%C3%ADctimas-de-la-trata

Vijayarani, S., Ilamathi, J. (2015). Preprocessing Techniques for Text Mining – An Overview. *International Journal of Computer Science & Communication Networks,* 5(1), 7-16

Viola, L (2018). Guistizia predittiva. *Treccani.* Obtenida el 25/04/2022 de https://www.treccani.it/enciclopedia/giustizia-predittiva_%28Diritto-on-line%29/

Wallach, H. (2006). Topic Modeling: beyond bag-of-words. *ICML '06: Proceedings of the 23rd international conference on Machine learning,* 977-984

Wickham, H. (2016*).* ggplot2: Elegant Graphics for Data Analysis. *Springer-Verlag New York*. Obtenida el 09/06/2022 de https://ggplot2.tidyverse.org

ANEXOS

#Código – aplicación práctica

#Librerías

```
library(ggplot2)
library(quanteda)
library(readtext)
library(topicmodels)
library(quanteda.textplots)
library(quanteda.textstats)
library(tidytext)
library(dplyr)
library(utf8)
library(udpipe)
library(text2vec)
```

#Importación de datos

```
pdf_path1<- "C:/Users/OneDrive/Escritorio/TFGS/TFG ADE/sentencias/Trata de menores"
pdf_path2<-"C:/Users/OneDrive/Escritorio/TFGS/TFG ADE/sentencias/Corrupción y prostitución"
trata_de_menores<- list.files(path = pdf_path1, pattern = 'pdf$',  full.names = TRUE)
prost_corrup<-list.files(path = pdf_path2, pattern = 'pdf$',  full.names = TRUE)
trata_de_menores<- readtext(trata_de_menores)
prost_corrup<-readtext(prost_corrup)
```

#Creación del corpus

```
c_trata_de_menores<-corpus(trata_de_menores)
c_prost_corrup<-corpus(prost_corrup)
```

#Proceso de *tokenización* y creación de la matriz TDF

```
tokens1<-tokens(c_trata_de_menores,remove_numbers = TRUE, remove_punct = TRUE)
midfm1<-dfm(tokens1)
```

```
dim(midfm1)
tokens2<-tokens(c_prost_corrup,remove_numbers = TRUE, remove_punct = TRUE )
midfm2<-dfm(tokens2)
dim(midfm2)
topfeatures(midfm2)
```

#Limpieza del texto

```
midfm1<-dfm_select(midfm1, pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-zA-Z]\\b"), selection = "remove", valuetype = "regex")
midfm2<-dfm_select(midfm2, pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-zA-Z]\\b"), selection = "remove", valuetype = "regex")
midfm1<-dfm_remove(midfm1, pattern = stopwords(language="spa"))
midfm2<-dfm_remove(midfm2, pattern = stopwords(language="spa"))
midfm1<-dfm_remove(midfm1, pattern = c("año", "artículo","delito","sentencia", "num", "art", "tribunal","delito"))
midfm2<-dfm_remove(midfm2, pattern = c("año", "artículo","delito","sentencia", "num", "art", "tribunal","delito"))
midfm1<- dfm_wordstem(midfm1,language = "spa")
midfm2<- dfm_wordstem(midfm2,language = "spa")
```

#Frecuencias

```
features_freq1<-textstat_frequency(midfm1, n = 20)
ggplot(features_freq1, aes(x = reorder(feature,-frequency), y = frequency)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  labs(x="Tokens", y="Frecuencia", title="Frecuencias - Trata de menores")

features_freq2<-textstat_frequency(midfm2, n = 20)
ggplot(features_freq2, aes(x = reorder(feature,-frequency), y = frequency)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  labs(x="Tokens", y="Frecuencia", title="Frecuencias - Prostitución y corrupción")
```

#N-Gramas: Brigramas

```
bigrams1<-quanteda::tokens(c_trata_de_menores,remove_punct = TRUE,
remove_numbers = TRUE)%>%
  tokens_select(pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-zA-Z]\\b"),
selection = "remove", valuetype = "regex") %>%
  tokens_remove(stopwords("spanish"))%>%
  tokens_ngrams(n=2) %>%
  dfm()
textplot_wordcloud(bigrams1, min_count=10, random_order = FALSE,
                   rotation = .25,
                   min_size = 1,
                   color = RColorBrewer::brewer.pal(3,"Dark2"))

bigrams2<-quanteda::tokens(c_prost_corrup,remove_punct = TRUE, remove_numbers
= TRUE)%>%
  tokens_select(pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-zA-Z]\\b"),
selection = "remove", valuetype = "regex") %>%
  tokens_remove(stopwords("spanish"))%>%
  tokens_ngrams(n=2) %>%
  dfm()
textplot_wordcloud(bigrams2, min_count=20, random_order = FALSE,
                   rotation = .25,
                   min_size = 1,
                   color = RColorBrewer::brewer.pal(3,"Accent"))
```

#Lematización

```
udmodel <- udpipe_download_model(language = "spanish")
udmodel<-udpipe_load_model(file = udmodel$file_model)
udmodel<- udpipe_load_model(file = "spanish-gsd-ud-2.5-191206.udpipe")   # para
obtenerlo hemos consultado list.files(getwd())

texto1<-as_utf8(c_trata_de_menores)
anot_texto1<-udpipe_annotate(udmodel,texto1,  tagger = "default", parser = "none")
anot_texto1<- as.data.frame(anot_texto1, detailed = TRUE)
```

```
anot_texto1 %>%
  filter(upos %in% c('NOUN','VERB','ADJ')) %>%
  count(upos, lemma) %>%
  group_by(upos) %>%
  slice_max(order_by = n, n =10) %>%
  ggplot()+
  geom_col(aes(x=reorder_within(lemma,n,upos),y=n,fill=lemma))+
  scale_x_reordered()+
  labs(x="Lemma", y="Frecuencia", title="Lematización - Trata de menores")+
  facet_wrap(vars(upos), scales = "free", ncol = 2)+
  coord_flip()+
  theme(legend.position = "none", text=element_text(size=18))

texto2<-as_utf8(c_prost_corrup)
anot_texto2<-udpipe_annotate(udmodel, texto2,  tagger = "default", parser = "none")
anot_texto2<- as.data.frame(anot_texto2, detailed = TRUE)
anot_texto2 %>%
  filter(upos %in% c('NOUN','VERB','ADJ')) %>%
  count(upos, lemma) %>%
  group_by(upos) %>%
  slice_max(order_by = n, n =10) %>%
  ggplot()+
  geom_col(aes(x=reorder_within(lemma,n,upos),y=n,fill=lemma))+
  scale_x_reordered()+
  labs(x="Lemma", y="Frecuencia", title="Lematización - Prostitución y corrupción")+
  facet_wrap(vars(upos), scales = "free", ncol = 2)+
  coord_flip()+
  theme(legend.position = "none", text=element_text(size=18))
```

#Pesos TF-IDF y similitud de documentos

```
c_conjunto<-c_trata_de_menores+c_prost_corrup
tokens3<-tokens(c_conjunto, remove_numbers = TRUE, remove_punct = TRUE)
midfm3<-dfm_select(midfm3, pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-zA-Z]\\b"), selection = "remove", valuetype = "regex")
```

```
midfm3<-dfm_remove(midfm3, pattern = stopwords(language="spa"))
midfm3<-dfm_remove(midfm3, pattern = c("año", "artículo","delito","sentencia",
"num", "art", "tribunal","delito"))
midfm3<- dfm_wordstem(midfm3,language = "spa")

mitfidf3<-dfm_tfidf(midfm3, scheme_tf="prop", base=2)
similitud<- textstat_simil(mitfidf3, margin="documents", method="cosine")
cluster<-hclust(as.dist(similitud))
cluster$labels<-docnames(midfm3)
fviz_dend(x = cluster, cex = 0.8, lwd = 0.8, k = 3,
        k_colors = c("green4", "orange2","purple3"),
        rect = TRUE,
        rect_border = "gray",
        main="Cosine Distance on Normalized Token Frequency",
        rect_fill = FALSE)
```

#Topic modeling: preparación de los datos

```
c_conjunto<-corpus_reshape(c_conjunto, to="paragraphs")
tokens4<-tokens(c_conjunto,remove_numbers = TRUE, remove_punct = TRUE )
midfm4<-dfm(tokens4)
midfm4<-dfm_select(midfm4, pattern = c("[0-9]+(?:st| st|nd| nd|rd| rd|th| th|s)","\\b[a-
zA-Z]\\b"), selection = "remove", valuetype = "regex")
midfm4<-dfm_remove(midfm4, pattern = stopwords(language="spa"))
midfm4<-dfm_remove(midfm4, pattern = c("año", "artículo","delito","sentencia",
"num", "art", "tribunal"))
midfm4 <- dfm_trim(midfm4, min_docfreq = 20)
dtm=convert(midfm4, to="topicmodels")
```

#Selección del número de *k topics*

```
set.seed(1)
tcm=crossprod(sign(as.matrix(dtm)))
logger = lgr::get_logger('text2vec')
logger$set_threshold('debug')
res1=data.frame()
```

```
set.seed(1)
range<-seq(2, 12,by=2)
for(i in range){
  topicModel     <- LDA(dtm, method="Gibbs", k=i, control=list(alpha=0.1))
  topicTerms<-terms(topicModel,k=10)
  topicTerms_matrix<-as.matrix(topicTerms)
  res1[i/2,1]=i
  res1[i/2,2]=mean(coherence(topicTerms_matrix, tcm, metrics =
c("mean_pmi"),n_doc_tcm = nrow(tcm)))
}

res2=data.frame()
set.seed(1)
range<-seq(2, 12,by=2)
for(i in range){
  topicModel     <- LDA(dtm, method="Gibbs", k=i, control=list(alpha=0.1))
  topicTerms<-terms(topicModel,k=10)
  topicTerms_matrix<-as.matrix(topicTerms)
  res2[i/2,1]=i
  res2[i/2,2]=mean(coherence(topicTerms_matrix, tcm, metrics =
c("mean_npmi_cosim"),n_doc_tcm = nrow(tcm)))
}

ggplot() +
  geom_line(data= res1, aes(x = V1, y = V2), color = 'red', stat="identity", size=1,
label="mean")+
  geom_line(data= res2, aes(x = V1, y = V2), color = 'purple', stat="identity", size=1)+
  labs(x="Número de K", y="Coherencia", title="Cálculo de coherencias")
```

#Modelo y análisis de resultados

```
topicModel<- LDA(dtm, method="Gibbs", k=10, control=list(alpha=0.1))
c_topics<- tidy(topicModel, matrix = "beta")
c_topics_term <- c_topics %>%
  group_by(topic) %>%
```

```
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)

c_topics_term %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()

corp_doc <- tidy(topicModel, matrix = "gamma")
corp_doc_filter<-corp_doc_filter%>%group_by(topic)%>% slice(which.max(gamma))
ggplot()+
  geom_bar(data=corp_doc_filter2, aes(x = topic, y = reorder(document,topic)), stat
="identity", fill=corp_doc_filter$topic)
```