Facultad de Ciencias Económicas y Empresariales

Text Mining en el Derecho: Topic Modeling y Sentiment Analysis

Clave: 201602404

MADRID | Abril 2021

Resumen

En el presente trabajo se pretende probar la aplicabilidad de técnicas de *Text Mining* tales como el *Sentiment Analysis* y el *Topic Modeling* (este último mediante el algoritmo LDA) descritas en la literatura anterior a este trabajo. Se aportará una perspectiva de cómo estas técnicas pueden hacer más eficiente la búsqueda de jurisprudencia en bases de datos online, y demostrando la posible influencia psicológica en el proceso interpretativo de la ley que realizan los jueces, la cual puede suponer una amenaza a los derechos de las partes en un proceso judicial.

Palabras y conceptos clave: *Text Mining*, *Sentiment Analysis*, *Topic Modeling*, LDA, sentencias, sentimientos y Derecho.

Abstract

The present project's aim is to prove the applicability of Text Mining techniques such as Sentiment Analysis and Topic Modeling (that one through LDA algorithm) which are described in papers which are already published. This project will use a perspective in which it will be explained how those techniques can make more efficient the process of searching for legal judgements in online databases, and it will also be proved how psychological influence may affect the interpretative process that each judge has to do when interpreting the law, an influence that could be a threat to some rights than each individual has during a judicial process.

Key words and concepts: Text Mining, Sentiment Analysis, Topic Modeling, LDA, legal judgements, sentiments and Law.

Índice

Índice de figuras

Listado de abreviaturas utilizadas

AAN	Auto de la Audiencia Nacional
AB - AG SEX	Abuso – Agresión Sexual
ATC	*Aggregate Topic Classifier*
BTC	*Binary Topic Classification*
BOW	*Bag of Words*
HP - CS	Hipoteca – Cláusula Suelo
LDA	*Latent Dirichlet Allocation*
LSA	*Latent Semantic Analysis*
NLP	*Natural Language Processing*
SAN	Sentencia de la Audiencia Nacional
SAP BA	Sentencia de la Audiencia Provincial de Badajoz
SAP C	Sentencia de la Audiencia Provincial de A Coruña
SAP O	Sentencia de la Audiencia Provincial de Oviedo
SAP OU	Sentencia de la Audiencia Provincial de Ourense
SAP SG	Sentencia de la Audiencia Provincial de Segovia
SAP TF	Sentencia de la Audiencia Provincial de Santa Cruz de Tenerife
STS	Sentencia del Tribunal Supremo
SVM	*Support Vector Machines*
TDM	*Term-Document Matrix*
TF-IDF	*Term Frequency-Inverse Document Frequency*
URI	*Uniform Resource Identifier*
VEM	*Varational Expectation-Maximization*

1. INTRODUCCIÓN

El *Big Data* y sus técnicas, entre las que se encuentra, por ejemplo, el *Machine Learning*, son técnicas de tratamiento masivo de datos que, dadas las posibilidades que ofrecen, se presentan como fundamentales para el devenir tanto futuro como presente de casi cualquier sector empresarial. Sin embargo, y aunque parezca menos relacionado, estas técnicas presentan un alto potencial en lo relativo a su implementación en el ámbito del Derecho. No son pocos los despachos de abogados que, o bien subcontratan una empresa especializada en *Big Data* para obtener algún tipo de ventaja, o bien cuentan con su propio departamento para obtener algún fin concreto. Con técnicas de colección de datos y creación de bases propias de jurisprudencia, o de *Text Mining* para la rápida extracción de información de sentencias, la entrada de estas técnicas en el mundo del derecho es algo innegable.

Uno de los principios del derecho en cualquier proceso judicial es la imparcialidad de los jueces. Esta imparcialidad, derivada del derecho de toda persona a tener un proceso justo, suele presentarse como un método excluyente de jueces cuando se demuestra que tienen algún tipo de relación construida con alguna de las partes, sea positiva o negativa, pero no en el ámbito de la interpretación de la ley. Los jueces son personas al fin y al cabo, y todo ser humano tiene prejuicios o influencias psicológicas o del subconsciente que hacen que su neutralidad pueda verse afectada de forma irracional, sin poder ser controlada. Herramientas como el *Sentiment Analysis*, según aduce la literatura, pueden ayudar a analizar si los jueces son verdaderamente imparciales, o si ese subconsciente puede poner en jaque a la misma, y poder adoptar así soluciones pertinentes al respecto.

Por otro lado, no puede negarse que hay una innumerable cantidad de bases de datos jurisprudenciales, muchas de ellas en línea, hoy en día. Los buscadores de estas bases de datos son extremadamente eficaces cuando la persona que las consulta busca una sentencia en concreto, con sus datos exclusivos y pertinentes, tales como el número de procedimiento. Sin embargo, dadas las múltiples ramas que tiene el Derecho, la búsqueda por palabras, por ejemplo, es a veces algo complicado, y la búsqueda por temas, prácticamente inexistente. La aplicación de técnicas de *Text Mining* tales como el *Topic Modeling* pueden ayudar a realizar una clasificación más exhaustiva de la base

jurisprudencial, tal y como dicen algunos autores, facilitando así al usuario que las consulta la búsqueda de sentencias por temática y su experiencia en la misma.

El objetivo de este trabajo, por tanto, es comprobar si la teoría escrita en lo relativo a implementación de algoritmos de *Text Mining* y *Data Mining*, más concretamente de *Sentiment Analysis* y el LDA, en el mundo del Derecho está en lo cierto y, en caso afirmativo, explicar cómo podría el algoritmo LDA mejorar los motores de búsqueda utilizados en las bases de búsqueda de jurisprudencia.

La metodología utilizada para el desarrollo del trabajo comenzó con lectura de literatura relativa al tema, más concretamente de los algoritmos, de su aplicación, y de los problemas presentados en párrafos anteriores. Posteriormente, se seleccionó una compilación de sentencias a modo de *data set* para la parte práctica de este trabajo, para concluir con el desarrollo de los algoritmos utilizados y la elaboración de conclusiones finales. Las fuentes de obtención de la literatura estudiada para este trabajo proceden, en su mayoría, de bases de datos como Google Scholar o Scorpus, e incluyen libros, manuales de derecho, revistas científicas y publicaciones sobre la implementación de las técnicas de *Text Mining* en el Derecho. Por otro lado, se estudió de la legislación vigente en materia de protección de datos, así como los paquetes de R objeto de aplicación en el desarrollo del algoritmo del presente trabajo.

Para lograr lo expuesto anteriormente, es necesaria una estructura que siga el método deductivo, es decir, empezar por el estudio de lo más general para concluir con lo más específico. Para ello, en el capítulo 1, se estudiarán conceptos básicos y generales relativos al *Big Data*, *Business Analytics* y *Text Mining*, entre otros, y se analizará cómo concurren con el Derecho, así como los problemas en este último sobre los que pueden actuar las técnicas utilizadas en el presente trabajo. En el capítulo 2 se explicarán, de forma teórica, los algoritmos y técnicas sobre las que se basa este trabajo, así como conceptos clave relativos al *text Mining* y las técnicas de preprocesamiento de textos. En el capítulo 3, se implementarán de forma práctica en RStudio los algoritmos y se analizarán los resultados, para finalizar el presente trabajo en el capítulo 4 con la redacción de conclusiones finales en base a los resultados obtenidos en el capítulo anterior.

2. EL TEXT MINING Y SU APLICABILIDAD AL MUNDO DEL DERECHO

Dadas las cuestiones planteadas en la introducción, y antes de entrar en un análisis más profundo de las mismas, es necesario explicar una serie de conceptos inherentes al *Tex Mining* y *Business Analytics* que sirvan como base para poder comprender los algoritmos objeto de estudio presentados en este trabajo, así como la relación de estos conceptos con el mundo del Derecho, relación que, a simple vista, puede parecer poco intuitiva. Tras estas nociones, se estudiará la problemática planteada en una mayor profundidad.

2.1. Big Data, Business Analytics y otros conceptos relacionados con el Text Mining

Es innegable afirmar el impacto que tienen las nuevas tecnologías en el día a día de cualquier empresa y, en general, en el día a día de la sociedad. Tal y como adelantaba en la introducción, el negocio de los de datos masivos es algo que ya forma parte del día a día de muchas empresas, pero es importante considerar que el mero de hecho de tener una gran cantidad de datos no sirve de nada si no se les da una utilidad clara, ya que la aplicación pura de los datos es imposible. Es aquí donde entra en juego el *Business Analytics* y todo aquello aunado bajo su concepto. *El Business Analytics* puede ser definido como el uso de datos para la toma de decisiones para negocios (Shmueli et al. 2018), es decir, *Business Analytics* es el proceso mediante el cual se intenta extraer todo el valor útil posible de un conjunto de datos con el fin de aplicarlo a la toma de decisiones empresariales y extraer un beneficio o una ventaja exclusiva para con la competencia, o lo que es lo mismo, mejorar el rendimiento de la empresa mediante toma de decisiones basadas en el análisis de datos. Por tanto, cabe afirmar que el *Business Analytics* y el *Big Data*, concepto que explicaré a continuación, van siempre y en todo momento de la mano.

El *Big Data*, puramente definido, puede ser entendido como "toda la información que no puede ser procesada o analizada utilizando herramientas o procesos tradicionales" (Puyol, 2014, p. 471), es decir, datos a gran escala almacenados en grandes colecciones. Estos datos, informatizados todos, se guardan en grandes bases de datos, y suelen estar caracterizados por cuatro características que marcan la utilidad marginal de los mismos.

Estas características son típicamente conocidas como las cuatro V's, las cuales según Shmueli (2018) son las siguientes:

a.- Volumen: Hace referencia a la cantidad de datos que están comprendidos en la base.

b.- Variedad: Hace referencia a los tipos diferentes de datos comprendidos en la colección.

c.- Velocidad: Hace referencia al tiempo en que estos datos pueden generarse y cambiar.

d.- Veracidad: Hace referencia al problema que puede darse en cuanto a la fiabilidad de los datos por el modo en que estos se crean. Estos datos suelen generarse mediante procesos en los que los controles de calidad, en muchos casos, no se dan, por lo que la veracidad de estos puede verse en jaque.

Por otro lado, además de dar nombre a estas grandes cantidades de datos, el *Big Data* también recoge todas las distintas técnicas destinadas al tratamiento de datos masivos en general. A continuación, explicaré brevemente conceptos básicos en lo relativo al tratamiento de datos que deben de presentarse antes de analizar en profundidad las técnicas objeto de estudio en este trabajo.

2.1.1. Machine Learning

El *Machine Learning* puede ser entendido como el estudio y aplicación de algoritmos computacionales a máquinas que funcionan mediante inteligencia artificial buscando, como fin último, la evolución de estas máquinas. Esta evolución debe estar enfocada en el autoaprendizaje de las mismas mediante el estudio del ambiente que las rodea (El Naqa y Murphy, 2015). Dicho aprendizaje puede ser supervisado y no supervisado. El aprendizaje supervisado puede ser entendido como aquel que aprendizaje en el cual la función desarrollada se deduce de los datos objetos de estudio (datos de entrenamiento), mientras que el no supervisado sigue un modelo distinto, en el cual el modelo desarrollado se adapta y ajusta a las observaciones objeto de estudio.

Para poder alcanzar el fin anterior, las técnicas empleadas se corresponden habitualmente con distintos lenguajes. En este trabajo me centraré en el estudio de las

técnicas relativas al estudio de textos, muchas de las cuáles pertenecen al campo del NLP (*Natural Language Processing*), por lo que creo necesario realizar una inferencia sobre este concepto.

El NLP es uno de los conjuntos de técnicas de análisis de textos más usados en el *Text Mining.* Relacionado con la inteligencia artificial (como cualquier técnica asociada al *Machine Learning*), tal y como describe Liddy (2001), el NLP es un "conjunto de técnicas computacionales que analizan y representan de forma natural lo que ocurre en los textos en distintos niveles del lenguaje" (p.1). El fin último es tratar de que, mediante el uso de este tipo de lenguaje, las máquinas que trabajan mediante inteligencia artificial, tal y como describe esta autora, alcancen un nivel de entendimiento y comprensión de textos similar al que realiza el ser humano. Aplicando estas técnicas tras la fase de preprocesamiento de textos, se adquiere un conocimiento mucho mayor del texto, lo que posibilita un mejor análisis y, potencialmente, un mejor resultado, dependa el mismo de la aplicación posterior de técnicas puras de *Data Mining* o no.

2.1.2. Data Mining

El *Data Mining*, o minería de datos, puede ser definido como el conjunto de métodos o técnicas utilizadas en el mundo de *Business Analytics* con el fin de "identificar datos válidos, potencialmente útiles, y sus correlaciones" (Chung y Gray, 1999, p.11). Poder interpretar las colecciones de datos masivos y obtener así un objetivo concreto, el cual puede ir desde una simple comprensión de los datos a la elaboración de un modelo que pueda predecir algo en base al comportamiento de estos a lo largo del tiempo, también es función del *Data Mining*.

Por todo esto, tal y como puede apreciarse, cabe afirmar que el *Data Mining* y el *Big Data* también son términos que van siempre de la mano. Esto es debido a que no hay *Data Mining* sin las colecciones de datos aportadas por el *Big Data*, y no hay utilidad del *Big Data* sin las técnicas de trabajo que comprende el *Data Mining*.

Dentro del concepto de *Data Mining* caben una gran cantidad de técnicas, las cuáles son objeto de clasificación y división. La división de técnicas más correcta es, probablemente, la que se puede realizar en base a los tipos de datos sobre los que pueden

tratar y trabajar, es decir, datos estructurados y no estructurados. Los datos estructurados son aquellos que, al almacenarse, lo hacen siguiendo una estructura (típicamente en tablas). Lenguajes de programación, tales como SQL, que buscan la identificación de datos relacionados entre sí, aplicarían sobre estos conjuntos de datos. Por el contrario, los datos no estructurados son aquellos que, al almacenarse, no lo hacen siguiendo una estructura concreta. Su ventaja frente a los datos estructurados es que su proceso de almacenamiento es mucho más rápido. Por último, destacar que existen datos con estructura híbrida (es decir, con parte estructurada y parte no estructurada) llamados datos semiestructurados.

2.1.3. Text Mining

El *Text Mining* es una de las múltiples variantes que existen en el *Data Mining*. En esta, en concreto, se busca el trabajo sobre textos. Hearst (2003) entiende que, al aplicar el proceso de *Text Mining*, se pretende obtener información nueva de los textos, que no está previamente escrita, a partir de la extracción y estudio de información que sí lo está, es decir, inferir información desconocida a partir de datos o información conocida. Esta misma autora, además, argumenta que fuera de lo anterior, la principal diferencia de estas técnicas con respecto a las generales de *Data Mining* es que, en las relativas a *Text Mining*, los patrones de los textos objeto de estudio se extraen mediante lenguaje natural, es decir, a partir de cuerpos de textos, mientras que los relativos al *Data Mining* lo hacen mediante bases de datos, sean estos estructurados o no, pero no de textos como tal.

Sin embargo, pese a las diferencias anteriores, cabe destacar que *Text Mining* y *Data Mining* no son conceptos incompatibles, sino que, de hecho, son complementarios. Explicaré en puntos posteriores las técnicas exclusivas de preprocesamiento de textos y conceptos que pertenecen exclusivamente al *Text Mining* en mayor profundidad, pero solo quiero destacar que, muchas veces, el objetivo al aplicar estas técnicas de *Text Mining* es el inferir información de textos con el fin de aplicar posteriormente técnicas de *Data Mining* para obtener un resultado concreto, tales como predicciones o agrupaciones vía *Clustering*, por ejemplo.

2.2. La relación entre Derecho y Text Mining

2.2.1. Contexto actual

El *Data Mining*, tal y como he dicho anteriormente, puede definirse genéricamente como el conjunto de técnicas destinadas al tratamiento masivo de datos y su futura interpretación. Con fines variados, no solo se antoja como el futuro en las empresas, sino que ya es presente. Este conjunto de técnicas puede, por ejemplo, dar una ventaja competitiva a una empresa en cuanto a clientes al posibilitar conocer mejor a los mismos. No en vano muchas veces se comenta que las empresas, especialmente las tecnológicas como puede ser Google, conocen muchas veces mejor que uno mismo los hábitos propios de compra de cualquier individuo.

Aunque parezcan a simple vista materias inconexas, la realidad es que el *Big Data* ya juega, a día de hoy, un papel importante en el Derecho. Este papel es destacable en dos vertientes fundamentales, por una parte, en lo relativo a la protección de datos y, por otra parte, en las aplicaciones prácticas de algoritmos de *Text Mining* sobre textos.

La obtención de datos, como he dicho antes, es un negocio, uno que tiene verdadero valor para las empresas por el posible uso de los mismos. Poder conocer de forma más concreta a la sociedad, los hábitos de consumo de las personas o similares, es una ventaja competitiva que puede ser una herramienta fundamental para poder encabezar un sector económico concreto a nivel empresarial, o para influenciar para obtener un objetivo concreto. Por ello, desde hace años hay empresas que destinan parte de su actividad a la compraventa de datos.

Para regular esto, que hasta hace unos años era considerado una laguna legal,[1] en el año 2016, la Unión Europea, más concretamente el Parlamento y Consejo Europeo, aprobaron el Reglamento 2016/679 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos, donde se regula tanto la obtención de los mismos, como la transmisión entre empresas de estos.

[1] No estoy diciendo aquí que hoy en día haya dejado de serlo. Todavía hoy muchas solicitudes que exigen la sobrerregulación de la protección de datos de, por ejemplo, la que realiza España, al entender que siguen existiendo vacíos legales y contradicciones con la legislación europea.

Este reglamento se traspuso en España en el año 2018 mediante la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de derechos digitales. Sin embargo, esta regulación, a día de hoy, sigue siendo insuficiente. Con casos como el de Cambridge Analytica, consultora que en el año 2018 adquirió datos de forma masiva e irregular provenientes de Facebook, se demuestra que en lo referido a la regulación de adquisición y transmisión de datos digitales queda aun un largo camino por recorrer. Asimismo, ese caso muestra lo potentes que pueden ser las aplicaciones de esos datos, ya que dicha consultora los usó para influenciar tanto en el referéndum en Inglaterra relativo al *Brexit*, como en las elecciones americanas donde Donald Trump consigue alzarse con la victoria. Al final, tal y como indica Ritcher (2015), los datos personales tecnológicos son parte de la identidad de la persona, la cual está ligada al concepto de dignidad humana, la cual es base del Derecho en cualquier país. La protección de datos, en palabras de dicho autor, debe ser reconocido como un derecho humano.

Por otro lado, ya se ha comentado antes que una de las ramas concretas del *Big Data* es el *Text Mining*, el cual hace referencia a las herramientas destinadas al procesamiento de textos y obtención de datos de los mismos. El uso práctico más habitual de este conjunto de técnicas puede encontrarse en las bases de datos de textos jurídicos.

Las bases de datos de textos jurídicos son compilaciones de documentos (sentencias, artículos, manuales, …) que tienen que ver con el Derecho. Todas ellas comparten una característica, y es que tienen un buscador al servicio del usuario que le posibilita localizar determinados textos en su base de datos. Estos buscadores funcionan mediante, habitualmente, el filtrado por palabras, aunque también es posible introducir otros filtros tales como el tipo de documento en concreto, o si son bases de datos centradas exclusivamente en jurisprudencia, es decir, sentencias o resoluciones provenientes de Tribunales, el Tribunal o Sala concreta de la que proviene dicha resolución.

Sin embargo, sobre todo en estos últimos, el proceso de búsqueda puede acarrear errores o dificultades para encontrar el documento deseado. Cuando la búsqueda se realiza en base a un documento concreto, donde se conoce el número de resolución o de procedimiento, los buscadores son eficaces, pero a la hora de encontrar un documento mediante palabras, la mezcla de filtros que pueden ser aplicados no es tan eficaz como para cuando se busca un documento en concreto. La implementación de algoritmos más

concretos, como puede ser el *Latent Dirichlet Allocation* (LDA), el cual se estudiará posteriormente, puede ayudar a la identificación más eficaz de palabras y facilitar la búsqueda temática del usuario. Sarioglu et al. (2016) argumentan en este sentido, y es que ellos, tras implementar LDA como modelo de extracción de *topics* sobre los cuáles, posteriormente, se aplicaron algoritmos clasificatorios en base a una colección de reportes clínicos, concluyeron que el uso de *Topic* Modeling, dadas sus características (las cuáles explicaré en puntos posteriores), genera procesos de clasificación más concretos y eficientes que con otras técnicas de análisis de textos previos al proceso de clasificación. Por último, es importante destacar que un proceso de clasificación eficaz es el paso previo para generar cualquier motor de búsqueda, tal y como explicaré posteriormente.

Fuera de las bases de datos, muchos despachos de abogados o departamentos jurídicos de empresas utilizan programas basados en algoritmos relativos al *Text Mining* con el fin de extraer datos específicos de documentos de jurisprudencia para acelerar los procesos. Esto es especialmente importante en países bajo un régimen o sistema de *Common Law*, donde la jurisprudencia es fuente del derecho, y saber los argumentos por los que un tribunal falla de una forma u otra implica saber de antemano cómo defender un caso concreto, aunque también se usa en los países bajo el régimen de *Civil Law*.[2] El implementar estas técnicas implica una reducción del tiempo estimado para analizar jurisprudencia, lo cual reduce también los costes en el corto/medio plazo pues, tal y como exponen Kumar y Raghuveer (2012), el LDA es una herramienta con demostrada gran eficacia en cuanto a lo referido al resumen de documentos de textos. Además, en esa misma publicación también prueban la viabilidad de implementarlo sobre textos jurídicos. Para ello, aplican LDA a un cuerpo de sentencias de diversos tribunales de India, y una vez los *topics* han sido obtenidos, comparan estos con los obtenidos por un cuerpo de abogados expertos que realizaron el mismo proceso de resumen y extracción de *topics* pero a mano. El resultado es que los *topics* obtenidos y, por tanto, el resumen

[2] El sistema de Common Law es el presente en países como Reino Unido o Estados Unidos. En él, la jurisprudencia es considerada como fuente de derecho, y por ello, las resoluciones de los Tribunales de Rango Superior son de obligatorio cumplimiento para los tribunales de rango inferior. Por otro lado, el sistema de Civil Law, el que tenemos en España, no cuenta con la jurisprudencia como fuente de derecho, sino que tal y como recoge el Código Civil Español en el artículo 1, "*Las fuentes del ordenamiento jurídico español son la ley, la costumbre y los principios generales del derecho*". Sin embargo, bajo mi punto de vista, recientemente se están produciendo hechos y actos que pueden llevar a pensar que los Tribunales de Justicia están creando jurisprudencia para "legislar", argumentando este hecho en la interpretación abierta y extensiva de la ley, algo contrario al sistema de Derecho que tenemos en España. De ser así, el papel de las técnicas de Big Data se potencia mucho más.

de las sentencias en ambos casos, eran muy similares, por lo que la precisión de resumen del algoritmo LDA era muy similar a la precisión que tenían los abogados expertos.

Asimismo, y tal y como desarrollaré a continuación, la ley no es un objeto de implementación automática en muchas ocasiones, sino que el modelo seguido requiere de una interpretación de la misma por parte del juez. Esta interpretación debe ser neutra, pero hay cuestiones psicológicas inherentes al ser humano que pueden distorsionar esta capacidad de interpretación. Técnicas como el *Sentiment Analysis* permitirían ver si esta contaminación se da y, en caso de darse, conocer la magnitud del problema para intentar ponerlo solución pues, como argumentan Gómez et al. (2016) tras implementar análisis de sentimiento en una compilación de textos jurídicos, aunque demuestran que los jueces están en cierto modo sesgados en cuanto a sentimientos se refieren, concluyen que es un punto donde se necesita mucha más investigación para poder afirmar esta tesis.

2.2.2. La influencia de los sentimientos en la toma de decisiones y su peligro para el principio de imparcialidad judicial. La conjunción con el Text Mining.

Determina el artículo 24.2 de la Constitución Española que todos tiene derecho a "un proceso público sin dilaciones indebidas y con todas las garantías". En primer lugar, este precepto legal hace referencia al proceso judicial en que un individuo puede estar involucrado y, en segundo lugar, queda a desarrollo posterior cuáles son esas garantías.

En sentencia 164/2008, de 15 de diciembre, el Tribunal Constitucional establece que, entre otras, "otra de las garantías fundamentales del derecho a un proceso justo es la del derecho al juez imparcial que, al propio tiempo, configura un derecho fundamental implícito en el derecho al juez legal" proclamado en dicho artículo. La Ley Orgánica 6/1985, de 1 de julio, del Poder Judicial, en los artículos 389 a 397 recoge una serie de incompatibilidades que, en caso de concurrir bajo la persona del juez, hacen que este sea incompatible con un caso concreto. Con esto se intenta que el juez pueda fallar de la forma más objetiva posible en cada caso concreto, asegurando el derecho a un juicio justo. A modo de ejemplo, cabe destacar que entre estas limitaciones se encuentra la imposibilidad de que sea juez en un proceso una persona que tenga relación de parentesco

con alguna de las partes. Sin embargo, al final, la regulación actual relativa a restricciones concretas no deja de ser una limitación de hechos de carácter objetivo.

Fuera de lo anterior, hay una serie de factores psicológicos inherentes a la figura del ser humano que, en primer lugar, no quedan regulados porque no se pueden regular (o al menos es muy complicado) y que, en segundo lugar, pueden influir en la toma de decisiones objetivas del juez, tales como prejuicios o simplemente porque la cuestión de fondo del caso concreto puede tocar más de cerca por cualquier razón. Estas son cuestiones subjetivas.

Sentimientos, emociones, prejuicios, … Son muy numerosas las manifestaciones psicológicas involuntarias, muchas veces provenientes del subconsciente, que pueden viciar la racionalidad del ser humano e influir en su capacidad de interpretación o toma de decisiones (Cruz, 2012).

El hecho de ser actos involuntarios que se escapan del racionalismo, muchas veces provoca que su control sea complicado, influenciando de manera directa, pero involuntaria, las acciones de cualquier persona en el día a día. Por ejemplo, en el ámbito empresarial, la toma de decisiones puras, sin influencias, es algo complicado. Factores como los sentimientos, valores personales o emociones influyen no solo en las decisiones del día a día, sino también en las importantes provenientes desde las esferas directivas (Hernández, 2018). Por todo lo anterior, los jueces no son una excepción a estas influencias al ser, al fin y al cabo, seres humanos.

Si en un proceso, la aplicación de la ley fuese pura y no hubiese que interpretarla, esto no sería un problema, pero en el Derecho español, la ley dicta, aunque siempre pueden darse zonas grises, es decir, zonas que quedan al amparo de evaluación del juez para aplicar la ley de una forma u otra. De aquí que puedan surgir discrepancias y que, en caso de que juzgue un órgano colegiado y no uno unipersonal, esas discrepancias se traduzcan en votos particulares. Para poner un ejemplo de la existencia de votos particulares, sin entrar en detalle, me referiré a una de las sentencias más mediáticas en la historia reciente, la Sentencia 38/2018 de la Sección Segunda de la Audiencia Provincial de Navarra, o más coloquialmente conocida como la sentencia sobre el caso

de “La Manada”, en la que uno de los tres magistrados emite un voto particular no acorde con la mayoría.

Por ello, cabe afirmar que la labor de un juez no es solo dictar sentencia en base a hechos objetivos, sino que son muy numerosas las veces en las que un juez debe realizar un doble proceso interpretativo, por un lado, de la ley, y por otro lado del caso concreto y de las circunstancias que lo envuelven. Este proceso, debiendo ser lo más objetivo y racional posible, es probable que pueda verse influenciado por la psicología involuntaria del juez, la cual puede llegar a viciar el proceso racional que debe realizar cada juez, induciéndole a fallar o argumentar de una u otra forma en base a factores no objetivos o irracionales, poniendo en jaque la idea de la imparcialidad judicial. Es importante no confundir este proceso previo con el argumentativo como tal. La influencia psicológica puede motivar a decantarse por un lado u otro, pero el proceso argumentativo y de interpretación posterior si es puramente racional.

Es por esto por lo que muchos autores argumenten que los sesgos están presentes en todo proceso racional jurídico no solo en la parte de la abogacía (claramente sesgada al tener que defender los intereses de una parte en concreto), sino también en la de los jueces. Para autores como Muñoz (2011), los jueces que potencian el estudio de su inconsciente y trabajan en técnica de objetivación, reducen su sesgo psicológico y logran potenciar la racionalidad y objetividad en sus escritos y, por tanto, en los procesos judiciales que llevan a cabo.

Sin embargo, no debe pasar por alto que la nota característica de estas emociones es la involuntariedad por las que se generan y se exteriorizan. Esto no supone tan solo un problema tal y como he explicado, sino que también brinda la posibilidad de poder identificarlas. Existen algoritmos de *Text Mining* y *Data Mining*, aunados bajo el concepto de *Sentiment Analysis* y que estudiaré en puntos posteriores, los cuáles permiten la captación y extracción de sentimientos plasmados en textos, y las sentencias no son una excepción a estos.

Por tanto, tal y como explicaré más adelante, la extracción de sentimientos de un texto puede ser algo interesante para ver si, a día de hoy, el sentimiento neutro es posible.

2.2.3. La problemática de las bases de datos de jurisprudencia. La conjunción con el Topic Modeling

Tal y como he explicado anteriormente, existen diversos tipos de bases de datos de documentos relacionados con el Derecho, de las cuáles las más típicas son las bases de datos que recogen sentencias, o, dicho con otras palabras, jurisprudencia. Es en estas bases de datos donde me centraré en el presente trabajo y no tanto en las que recogen compilaciones de textos de otra índole, tales como manuales o artículos de revista.

Por antonomasia, dentro de las bases de textos jurisprudenciales, la más importante y gratuita, abierta a público, es la base de datos del Poder Judicial. La diferencia en cuanto al resto de bases la marca en dos sentidos distintos. Por una parte, en cuanto a contenido y su veracidad, y por otra parte, en cuanto a la cantidad de información que recoge y lo completa que es. Es esto lo que la hace diferenciarse del resto, porque a nivel servicios tipo buscador y filtrado que proporciona, son realmente parecidos al ofertado por otras plataformas.

Si la búsqueda que se desea realizar es la búsqueda de un documento individual de forma concreta, para poder encontrarlo de forma inmediata, es necesario contar con los datos exactos del mismo (número de proceso, jueces, ponentes, …), algo que, si se es parte en el proceso, es fácil de conocer, pero si se es ajeno a ello, es más complicado de tener. La otra forma de búsqueda que posibilita realizar es la búsqueda por palabras o frases concretas, gracias a un buscador de palabras donde a modo de sugerencia, van saliendo frases complementarias a la información introducida por el usuario (por ejemplo, de poner la palabra “asesinato”, el buscador recomendará frases como “asesinato con alevosía”, “asesinato con dolo” o “asesinato en tentativa” entre otras). A la par de poder filtrar mediante palabras o frases, se pueden introducir una serie de filtros complementarios, como puede ser el órgano o sala encargada de dictar sentencia, el tipo de resolución, y el año de resolución (son filtros más limitados en cuanto a concreción que los ofrecidos en el primer tipo de búsqueda que se puede realizar, el de resolución concreta).

No hay más formas de buscar que estas dos. Como se puede apreciar, si la búsqueda realizada tiene el fin de encontrar una resolución concreta, o unas palabras muy

específicas que pueden limitar mucho las sentencias mostradas por la base tras el filtrado, la base y el funcionamiento de su búsqueda son especialmente eficaces. Si, por otro lado, el filtrado es por un tema en general, que puede estar representado o no por una serie de palabras que pueden ser consideradas como comunes, al aparecer en muchas resoluciones, entonces el buscador no es tan eficaz, por muchos filtros complementarios que puedan introducirse. Esto, por ejemplo, para trabajos de investigación es un problema. No poder realizar búsqueda de resoluciones por temas abiertos y no reducidos a palabras retrasa el proceso de investigación.

Por otro lado, no existe un filtrado como tal de sentido de la resolución, y las sentencias no tienen resúmenes. Es decir, el filtrado por palabras posibilita que aparezcan solo sentencias que contengan las palabras buscadas, pero tiene dos problemas. Por una parte, puede que las palabras no sean el tema principal de la resolución, apareciendo como mera referencia, o pasando muy por encima de ellas. Por otra parte, no es lo mismo una resolución argumentando a favor de que en un caso medie, por ejemplo, asesinato con alevosía, que una argumentación en contra de la existencia de esta en otro caso distinto, pero el filtrado por palabras muestra las dos. Al carecer las sentencias de resúmenes como tal, el hecho de tener que identificar otras cuestiones que no son la presencia o no de las palabras buscadas, hace que el proceso de recopilación de información concreta vía buscadores sea costoso y lento por lo general.

Es muy probable que la implementación de algoritmos como el *Topic Modeling*, relativos al *Data Mining* y que explicaré posteriormente, favorezcan a poder obtener una búsqueda más eficaz en cuanto a la búsqueda de resoluciones abierta. El hecho de poder extraer mucho más contenido de los textos incluidos en la base de datos (como puede ser por ejemplo sentido del fallo o tema general de la resolución), posibilitaría crear un espectro mayor de filtros y, por tanto, poder realizar una búsqueda de mayor calidad y velocidad.

3. EXPLICACIÓN DE LAS TÉCNICAS A EMPLEAR

Tal y como mencionaba en la introducción, en este trabajo se pretende implementar una serie de técnicas relativas al *Text Mining* sobre documentos de carácter jurídico, con el fin de analizar y proponer soluciones a los problemas expuestos en puntos anteriores.

Antes de proceder a la implementación práctica de los algoritmos, es necesario la aclaración y estudio teórico de determinados conceptos claves que son la base de todo proceso de *Text Mining*.

3.1. Conceptos relativos al Text Mining

3.1.1. Términos y documentos. La base del Text Mining

Siguiendo a Shmueli et al. (2018), todo análisis de textos basa su funcionamiento en la identificación por algoritmo de dos conceptos distintos, por un lado, los términos, y por otro, los documentos:

a.- Términos: Los términos son todas y cada una de las palabras que conforman los textos. Este concepto no depende del número de caracteres del que esté formado cada palabra.

b.- Documentos: Los documentos son agrupaciones de términos. Estos pueden ser frases o figuras más extensas tales como obras literarias, sentencias, *Tweets*, … Por tanto, depende de cada programador o de cada algoritmo aplicado a un texto qué se considera documento y cuántos puede tener un texto completo. Al conjunto de documentos se le conoce como *corpus*, es decir, si por ejemplo, cada documento es una frase, el *corpus* será el conjunto de frases del texto, si cada documento es un *Tweet*, el *corpus* será el conjunto de *Tweets* objeto de estudio, o si cada documento es una sentencia, el *corpus* será el conjunto de sentencias.

3.1.2. Bag of Words

El BOW (*Bag of Words*) puede ser entendido como el análisis de las características propias de cada documento (características entendidas como número de palabras claves en un texto y su frecuencia de aparición). Por tanto, el resultado de todo BOW, es decir, lo que se puede extraer, es un vector en el que cada palabra clave está representada con un número (asociado al número de repeticiones de la palabra) en el mismo, y donde la longitud del mismo vendrá dada según el número de palabras, consideradas como clave, que se hayan determinado que existen. Es importante considerar que cada palabra es considerada una entidad única, es decir, que si una palabra se repite, las repeticiones no se deben tener en cuenta.

Los vectores de frecuencia utilizados a modo de representación del BOW son variados, y los hay tanto simples como complejos. El modo de representación más simple de vectores es una tabla o matriz que recoja en cada columna cada término, y cada fila corresponda a un documento, donde cada celda recoja el número de repeticiones de cada término (matriz TDM). Para entender esto mejor, procederé a realizar un ejemplo. Supongamos que tenemos dos documentos. El primero es "Juan es listo", y el segundo es "Juan es guapo. Juan y Pedro son hermanos".

	Juan	Es	Listo	Guapo	Y	Pedro	Son	Hermanos
Doc 1	1	1	1	0	0	0	0	0
Doc 2	2	1	0	1	1	1	1	1

Figura 1. Ejemplo matriz TDM. Fuente: Elaboración propia

El vector del documento 1 sería [1,1,1,0,0,0,0,0].

El vector del documento 2 sería [2,1,0,1,1,1,1,1].

En este caso, la longitud ambos vectores es 8 al tener 8 palabras clave en el *corpus*.

El problema del uso del enfoque BOW es que no tiene en cuenta el orden de las palabras, tan solo la frecuencia de aparición de las mismas. Por ello, todo algoritmo que utilice un enfoque BOW, tiene el peligro de perder información relativa al sentido o significado del *corpus* ya que el orden de las palabras de un texto suele afectar al significado del mismo. A modo de ejemplo, no es lo mismo decir que "Juan ha comprado

un coche a Pedro", a decir que "Pedro ha comprado un coche a Juan". En cada frase el comprador del coche es una persona distinta, y por tanto, el sentido de ambas no es el mismo, aunque la acción descrita (comprar un coche) sea la misma en las dos.

3.1.3. Matriz TF-IDF

La matriz TF-IDF (*Term Frequency – Inverse Document Frequency*) permite mejorar la representación vectorial de la BOW. Cuando se hablaba de la BOW y su representación vectorial, se mostraba tan solo la frecuencia de aparición de determinados términos en el *corpus*, y se obviaba totalmente el contexto en el que se movía este último. Esta matriz permite normalizar la frecuencia de los términos con respecto el *corpus*. Para ello, se calculan dos frecuencias:

a.- Frecuencia de cada término (tf): Representa el número de veces que aparece cada término (t) en un documento (d), es decir, su frecuencia (f). El cálculo se realiza mediante la siguiente fórmula: $tf\ (t,d) = \frac{f(t,d)}{longitud\ (d)}$

b.- Frecuencia inversa del documento (idf): Representa la importancia de un término (t) en un *corpus*. Para ello toma como referencia el número (n) de documentos que forman el *corpus*, y el número de documentos que contienen el término concreto (n'). El cálculo de esta frecuencia es el siguiente: $idf\ (t) = \log(\frac{n}{n'})$

La matriz TF-IDF viene a solventar un problema relativo al término de frecuencias. Es común que en un texto haya palabras que, aun repitiéndose mucho, pueden no ser tan relevantes para el *corpus*, puesto que son palabras que se repiten en muchos documentos de éste. Un caso claro son las preposiciones, por ejemplo, palabras muy comunes que aparecen repetidas en muchos documentos de un *corpus* (aunque éstas suelen ser consideradas *stopwords* y, por tanto, eliminadas). Por otro lado, imaginemos un texto donde se hable de composiciones musicales. Palabras como "música" deberían aparecer en muchos documentos y, salvo que se considere como *stopword* de forma manual, no debería ser eliminada. En este supuesto, la tf de la palabra "música" debería ser muy alta, pero el término en sí no debería aportar mucha información al texto dada su temática, es decir, no es un término relevante. Para solucionar esto, es necesario multiplicar a la tf por

un factor que tenga en cuánta cómo de frecuente es este término en el *corpus*, es decir, la idf. Por ello, la TF-IDF se calcula de la siguiente manera:

$$Tf - idf\ (t, d) = tf\ (t, d)\ x\ idf\ (t)$$

Una vez calculado cada valor de cada término, es ese el que va en el vector de representación de frecuencias de la BOW y, por tanto, en cada celda de la matriz Documentos-Términos.

3.1.4. Tokenización

Tal y como mencionaba anteriormente, todo *corpus* debe poder dividirse en términos. Este proceso de división es el conocido bajo el nombre de *Tokenización*, proceso en el que cada palabra será conocida como *token*, y cada *token* será habitualmente una columna en la matriz TDM, tal y como se ha visto anteriormente.

El problema de este sistema es que es común encontrar documentos con muchos *tokens* que no solo dificultan este proceso, sino que también dificultan la implementación de otras técnicas de *Data Mining* posteriores al considerar que cada *token* es una variable en estas últimas técnicas. Por ello es importante implementar técnicas de preprocesamiento de textos para dar solución a esto.

3.2. El preprocesamiento de textos y sus técnicas

Existen distintas técnicas de preprocesamiento de textos, aunque las más comunes son las técnicas de reducción de textos. Con el uso de estas técnicas se pretende reducir la cantidad de *tokens* en un texto y, por tanto, reducir la longitud de este último.

Hay distintas formas de conseguir este resultado. La más intuitiva, el eliminar palabras, aunque también se dan otras tales como la eliminación de signos de puntuación, convertir el texto a minúsculas o la eliminación de números. Hay muchas palabras en todo texto (como pueden ser determinantes o preposiciones) que aparecen repetidas muchas veces para dar sentido al texto pero que no aportan nada de información al mismo.

Ejemplos de este tipo de palabras pueden ser “y”, “o”, “la” o “el”, y a cada una de estas palabras se les conoce bajo el término de *stopword*.

Para poder eliminar *stopwords*, la forma más sencilla en R es mediante la implementación de paquetes. Estos paquetes incluyen un listado de palabras consideradas como *stopwords* que son eliminadas tras el uso de los mismos.

Además de esta forma de eliminar palabras que no aportan nada de información al texto, existen otra serie de técnicas complementarias tales como el Stemming y la Lematización.

3.2.1. Stemming

El *Stemming* es una técnica de reducción de textos centrada en la eliminación de prefijos y sufijos de los distintos *tokens* del *corpus*, con el fin de obtener la raíz de los mismos, es decir, la parte que de verdad aporta información al texto. Tal y como describe Panessi (2001), el algoritmo más utilizado para el uso de esta técnica es el de Porter.

Programado en 1980, el algoritmo de *Stemming* de Porter permite la eliminación de prefijos y sufijos mediante la elaboración de reglas concretas. Para ello, debe identificarse los prefijos y los sufijos objeto de búsqueda y establecerse el número mínimo de caracteres que debe tener la raíz de cada *token* una vez se le ha eliminado el prefijo o sufijo concreto.

Esta técnica de normalización tiene dos peligros, el *overstemming* y el *understemming*:

a.- El *overstemming* consiste en la eliminación de sufijos o prefijos determinados y de distinta longitud que generan raíces iguales a partir de palabras con significados completamente distintos. Un ejemplo sería “universo” y “universidad”. Determinados algoritmos de *Stemming* podrían dejar ambas palabras el la raíz “univers–“, por lo que se perdería información.

b.- El *understemming* consiste en la reducción a raíces distintas de palabras que deberían tener la misma raíz. Un ejemplo serían las palabras en inglés "*data*" y "*datum*". Los algoritmos de *Stemming* podrían dejar en raíces "dat-" y "datu-", cuando deberían compartir raíz.

3.2.2. Lematización

La lematización es un proceso de normalización de palabras que, a diferencia del Stemming, busca obtener el lema de una palabra de forma normalizada para que aplique a una gran cantidad de t*okens* presentes en el *corpus*. Por ejemplo, si en el *corpus* se encuentran las palabras "*working*", "*works*" y "*worked*", el algoritmo las normalizará y las contará tan solo bajo su palabra primitiva, es decir, "*work*" (Plisson et al. 2004), o por ejemplo en la palabra "*sharing*", donde tras aplicar Lematización, quedaría "*share*", mientras que si se aplica *Stemming*, quedaría "*shar-*".

El problema de la Lematización se produce cuando el algoritmo se enfrenta a lenguas en las que la palabra de origen tiene muchas palabras derivadas y, por tanto, este proceso resulta ser muy lento.

3.3. Sentiment Analysis

El *Sentiment Analysis* u *Opinion Mining* puede ser definido como el conjunto de técnicas destinadas al estudio de opiniones, sentimientos, y emociones expresadas en un texto (Keith et al. 2019), o lo que es lo mismo, es el conjunto de técnicas utilizadas que, mediante el análisis de textos concretos, es decir, palabras, frases, construcciones gramaticales y léxicas, … son capaces de inferir los sentimientos plasmados en el texto que fueron transmitidos por el autor en el momento de su escritura.

Tal y como mencionaba anteriormente, el mundo de las emociones y sentimientos es, cuanto menos, subjetivo, es decir, no hay normas establecidas, y la percepción de los mismos varía según cada persona. Esto hace que la clasificación de los sentimientos sea algo complicado y no esté estandarizado. Sin embargo, en general, dos son las formas

mediante las cuáles estas técnicas pueden clasificar los sentimientos mostrados en un texto:

a.- Dicotomía positiva-negativa: Se basa en la generalización de los sentimientos plasmados en un texto en sentimientos positivos y sentimientos negativos, es decir, en la polaridad de sentimientos de un texto, entendida ésta última como la valoración a través la cual se clasifica el mensaje de un texto en base a la intención que tenga el autor cuando lo realiza, esto es, positivo, negativo o neutro. Lo anterior no implica que el algoritmo no sea capaz de identificar más sentimientos, pero la programación del mismo hace que muestre como resultado tan solo si lo plasmado se refiere a sentimientos positivos o negativos.

b.- Dualidad sentimientos – emociones: En este tipo de algoritmos se diferencia los sentimientos de las emociones. En el Lexicón NRC de Mohammad (2020) (explicaré a continuación qué es un lexicón), por ejemplo, por un lado entiende que los sentimientos son positivos o negativos, pero las emociones son muchas y variadas (cada una de ellas asociada a un sentimiento positivo o uno negativo), tal y como muestra la siguiente figura.

	# TÉRMINOS	CATEGORÍAS
LEXICÓN NRC	14.182 unigramas (palabras)	Sentimientos: Positivos o negativos
	25.000 sensaciones asociadas a sentimientos y emociones	Emociones: Enfado, anticipación, disgusto, miedo, alegría, tristeza, sorpresa y confianza

Figura 2. Resumen lexicón NRC. Fuente: Elaboración propia a partir de Mohammad (2010)

De la anterior clasificación de formas de mostrar los sentimientos y emociones, se puede inferir que estos algoritmos pueden funcionar de formas distintas. Así, una clasificación genérica de estos algoritmos según el enfoque de funcionamiento usado puede ser la siguiente (D'Andrea et al. 2015):

a.- Enfoque de *Machine Learning*: Utilizado para identificar y predecir la polaridad de los sentimientos del texto mediante la identificación de los mismos a nivel documento.

Se basa en la aplicación de algoritmos de *Machine Learning* para realizar su función, tales como los vectores SVM (*Support Vector Machines*) o el modelo de clasificación Naïve Bayes, es decir, algoritmos de aprendizaje supervisado. La principal ventaja es que tienen un alto nivel de adaptación a cada caso concreto ya que deben funcionan adaptándose al conjunto de datos concreto.

b.- Enfoque Lexicón: Un lexicón es un listado o diccionario de palabras creado de antemano. En este enfoque, cada palabra del lexicón aplicado está asociada a un sentimiento concreto. Las principales ventajas frente al enfoque anterior son que en éste, en primer lugar, la precisión es mayor al usar diccionarios de palabras extensos creados de antemano para su funcionamiento, asociando las palabras a sentimientos concretos en ellos y que, en segundo lugar, funciona mejor con cuerpos de datos nuevos por esa misma razón. Por otro lado, la principal desventaja es que es un tanto limitado ya que el número de palabras del lexicón es finito, al igual que la asociación de cada una con sentimientos específicos. Mohammad (2020) también destaca que las emociones no son términos inmutables, sino que son subjetivos. Entiende que el riesgo de usar un lexicón es que puede no funcionar bien dada la rigidez del mismo. Un ejemplo puede ser la palabra "fiesta". Según este autor, esta palabra suele estar al concepto de alegría, pero en sí misma, no denota la emoción alegría. Un ejemplo de lexicón altamente aplicado es el NRC, el cual, por ejemplo, tiene implementadas las ocho emociones básicas de Plutchik (1984), es decir, alegría, confianza, miedo, sorpresa, tristeza, aversión, ira y anticipación (ver figura 2 en la página 30).

c.- Enfoque híbrido: Usa tanto M*achine Learning* como lexicón. En un primer momento, identifica los sentimientos presentes en el texto mediante la captación de palabras y, posteriormente, aplica las técnicas de *Machine Learning* relativas a la clasificación de sentimientos. La principal ventaja de este enfoque es la captación e identificación de los sentimientos es muy específica. Sin embargo, la principal desventaja es que el resultado suele ser poco intuitivo y difícil de entender dada su complejidad.

Por último, mencionar que la principal área de aplicación y estudio de estas técnicas, en los últimos años, ha sido *Twitter*.

Twitter es una red social con un gran impacto en la sociedad donde, en primer lugar, los documentos son relativamente pequeños (*tweets* de menos de 200 caracteres), pero la gran cantidad de publicaciones diarias hace que estos *corpus* sean altamente ricos en información. Por otro lado, al no existir un control rígido de las publicaciones y haber total libertad de expresión, la identificación de sentimientos en estos *corpus* es muy completa, y tiene muchas utilidades, como pueden ser estudios de impacto social de determinadas políticas o noticias.

3.4. Topic Modeling: Algoritmo LDA

3.4.1. Análisis histórico del Topic Modeling

El *Topic Modeling* surge en un primer momento para paliar la necesidad de analizar grandes cantidades de textos en un periodo de tiempo relativamente corto, y más concretamente, surge con el único fin de calcular la frecuencia en que una palabra o un grupo de palabras aparecen en un *corpus* determinado con el fin de obtener un tema concreto de clasificación general.

Sin embargo, en el primer momento de creación del algoritmo, las técnicas de preprocesamiento de textos descritas en puntos anteriores no existían, y por ello, las palabras con una ratio de aparición mayor eran simples conjunciones o determinantes, es decir, palabras cuya información aportada al texto es nula y que hoy en día se eliminan al considerarse *stopwords*. Para paliar este problema, antes de proceder a la eliminación de las palabras, aparece la métrica TF-IDF explicada anteriormente. De esta forma, se pueden obtener ratios de aparición de palabras clave contrastándolos con la importancia de estas en el *corpus* objeto de estudio.

En 1990 aparece el algoritmo LSA (*Latent Semantic Analysis*). Scott Deerwester, en una de sus publicaciones, añade a ese concepto de ratios de aparición la forma de visualización del mismo, es decir, recoge por primera vez el concepto de bolsa de palabras y la forma de graficar mediante la matriz términos-documentos, también definida anteriormente. A partir de esa matriz, el LSA factoriza la misma para extraer la relación existente entre las palabras que la integran, y analizar la ratio de aparición de las mismas (Deerwester et al. 1990).

Más adelante, en 1999, Hoffman, basándose en el LSA, aporta un elemento nuevo a este, el tema. Tal y como describe en su obra, la relación general que existe entre distintas palabras puede ser calificada como tema (Hoffman, 1999). Es la primera vez que este concepto se extrapola al *corpus* con el fin ya no solo de contar palabras o ver relaciones, sino clasificar. Años más tarde, en 2003, se desarrolla el LDA, tomando como base el LSA pero entendido el primero como un modelo capaz generativo de documentos que estudia las estructuras latentes de un texto (Blei et al. 2003). Posteriormente, este algoritmo tan solo se ha ido perfeccionando con el fin de hacerle capaz de analizar textos procedentes de distintas fuentes.

3.4.2. El Topic Modeling

El *Topic Modeling* puede ser definido como un conjunto de técnicas y algoritmos concretos que se pueden implementar sobre textos con el fin de extraer temas concretos que agrupen palabras. El modo de funcionamiento de estos algoritmos, típicamente el LDA (algoritmo objeto a estudio posterior en este trabajo), lo describe Lisa M. Rhody en un documento donde analiza su implementación en textos literarios con lenguaje figurativo. Básicamente, estos algoritmos "generan categorías de temas (*topics*) y clasifican palabras y textos según los mismos" (Rhody, 2012, p.24). También enumera algunas posibilidades que ofrecen estos algoritmos. Concretamente, destaca que el "*Topic Modeling* permite la identificación de patrones ocultos entre palabras en grandes colecciones de textos" (Rhody, 2012, p.24), o lo que es lo mismo, en *corpus* muy extensos. Por tanto, cabe afirmar que un *topic* es un conjunto de palabras que aparecen juntas con mucha frecuencia.

Cuando en puntos anteriores describía los problemas de la búsqueda y filtrado de las bases de datos de jurisprudencia, destacaba la imposibilidad de buscar activamente por temas generales que no se limitasen a palabras concretas. La implementación de algoritmos en el código de búsqueda relacionados con el *Topic Modeling* pueden llevar a resolver este problema ya que, una vez se ha implementado el algoritmo y se extraen los resultados, pueden aplicarse algoritmos de clasificación propios del *Data Mining* que permitan la búsqueda activa por los *topics* extraídos en el desarrollo del algoritmo de *Topic Modeling*. Esto es muy importante, y es que no se debe confundir *Topic Modeling* con clasificación de documentos en base a sus *topics*. El segundo es un proceso de

lenguaje supervisado que toma los *topics* de los textos y clasifica cada texto en base a los primeros, mientras que el primero es un proceso de lenguaje no supervisado que identifica *topics* detectando patrones y frecuencias en el uso de palabras, aunque ambos conceptos van innegablemente de la mano cuando se habla de motores de búsqueda.

Como se ha mencionado en apartados anteriores, los motores de búsqueda jurisprudenciales son limitados y suelen plantear problemas. Generalmente, todo motor de búsqueda funciona de la misma forma, se obtiene información de los documentos, se clasifican, y esto permite el filtrado posterior del usuario. Por tanto, los tres conceptos anteriores (detección de *topics*, clasificación en base a *topics*, y motores de búsqueda) están relacionados. La relación entre *Topic Modeling* y clasificación la desarrollan, entre otros autores, Sarioglu et al. (2016), y es que, según ellos, el proceso de clasificación surge mediante la identificación concreta de *topics* (especialmente eficaz cuando se habla de *Topic Modeling* y LDA). Una vez se tienen los *topics*, se deben implementar algoritmos específicos de clasificación en un set de entrenamiento, para encontrar modelos base que poder aplicar posteriormente a mayor escala en cuanto a datos (estos autores usan ATC y BTC como algoritmos de clasificación de *topics*)[3]. Por último, Zhu (2011) entiende que la clave para mejorar todo motor de búsqueda es la mejora en el proceso de clasificación. Por tanto, una mejor clasificación de *topics* posibilita unos motores de búsqueda mucho más eficaces en cualquier ámbito, incluido el Derecho potencialmente.

Por lo anterior, puede concluirse que la función del *Topic Modeling* es la identificación de *topics* de un documento y la representación de cada documento como una distribución de sus *topics*, aunque en los motores de búsqueda ambos conceptos depende el uno del otro.

Por último, destacar que este tipo de algoritmos funcionan, en general, mediante el BOW, es decir, que tienen los problemas descritos en puntos anteriores asociados al uso del BOW. No considerar el orden de palabras del texto, sino tan solo la frecuencia de

[3] Los algoritmos *Aggregate Topic Classifier* (ATC) y *Binary Topic Classification* (BTC) permiten la clasificación de documentos en base a *topics*. El primero, para clasificar, toma valores promedios de aparición de cada *topic* en el conjunto del *corpus* para clasificar, mientras que el segundo se busca la obtención de dos *topics* y clasifica según el que tenga un mayor valor de aparición.

aparición de las mismas, puede ser un problema al producir pérdida de información necesaria en el *corpus*. Por ello este tipo de algoritmos sirven si el fin del estudio del *corpus* es clasificar el mismo y extraer su temática general. Si se desea realizar un análisis más profundo del *corpus* donde se extraiga información concreta relativa al sentido de este último, los algoritmos a implementar no deben ser aquellos basados en el BOW.

3.4.1. El funcionamiento del algoritmo LDA

El algoritmo LDA, tal y como exponen Silge y Robinson (2021), y perteneciente al *Topic Modeling*, es un algoritmo que funciona mediante el concepto BOW, y que es capaz de extraer *topics* de textos según el uso de palabras en los mismos. El LDA, por tanto, asume que cada *corpus* está formado por un conjunto de documentos y *topics*, y cada *topic* por un conjunto de términos, es decir, palabras. Partiendo de esta base, estas técnicas tratan de asociar a cada documento unos *topics*, asignando a su vez para ello cada palabra a un *topic* concreto. Esta asignación se realiza en términos de probabilidad condicional, es decir, estimando la probabilidad de pertenencia de cada palabra a un *topic*, y el peso de cada *topic* en un documento.

Antes de todo esto, el programador debe determinar el número de topics que desea identificar en el documento. Al considerar cada documento como un conjunto de topics, cada documento podría representarse de la siguiente forma:

$$Di = W1i \ x\, Topic\ 1 + W2i\ x\, Topic\ 2 + \cdots + Wni\ x\, Topic\ n$$

Donde D_i es la distribución de *topics* de un documento, W_{ni} es el peso de un *topic* concreto en el documento, y *Topic* n es un *topic* en particular.

El proceso por el cual funciona el algoritmo LDA, explicado por Sharma (2020), es el siguiente:

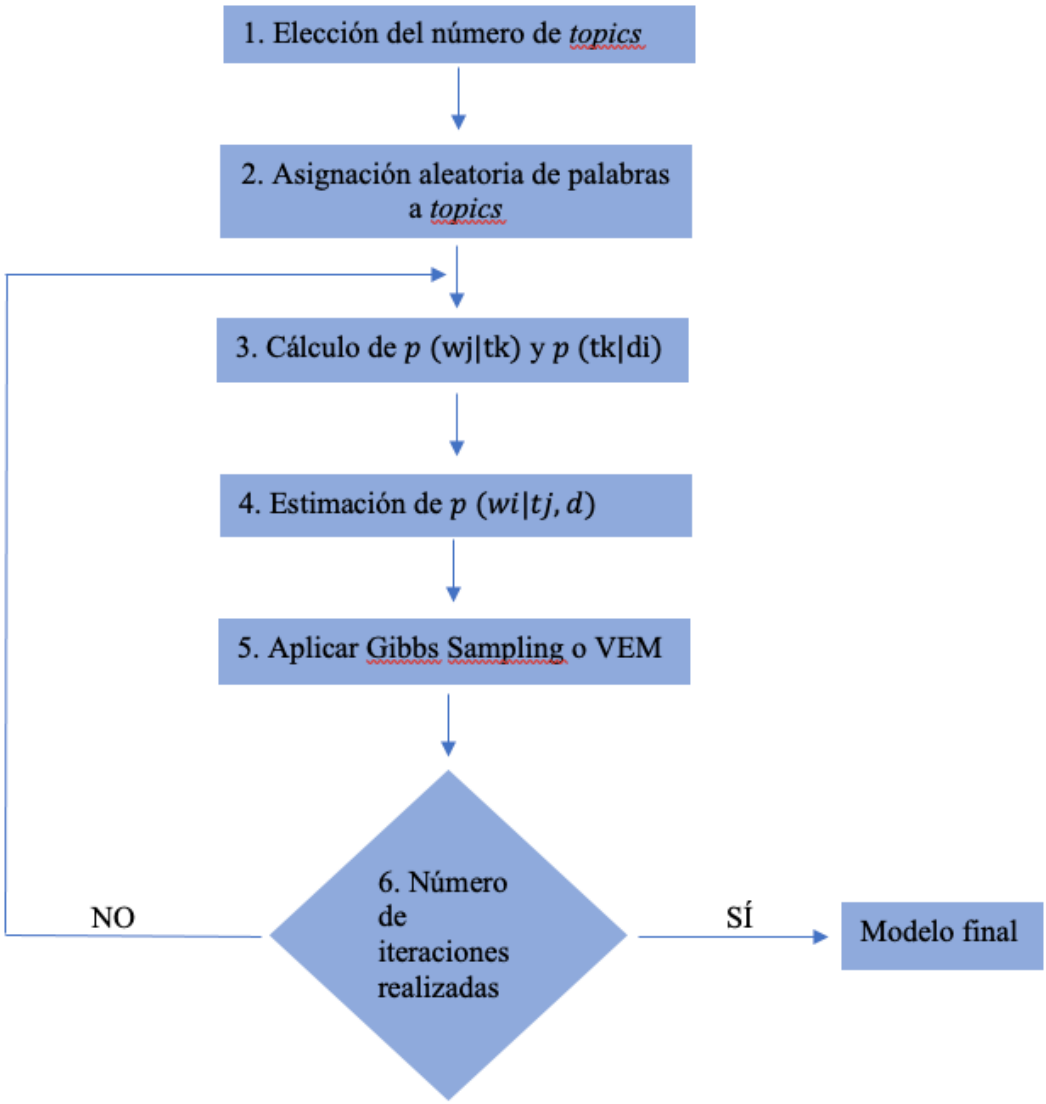

Figura 3. Explicación algoritmo LDA. Fuente: Elaboración propia a partir de Sharma (2020)

1.- Determinación del número de *topics* deseados: Se realiza al comenzar el código. Aunque existen formas de calcularlo de forma automática y mecanizada, en este trabajo se determinará de forma manual tal y como justificaré en puntos posteriores.

2.- Asignación de forma aleatoria a cada *topic* de las palabras. Ninguna palabra queda sin ser asignada y, en este punto del proceso, cada palabra puede ser asignada a varios *topics* a nivel instancias de la misma, es decir, una palabra que aparece repetida varias veces puede ser clasificada cada vez que aparece en un *topic* distinto. Esto se realiza iniciando un bucle en el *corpus* que recorre cada documento, asignando un *topic* a cada palabra presente en cada documento.

3.- Para cada documento del *corpus*, recorrer todas las palabras del mismo y calcular:

1. La probabilidad (p) de cada palabra (w_j) de aparecer en un *topic* concreto (t_k) con respecto al resto de palabras incluídas en ese mismo topic, es decir, $p\ (\mathrm{wj}|\mathrm{tk})$.

2. La probabilidad de que un *topic* (t_k) aparezca en un documento concreto (d_i), es decir, $p\ (\mathrm{tk}|\mathrm{di})$.

4.- Actualizar $p\ (wi|tk, d)$ a través de la siguiente igualdad:

$$p\ (wj|tk, di) = p\ (wj|tk)x\ p\ (tk|di)$$

5.- Aplicar *Gibbs Sampling* o VEM, es decir, reasignar cada palabra presente en el documento a otro *topic* bajo el estándar de $p\ (wj|tk, di)$

6.- Se repiten los pasos 3 a 5 hasta completar un número de iteraciones concreto y determinado previamente.

Por último, destacar que este proceso del LDA depende de 3 parámetros:

a.- α, el cual controla el número esperado de *topics* en cada documento de texto (densidad documento – *topic*)

b.- β, el cual controla la distribución de palabras por cada *topic*.

c.- K, el cual es el número de *topics* a considerar y es elegido, o bien por el programador a propia voluntad, o calculado mediante algoritmos específicos.

Una vez el proceso ha finalizado, se puede obtener el peso que tiene cada *topic* para cada documento. Este peso es representado bajo el parámetro gamma (γ). Al ser cada documento un conjunto de *topics*, la suma de los gamma de un documento concreto, siempre tiene que resultar ser 1.

3.4.2. Evolución del algoritmo LDA: Aplicaciones y avances

Tal y como describía anteriormente en el punto relativo a la historia del *Topic Modeling*, las actualizaciones más recientes en el algoritmo LDA se han basado sobre todo en aumentar las fuentes de las que pueden proceder los datos objeto de análisis. Estas actualizaciones han posibilitado expandir el algoritmo LDA y poder aplicarlo a fuentes de datos que no son textos (aunque la fuente típica de procedencia de datos siga siendo esta). Así, la evolución del algoritmo LDA a lo largo de los años desde su publicación ha sido la siguiente:

a.- Desvinculación del LDA con el *bag of words*: Explicaba en puntos anteriores que no solo en el LDA, sino que casi todos los algoritmos que integran el *Topic Modeling* basan su funcionamiento en el *bag of words*, es decir, que el orden de las palabras no es algo importante de cara a la clasificación. Sin embargo, tal y como expone Wallach (2006), se ha logrado desvincular al algoritmo LDA del *bag of words*, lo que posibilita un mayor conocimiento de los documentos integrados en el *corpus*.

b.- Jerarquización de los documentos que forman el *corpus*: El *corpus*, tal y como he explicado anteriormente, es un conjunto de textos o documentos objeto de estudio por las técnicas de *Text Mining*. Sin embargo, cuando el algoritmo se publicó, todos estos documentos tenían la misma ponderación en cuanto al *corpus*, es decir, y concretando ya con el LDA, los temas obtenidos por los documentos tenían el mismo peso en el *corpus* porque los documentos a los que hacían referencia pesaban lo mismo en éste. Sin embargo, mediante modificaciones en cuanto a ponderaciones de los documentos, se ha posibilitado la jerarquización de los temas asociados a cada *corpus* con el fin de realizar una clasificación de este mucho más precisa (Blei et al. 2010).

c.- Evolución de los temas: Para poder analizar la evolución de un tema es necesario hacer sobre el mismo un análisis cronológico. Tal y como expone David Blei, la evolución del LDA al *Dynamic Topic Modeling* posibilita el análisis de un tema a lo largo del tiempo y no reducirlo tan solo a un momento y texto concreto (Blei y Lafferty, 2006).

d.- Fuentes de datos ajenas a textos: Fundamentalmente sobre imágenes. Las imágenes pueden ser entendidas como la consecución de patrones visuales siguiendo un

orden concreto y con una proporción concreta. La gran similitud de esta formación con los textos (por ejemplo, cada patrón visual en una imagen puede asemejarse a una palabra en un texto) posibilita que el LDA pueda aplicarse no solo en la clasificación de imágenes (Fei-Fei y Perona, 2005), sino también es el reconocimiento de objetos a partir de imágenes o en tiempo real (Li et al. 2010). Este es el ámbito de mayor crecimiento y que se ha potenciado más en la evolución reciente del LDA.

4. APLICACIÓN PRÁCTICA DE LOS ALGORITMOS

En este capítulo procederé a la explicación y desarrollo del código utilizado en RStudio para este trabajo. Para ello, comenzaré explicando brevemente la razón concreta por la que se escogió el *Data Set* utilizado, explicaré los códigos y terminaré analizando los resultados obtenidos. Posteriormente, por un lado, se implementará un algoritmo de *Sentiment Analysis* basado en el uso del lexicón NRC a un texto jurídico, con el fin de comprobar la carga de emociones y sentimientos presentes en él. Por otro lado, se implementará el algoritmo LDA al *corpus* de sentencias para comprobar cómo esta técnica puede aplicarse a este tipo de documentos y comprobar si la extracción de *topics* que realiza es efectiva.

A continuación se explicará cómo se ha realizado la selección del *corpus* a utilizar.

4.1. Datos

Los datos que se utilizarán en ambos algoritmos (tanto *Sentiment Analysis* como LDA) serán una compilación de documentos formados por treinta autos y sentencias procedentes de distintas instancias judiciales. Por motivos del algoritmo LDA, que se explicará más adelante en el punto 4.3, la compilación de treinta sentencias estará formada por bloques de seis sentencias en cuanto a temática, siendo ésta homicidio, sucesiones, asilo internacional, hipotecas y cláusula suelo y abuso y agresión sexual, todas ellas obtenidas de la base de datos del Consejo General del Poder Judicial (2021).

Homicidio	SAP BA 48/2021, de 29 de marzo	SAP O 134/2021, de 31 de marzo	SAP OU 58/2021, de 25 de marzo	SAP OU 69/2021, de 6 de abril	SAP SG 4/2021, de 15 de marzo	STS 284/2021, de 30 de marzo
Sucesiones	STS 134/2019, de 6 de marzo	STS 267/2019, de 13 de mayo	STS 375/2019, de 27 de junio	STS 384/2019, de 2 de julio	STS 468/2019, de 17 de septiembre	STS 578/2019, de 5 de noviembre

Asilo	AAN 254/2021, de 23 de marzo	SAN sobre el recurso 280/2019, de 5 de marzo de 2021	SAN sobre el recurso 187/2019, de 12 de marzo de 2021	SAN sobre el recurso 196/2019, de 4 de marzo de 2021	SAN sobre el recurso 336/2019, de 9 de marzo de 2021	SAN sobre el recurso 2027/2019, de 11 de marzo de 2021
Hipotecas / Cláusula Suelo	STS 125/2021, de 8 de marzo	STS 131/2021, de 9 de marzo	STS 126/2021, de 8 de marzo	STS 149/2021, de 16 de marzo	STS 151/2021, de 16 de marzo	STS 155/2021, de 16 de marzo
Agresión y abuso sexual	SAN 4/2021, de 18 de marzo	SAP C 159/2021, de 25 de marzo	SAP SG 8/2021, de 29 de marzo	SAP TF sobre el recurso 61/2020, de 18 de marzo de 2021	STS 268/2021, de 24 de marzo	STS 258/2021, de 18 de marzo

Figura 4. Compilación de sentencias que conforman el data set. Fuente: Elaboración propia

4.2. Sentiment Analysis

4.2.1. Instalación de paquetes y carga de librerías

Lo primero que se debe realizar es la instalación y carga de los paquetes necesarios. Para la implementación del algoritmo de *Sentiment Analysis* utilizaré los siguientes paquetes:

1.- "Pdftools": Los datos que utilizaré, anteriormente detallados, son textos en formato pdf. El paquete pdftools permite extraer el texto de dichos documentos para poder trabajar con él.

2.- “Tm”: Paquete especializado para la realización de técnicas de *Text Mining*. Será el paquete utilizado para realizar la limpieza del texto y todas las etapas relativas al preprocesamiento de éste.

3.- “NLP”: Paquete asociado al paquete “tm”.

4.- “Syuzhet”: Paquete específico para realizar el proceso de *Sentiment Analysis*. En él, tal y como especificaré posteriormente, puede encontrarse el lexicón NRC. La razón por la cual uso este paquete es porque el lexicón que ofrece es de los más completos en cuanto al español ya que no hay muchos paquetes que ofrezcan diccionarios con palabras en castellano. Sin embargo, puede que la forma por la que se genera este diccionario en otras lenguas que no sean el inglés es poco fiable, pues el lexicón NRC genera estos diccionarios en otros idiomas mediante el uso del traductor ofrecido por Google en su web, traduciendo la palabra utilizada al inglés, y contrastándola con el lexicón que tiene en este idioma.

5.- “Ggplot2”: Paquete utilizado para graficar los resultados.

4.2.2. Carga de datos y creación del corpus

```
documentos <- list.files(pattern = "pdf$")
corp <- Corpus(URISource(documentos),
readerControl = list(reader = readPDF))
```

Tras la carga de paquetes y librerías, se procede a la lectura de datos y la creación del *corpus*. Para cargar los datos se utiliza la sentencia *list.files(pattern = "pdf$")*, procedente del paquete “pdftools”. Esta sentencia automatiza el proceso de carga, y selecciona todos los documentos en formato pdf presentes en el directorio de trabajo. Quedarán guardados bajo el objeto *documentos*, el cual es un vector formado por cada documento.

Tras esto, se procede a la creación del *corpus*. Para ello, y también procedente del paquete “pdftools”, se utiliza la función *Corpus()*. En este caso, con el argumento *URISource* se indica a la función *Corpus()* que los datos cargados son un vector, y con el

segundo argumento, se indica que los documentos están en formato pdf. El objeto del cual se leerán los textos es *documentos* y, una vez implementada esta sentencia, éstos quedarán recogidos bajo el objeto *corp*.

4.2.3. Preprocesamiento del texto: Limpieza del Corpus

```
corp <- tm_map(corp, content_transformer(tolower))
corp <- tm_map(corp, removeWords, stopwords("spanish"))
corp <- tm_map(corp, removeWords, stopwords(kind = "es"))
p.stopwords <- c("jurisprudencia")
corp <- tm_map(corp, removeWords, p.stopwords)
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, removeNumbers)
corp <- tm_map(corp, stripWhitespace)
corp <- tm_map(corp, stemDocument)
```

Ya se ha explicado en puntos anteriores cuáles son pasos más comunes en cuanto a preprocesamiento de textos, por lo que no entraré en detalle. En este punto, mediante la función *tm_map()*, procedente del paquete "tm", y siempre sobre el objeto *corp*, se procederá a la limpieza del texto. *Tm_map()* requiere indicar, primero, el objeto sobre el que realizarlo, y en segundo lugar, la función concreta de limpieza. Así, con *content_transformer(tolower)* el texto se transforma en minúsculas, con *removeWords, stopwords("spanish")* y con *removeWords, stopwords(kind = "es")* se implementan dos diccionarios (lexicón) de *stopwords* en castellano (palabras comunes en textos que no aportan significado) y se eliminan del objeto *corp* aquellas palabras presentes en dichos diccionarios, con *removePunctuation* se eliminan los signos de puntuación, con *removeNumbers* se eliminan los valores numéricos, con *stripWhitespace* se eliminan los espaciados innecesarios presentes en el texto y, por último, con *stemDocument*, se aplica *Stemming* sobre las palabras del texto. Destacar que la *Tokenización* se hace automáticamente.

Por otro lado, el objeto *p.stopwords* es un vector con un término, "jurisprudencia", creado por voluntad propia. La razón es que dentro del *Corpus*, la sentencia *get_nrc_sentiment* que explicaré posteriormente toma como documentos a cada página

del texto. El encabezado de cada página de las sentencias contiene dicha palabra, por lo que la considero como una *stopword*. Para eliminarla, como he dicho, se debe crear un vector, y bajo el argumento *removeWords* se elimina.

Una vez este proceso ha finalizado, el objeto *corp* no será un vector, sino que será una lista.

4.2.4. Generación de sentimiento y asociación a palabras

```
corp_ch <- unlist(corp)
corp_sent <- get_nrc_sentiment(corp_ch, cl = NULL, language = "spanish", lowercase = TRUE)
```

Finalizado el preprocesamiento del *corpus*, se procede a implementar las sentencias relativas al algoritmo de *Sentiment Analysis*. El objeto *corp*, tal y como he mencionado antes, tras el preprocesamiento y la limpieza ha dejado de ser un vector para ser una lista de treinta objetos (uno por documento). Sin embargo, para que la función *get_nrc_sentiment()* pueda leerlo, necesita volver a vectorizarse. Por ello, extraemos el contenido de la lista mediante la función *unlist()* presente en el paquete base de RStudio. Dicho vector se guarda en el objeto *corp_ch*. Fuera de lo anterior, es importante considerar que para la función *get_nrc_sentiment()*, cada página de los archivos de sentencias originales será considerado como un documento.

A continuación se aplica la función *get_nrc_sentiment()* del paquete "syuzhet" al objeto *corp_ch*, y se le pasa el argumento *language = "spanish"* para indicar así que el lexicón NRC a utilizar es el que está en castellano, y no en otro idioma. Este lexicón será el encargado de asociar a cada palabra un sentimiento. Para ello coge el vector y crea documentos (páginas, en este caso). Otros diccionarios disponibles en el paquete "syuzhet" son *bing* y *afinn*. Sin embargo, utilizo NRC porque es el único que posibilita un estudio detallado de sentimientos y emociones, así como polaridad positiva, negativa y neutra del texto. Los otros dos diccionarios solo muestran la polaridad.

Tras aplicar esta función, la información se guarda en el objeto *corp_sent*, el cual se graficará a continuación.

4.2.5. Graficación y análisis de resultados

Una vez se tiene el objeto con la asociación de palabras a sentimientos y emociones, se puede graficar. Para ello, se deben sumar la cantidad de documentos creados por la función *get_nrc_sentiment()* para cada emoción y cada sentimiento (aunque en este caso se mostrarán los datos en forma de porcentaje en el gráfico sobre emociones para ver el grado de aparición de las misas en el corpus). Me gustaría recordar que el NRC pertenece a la clasificación de sentimientos – emociones tal y como expliqué en puntos anteriores, por lo que se puede analizar las emociones plasmadas y la polaridad positiva-negativa del texto.

```
h.sent<-data.frame(t(corp_sent))
h.sent_1 <- data.frame(rowSums(prop.table(h.sent[2:492])))
names(h.sent_1)[1] <- "Cantidad"
h.sent_1 <- cbind("Emociones" = rownames(h.sent_1), h.sent_1)
rownames(h.sent_1) <- NULL
h.sent_2<-h.sent_1[1:8,]
quickplot(Emociones, data=h.sent_2, weight=Cantidad, geom="bar", fill=Emociones, ylab="Contador")+ggtitle("Gráfico emociones")
```

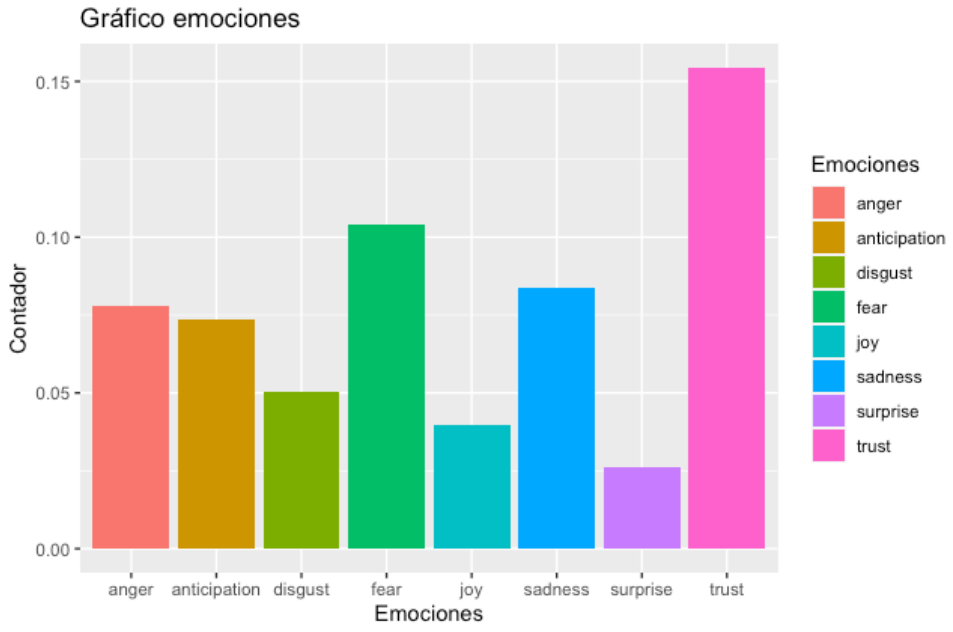

Figura 5. Gráfico general de emociones. Fuente: Elaboración propia

El gráfico anterior revela, en primer lugar, que los textos objeto de estudio presentan emociones, por lo que la abstracción total que debería de darse en un proceso totalmente objetivo, no se da. Sin embargo, los resultados son razonables, pues palabras con carga sentimental y emocional, como puede ser "asesinato", son de uso inevitable en diversas sentencias del *corpus*, al aparecer, por ejemplo, en la descripción de hechos de las mismas.

En segundo lugar, muestra que la emoción más presente es la confianza. Dado el proceso judicial con las características explicadas en puntos anteriores, no sorprende que esta sea la emoción predominante, pero a la vez, encontrarse con este valor, es una buena noticia. El hecho de que en un proceso interpretativo se tengan altos valores de confianza conlleva que existe una seguridad en la interpretación de textos legales. Para poder realizar esto último, es necesario un control sobre las emociones, que es algo bastante probable de darse a la vista del gráfico anterior. Sin embargo, emociones como la tristeza, el miedo, o el enfado también siguen estando presentes y son altas (aunque no tanto, comparativamente hablando, al ver el porcentaje de palabras totales comprendidas en ellas y lo lejos que está cualquiera de la emoción confianza). Asimismo, es importante considerar que sus valores deben ser interpretados en el mismo sentido que lo expuesto en el párrafo anterior, pues palabras concretas pueden tener cargas de emoción muy altas y, sin embargo, no afectar al juez al ser puramente descriptivas (por ejemplo, la carga emocional de "homicidio" siempre es la misma según el lexicón, pero no es lo mismo usarla como descripción en una exposición de hechos probados, que en un extracto de texto destinado a interpretar la ley para argumentar el sentido del fallo posterior). Por último, destacar que la emoción confianza no es una emoción que, aparentemente, pueda tener connotaciones negativas contra la objetividad y neutralidad por lo que significa, pero el resto sí pueden suponer un peligro contra la misma.

Por otra parte, es interesante estudiar la distribución emocional anterior sobre cada tipo de sentencias en concreto, y contrastar los resultados para cada uno de ellos. El hecho de que en la gráfica anterior se muestre que la emoción confianza es la más presente, no implica necesariamente que en cada tipo esto se de, así como que la distribución de las emociones secundarias sea la misma, pues, en primer lugar, cada tipo de sentencia es un mundo y usa un tipo de lenguaje determinado con palabras propias y muchas veces exclusivas y, en segundo lugar, la carga emocional de cada juez es individual y propia, es

decir, hay jueces que pueden dejarse llevar más por emociones y plasmar más en el papel a la hora de redactar (es decir, en este caso, en el *corpus* hay muchos jueces distintos, que realizan sentencias distintas, las cuáles versan sobre temáticas distintas). Por ello, procede un estudio más concreto sobre la distribución emocional individual de cada temática concreta. Para ello tomaré una sentencia de cada tipo (todas de igual o parecida longitud) y graficaré los resultados obtenidos en cada una.

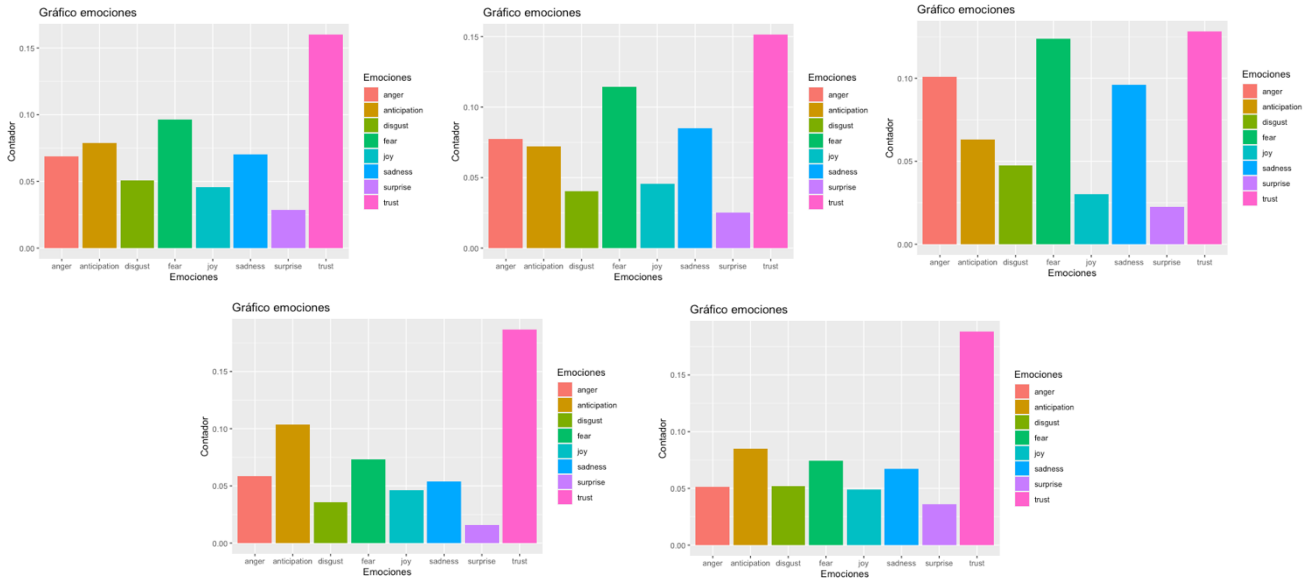

Figura 6. Análisis emocional de cada temática de sentencia. De izquierda a derecha, abuso sexual, asilo, homicidio, hipoteca y sucesión. Fuente: Elaboración propia

De los anteriores gráficos caben destacar distintas cosas. Primeramente que, en todos, la emoción más presente es la de confianza y que, dada la diferencia con respecto al resto de emociones, la de temática hipoteca y sucesión muestran los mejores resultados (el resto de emociones están menos presentes en ellas). Por otra parte, los resultados de la sentencia asilo, pero, sobre todo, de la sentencia homicidio, son preocupantes. La carga emocional de ambas en lo que respecta a emociones distintas a la de confianza es muy alta, y en ambas, tristeza y miedo se acercan a los valores que presenta ésta (de hecho, en la de homicidio, la emoción tristeza casi alcanza el valor de confianza, y la emoción enfado también sobresale). La razón puede ser la apuntada anteriormente, el uso de palabras descriptivas con carga emocional muy alta y que no influyen directamente en el juez, pero también puede ser el sesgo psicológico explicado en puntos anteriores, sobre todo viendo los resultados mostrados por la sentencia homicidio, donde pueden encontrarse valores equiparables al de confianza en otras emociones. Sin embargo, es posible que esos valores altos en emociones que no son confianza conlleven que esa carga emocional está presente

no solo en las partes descriptivas de la sentencia, sino también en las argumentativas, suponiendo un peligro para el derecho a un proceso justo.

Analizadas las emociones presentes en los datos estudiados, procedo al análisis de la polaridad del sentimiento plasmado en el texto. Para ello, y a diferencia de los gráficos anteriores, solo se tomarán los sentimientos positivos, negativos.

```
h.sent<-data.frame(t(corp_sent))
h.sent_1 <- data.frame(rowSums(h.sent[2:492]))
names(h.sent_1)[1] <- "Cantidad"
h.sent_1 <- cbind("Sentimientos" = rownames(h.sent_1), h.sent_1)
rownames(h.sent_1) <- NULL
h.sent_2<-h.sent_1[9:10,]
quickplot(Sentimientos, data=h.sent_2, weight=Cantidad, geom="bar", fill=Sentimientos, ylab="Contador")+ggtitle("Gráfico sentimientos")
```

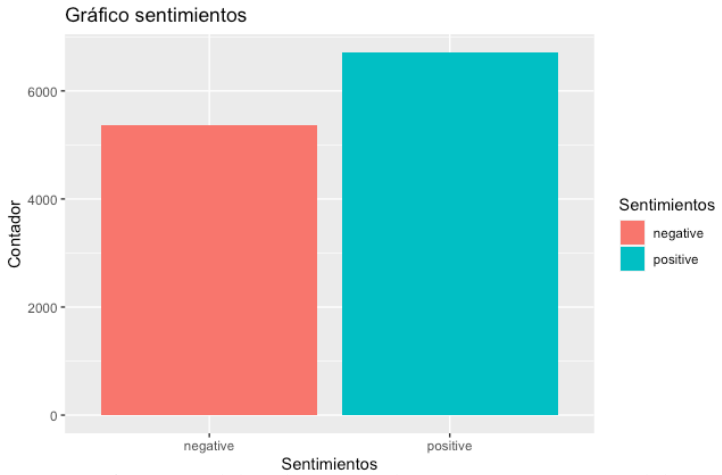

Figura 7. Gráfico general de polarización de sentimientos. Fuente: Elaboración propia

Tras analizar el gráfico, cabe destacar que, en términos agregados, es decir, analizando todo el conjunto de sentencias, el sentimiento neutro no se da. Tal y como puede apreciarse, el valor de términos positivos es mayor que el de los negativos, lo que implica que las sentencias, a nivel individual, no cumplen con la neutralidad en sentimientos que les debe ser exigida (algo razonable ya que las palabras usadas están asociadas a sentimientos en el lexicón). Sin embargo, este valor neutro no es lejano a los

resultados obtenidos, pues la diferencia entre el total de valores positivos y negativos es relativamente pequeña.

Esto último tiene una connotación clara. En el proceso de interpretación de la ley por parte de los jueces, los sesgos involuntarios e inherentes a la psicología humana que he explicado en puntos anteriores se dan, pues el sentimiento no es neutro, y hay sentimientos recogidos, pero es un sesgo "controlado" dada la cercanía de la polaridad sentimental neutra. Aun así, ese sesgo sigue suponiendo un peligro al derecho a un proceso justo.

Por último, cabe reflexionar sobre el lenguaje utilizado en el mundo del Derecho. En las sentencias se utilizan palabras como "homicidio", que muy probablemente tengan asociadas en el lexicón NRC sentimientos negativos (al igual que ocurría con las emociones). Sin embargo, el uso de estas palabras es necesario, pues la razón de su utilización es la descripción de hechos del caso concreto, y muy probablemente no impliquen directamente una emoción negativa, o al menos no tanto como podría conllevar su uso en otras fuentes de textos tales como *tweets* o textos de opinión. Por otro lado, temáticas que no conlleven términos tan negativos (asilo, por ejemplo), es probable que tengan el efecto contrario, y la neutralidad, o el sentimiento positivo sea mayor que el negativo. Al final, tal y como exponía al hablar del análisis de sentimientos, cada sentencia y cada juez son un mundo (en este caso, jueces distintos elaborando sentencias distintas, sobre temáticas distintas), por lo que es necesario proceder a un análisis de polaridad sentimental individual sobre una sentencia de cada temática (todas de similar extensión) y analizar los resultados.

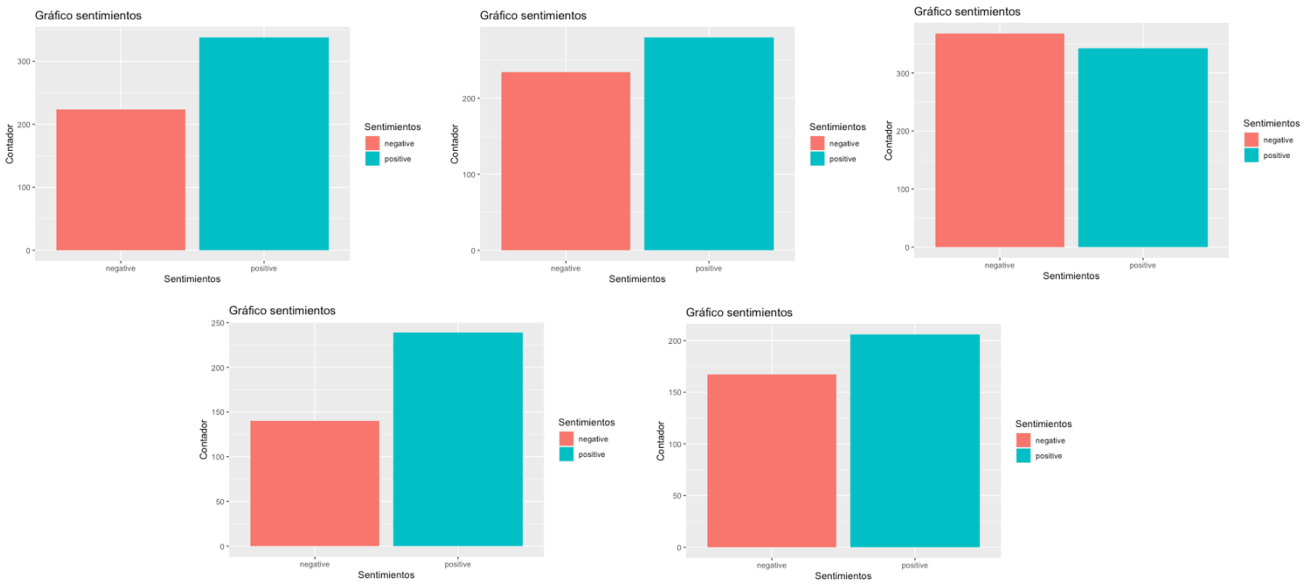

Figura 8. Análisis polaridad sentimental de cada temática de sentencia. De izquierda a derecha, abuso sexual, asilo, homicidio, hipoteca y sucesión. Fuente: Elaboración propia

Tal y como puede apreciarse, los sentimientos mayoritarios en la de homicidio son negativos, mientras que en el resto son positivos. Por ello, es probable que, aunque los jueces sí puedan verse sesgados, el uso de determinadas palabras de forma repetida en una sentencia, las cuáles se asocian, según el lexicón usado, a un sentimiento positivo o negativo, ayuden a generar la no neutralidad sentimental pese a ser ajenas a la parcialidad del juez, y no influir del todo sobre ésta (o al menos en menor medida que si apareciesen como base de la argumentación). Por ejemplo, en una sentencia de asesinato, en la descripción de hechos, es probable que aparezcan palabras como "muerte", "alevosía" o "cuchilladas". Debido a su aparición en los hechos, es probable que tan solo sean una descripción de lo ocurrido, es decir, que no implica necesariamente que el juez esté sesgado, o que generen una carga sentimental sobre el mismo, sino que deberían ser neutros en cuanto a sentimiento, al ser conceptos puramente descriptivos. Cosa distinta sería si aparecen en la parte de argumentación y fallo. Pese a esto, la neutralidad sentimental no es alcanzada por ninguna de ellas, aunque, salvo en la de hipoteca y abuso/agresión sexual, en el resto no es del todo lejana (para darse neutralidad, ambos valores, positivos y negativos, deberían estar al mismo nivel).

Por ello, cabe concluir que es importante, al analizar *Sentiment Analysis*, tener en cuenta el contexto del documento, pues no es lo mismo la connotación de ciertas palabras según qué fuente sea donde se usen, ni la parte concreta del texto donde aparezcan, aunque

a la vista de los resultados obtenidos, teniendo esto último en cuenta, no es suficiente para poder negar la posible tesis relativa al sesgo sentimental de los jueces.

4.3. Topic Modeling – LDA

4.3.1. Breve aclaración sobre el conjunto de datos

Tal y como he explicado en puntos anteriores, el primer paso a realizar en el algoritmo LDA es determinar el número de *topics* a buscar. La forma óptima es mediante algoritmos que lo lleven a cabo de forma automática y precisa. Sin embargo, en este trabajo, será elegido de antemano dado el enfoque del mismo. La razón de esto último es que en este trabajo, tal y como aparecía descrito en los objetivos del mismo, se pretende analizar cómo el LDA funciona y detecta temas, no clasificar de forma exhaustiva un cuerpo de sentencias.

Para potenciar este objetivo anterior, los datos han sido seleccionados a conciencia ya que, de las treinta sentencias elegidas, hay cinco temas concretos, cada uno de los cuáles aplica a seis sentencias en específico, y al resto en menor o nula medidas. El objetivo, por tanto, es crear el modelo LDA, generar los *topics*, y ver si esos *topics* pueden, en primer lugar, relacionarse con los cinco temas centrales sobre los que versan las sentencias analizando los términos más presentes en ellos y si, en segundo lugar, estos *topics* tienen más peso en las sentencias que, en teoría, están relacionadas con ellos y, por tanto, su asociación es efectiva. Por ello, se elegirá como número de *topics* a encontrar cinco, puesto que son sentencias con cinco temáticas diferentes.

4.3.2. Instalación de paquetes y carga de librerías

Tal y como se ha realizado en el código relativo al *Sentiment Analysis*, lo primero que se debe hacer es instalar los paquetes a utilizar y cargar sus librerías. Para este algoritmo, se han utilizado los paquetes “pdftools”, “tm”, “NLP” y “ggplot2”, ya mencionados anteriormente, junto con los siguientes:

1.- “Topicmodels”: Paquete utilizado para la implementación de funciones específicas relativas al *Topic Modeling* que incluye todo lo relativo específicamente al LDA.

2.- “Tidytext”, “Tidyr” y “Dplyr”: Paquetes que ofrecen técnicas de tratamiento de textos. Se utiliza para el cálculo de frecuencias y probabilidades finales.

Para proceder a la instalación de los mismos se utilizará la función *install.packages()*, y para cargar las librerías la función *library()*:

4.3.3. Carga de datos y creación del corpus

Utilizando el mismo código para leer los documentos que en algoritmo de *Sentiment Analysis*, se han cargado los datos y generado el *corpus*.

4.3.4. Preprocesamiento del texto: Limpieza del Corpus

Para el preprocesamiento y limpieza de textos se han seguido los mismos pasos que en el algoritmo de *Sentiment Analysis*, es decir, convertir el texto en minúsculas, eliminar las *stopwords* en castellano, eliminar los signos de puntuación, eliminar los valores numéricos, eliminar los espacios vacíos que sobran y realizar *Stemming*. Además, esta vez no solo se ha eliminado la palabra “jurisprudencia” mediante la creación de un vector, sino que también se han eliminado más términos que he considerado como *stopwords* dada la poca información que aportan y la más que probable cantidad de veces que pueden aparecer en el texto.

```
p.stopwords <- c("artículo", "art", "articulo", "derecho", "recurso", "casación", "casacion", "prueba", "motivo", "sala", "costa", "sentencia", "penal", "civil", "jurisprudencia", "jurado", "defensa", "víctima", "acusado", "tribunal", "tribun", "delito", "demanda", "infracción", "francisco", "serafina", "hecho", "proce", "fecha", "hilario", "año", "part", "fernando", "proces", "procesado","serafin")
corp <- tm_map(corp, removeWords, p.stopwords)
```

De igual forma que en el punto anterior de *Sentiment Analysis*, el vector *corp* queda convertido en lista, y el proceso de Tokenización se realiza de forma automática y en segundo plano.

4.3.5. Construcción de la matriz TDM

```
corp.tdm <- DocumentTermMatrix(corp)
```

Para poder aplicar las funciones específicas a la construcción del modelo LDA, es necesario que el objeto tenga forma de matriz TDM. Por ello, es necesario que la lista relativa al *corpus* tenga formato de matriz TDM. Para ello se utiliza la función *DocumentTermMatrix()* del paquete "tm", el cual, tan solo indicando el objeto del cual se pretende extraer la matriz, obtiene la misma. El resultado será guardado en el objeto *corp.tdm*.

4.3.6. Construcción del modelo LDA

```
corp.lda <- LDA(corp.tdm, k = 5, method = "VEM", control = list(seed = 1234))
```

Mediante la función *LDA()*, presente en el paquete "topicmodels", se genera el modelo LDA del código. En él se debe introducir el valor de k, es decir, el número de *topics* deseados, el método utilizado, en este caso, VEM, y la función de control, para hacer que los resultados sean replicables. El modelo se guarda bajo el objeto *corp.lda*.

4.3.7. Estudio de la frecuencia de cada palabra con respecto a cada topic

```
corp_topics <- tidy(corp.lda, matrix = "beta")
```

Mediante la función *tidy()*, del paquete "tidytext", se genera una matriz donde se recogerá por cada palabra, la frecuencia de ésta en cada *topic* (beta). No es una matriz que aporte a simple vista mucha información útil dada su gran tamaño, pero es necesaria para poder graficar y ver las palabras con mayor presencia en cada *topic*, lo cual se estudiará e interpretará en el próximo punto. Esta matriz se guardará en el objeto *cor_topics*.

4.3.8. Graficación de los topics y análisis de resultados

```
corp_topics_term <- corp_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)

corp_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()
```

El primer bloque de código sirve para agrupar las palabras por *topics* en función de su valor de beta. En este caso, se mostrarán las 10 palabras más comunes para cada topic (valor beta), quedando recogido bajo el objeto *corp_topics_term*. El segundo bloque de código grafica lo anterior.

Figura 9. Resultados LDA. Fuente: Elaboración propia

Puede apreciarse que se obtienen cinco *topics*, los cuáles estudiaré uno por uno y trataré de asociar a cada uno de los temas específicos de las sentencias objeto de estudio:

1.- Topic 1: Destacan términos como "atenuante", "imprudencia", "grave" o "conducta". Estos términos son ampliamente utilizados en sentencias donde el tema central es el homicidio, más específicamente, en los supuestos de homicidio con dolo o en los supuestos de homicidio de comisión por omisión.

2.- Topic 2: Destacan términos como "legítima", "donación", "testamento" o "causante" ("causant-" tras aplicar *Stemming*). Son términos ampliamente utilizados y relacionados con el concepto de sucesión y herencia.

3.- Topic 3: Destacan términos como "sexual", "caso" o "pena", términos altamente relacionados con el concepto de abuso o agresión sexual.

4.- Topic 4: Destacan términos como "asilo", "internacional" ("internacion" tras *Stemming*), "protección", "solicitud" o "persecución", términos altamente relacionados con lo relativo a protección internacional y asilo.

5.- Topic 5: Destacan términos como "cláusula", "préstamo", "consumidor" o "contrato". Son términos de los que, si bien no se puede inferir claramente el concepto de hipoteca o cláusula suelo ya que son comunes a muchos otros ámbitos del Derecho, sí que guardan relación con estos.

Tal y como se ha podido comprobar, el algoritmo LDA aplicado sobre el *corpus* de sentencias ha funcionado al identificar bien, o relativamente bien, cada una de las temáticas que había en el mismo. Por tanto, cabe afirmar que LDA es una herramienta útil para resumir sentencias, tal y como argumentaban autores citados en puntos anteriores, pues obtener los *topics* de un grupo de sentencias, teniendo conocimientos de Derecho, posibilita una reducción de tiempo en cuanto a trabajo de análisis sobre las mismas, tal y como se avanzaba al comienzo de este trabajo. Además, destacar que esta idea es, por ejemplo, la defendida por Kumar y Raghuveer (2012), los cuáles implementaron LDA con el fin de resumir sentencias en India, y concluyen que es

necesario potenciar la implementación del LDA en este tipo de documentos dados los buenos resultados obtenidos en su proceso de investigación.

4.3.9. Estudio de la probabilidad por topic por documento

```
corp_doc <- tidy(corp.lda, matrix = "gamma")
```

Una vez el modelo LDA creado, es posible la obtención de gamma, es decir, la frecuencia o peso de cada *topic* en cada documento objeto de estudio. Para obtener la matriz con estos valores, se debe utilizar de nuevo la función *tidy()*, pero esta vez, con la probabilidad gamma (recordar que en puntos anteriores se usó con la probabilidad beta con el fin de poder obtener los 10 términos con mayor importancia en cada *topic*).

Es importante recordar que la suma de los gammas de cada topic para cada documento, tal y como he explicado en puntos anteriores, tiene que resultar ser 1.

Documentos	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
AB - AG SEX 1	3,47E+00	3,47E+00	1,00E+06	3,47E+00	3,47E+00
AB - AG SEX 2	3,96E+00	3,96E+00	4,94E+05	3,96E+00	5,06E+05
AB - AG SEX 3	4,71E+00	4,71E+00	1,00E+06	4,71E+00	4,71E+00
AB - AG SEX 4	8,41E+00	8,41E+00	1,00E+06	8,41E+00	8,41E+00
AB - AG SEX 5	3,43E+00	3,43E+00	2,66E+05	3,43E+00	7,34E+05
AB - AG SEX 6	1,24E+04	6,32E+00	2,93E+05	4,37E+03	6,90E+05
Asilo 1	5,94E+01	6,09E+05	5,94E+01	3,91E+05	5,94E+01
Asilo 2	1,14E+01	1,14E+01	1,14E+01	1,00E+06	1,14E+01
Asilo 3	1,00E+01	1,00E+01	1,00E+01	1,00E+06	1,00E+01
Asilo 4	4,13E+00	4,13E+00	4,13E+00	1,00E+06	4,13E+00
Asilo 5	3,77E+00	3,77E+00	3,77E+00	1,00E+06	3,77E+00
Asilo 6	8,12E+00	8,12E+00	8,12E+00	1,00E+06	8,12E+00
Homicidio 1	1,00E+06	1,67E+00	1,67E+00	1,67E+00	1,67E+00
Homicidio 2	1,00E+06	3,08E+00	3,08E+00	3,08E+00	3,08E+00
Homicidio 3	8,36E+00	8,36E+00	1,00E+06	8,36E+00	8,36E+00
Homicidio 4	1,00E+06	6,17E+00	6,17E+00	6,17E+00	6,17E+00
Homicidio 5	3,12E+00	3,12E+00	1,00E+06	3,12E+00	3,12E+00
Homicidio 6	1,00E+06	2,58E+00	2,58E+00	2,58E+00	2,58E+00
HP-CS 1	9,61E+00	9,61E+00	9,61E+00	9,61E+00	1,00E+06
HP-CS 2	9,24E+00	9,24E+00	9,24E+00	9,24E+00	1,00E+06
HP-CS 3	9,35E+00	9,35E+00	9,35E+00	9,35E+00	1,00E+06

HP-CS 4	9,57E+00	9,57E+00	9,57E+00	9,57E+00	1,00E+06
HP-CS 5	7,89E+00	7,89E+00	7,89E+00	7,89E+00	1,00E+06
HP-CS 6	4,98E+00	4,98E+00	3,36E+05	4,98E+00	6,64E+05
Sucesión 1	5,34E+00	1,00E+06	5,34E+00	5,34E+00	5,34E+00
Sucesión 2	6,74E+00	1,00E+06	6,74E+00	6,74E+00	6,74E+00
Sucesión 3	6,68E+00	1,00E+06	6,68E+00	6,68E+00	6,68E+00
Sucesión 4	1,26E+01	1,00E+06	1,26E+01	1,26E+01	1,26E+01
Sucesión 5	4,46E+00	1,00E+06	4,46E+00	4,46E+00	4,46E+00
Sucesión 6	6,95E+00	1,00E+06	6,95E+00	6,95E+00	6,95E+00

Figura 10. Matriz valores gamma de cada topic en cada documento. Fuente: Elaboración propia

Tras generar la matriz, puede observarse que, en general, los pesos mayores para el *topic* 1 se dan, efectivamente, para las sentencias con temática de homicidios, que los del *topic* 2 se dan para las sentencias con temática sucesión, que los del *topic* 3 se dan para las sentencias con temática de abuso y agresión sexual, que los del *topic* 4 se dan para sentencias con temática asilo, y que los del *topic* 5 se dan para sentencias con temática hipoteca – cláusula suelo.

5. CONCLUSIONES

Expuesta y analizada la teoría y literatura, y examinados los resultados de la parte práctica a lo largo de los puntos anteriores del presente trabajo, cabe concluir que:

1º.- El análisis de documentos de carácter jurídico con técnicas de *Text Mining* es algo presente en la sociedad y que cada vez se va potenciando más. La relación entre el Derecho y el *Big Data* y *Data Mining*, y sus técnicas, está en el día a día ya no solo en la implementación de algoritmos concretos, sino también en el ámbito de la protección de datos.

2º.- De entre las formas en las que se puede utilizar el LDA o el *Sentiment Analysis*, el uso del lexicón es la más generalizada dada su sencillez. Sin embargo, parece necesario profundizar en los mismos, dada su limitación en idiomas que no sean en inglés. Tan solo el lexicón NRC, y pocas excepciones más, se abre a otros idiomas, y la forma mediante la que se abren es poco fiable.

3º.- Los sesgos psicológicos involuntarios presentes en el ser humano, y que aparecen especialmente cuando se toman decisiones o se interpretan situaciones concretas, también se dan en el proceso de interpretación y aplicación de la ley por parte de los jueces. Los resultados obtenidos en el *Sentiment Analysis* desarrollado en este trabajo así lo indican, aunque también muestran que no es algo muy grave dadas las emociones que prevalecen (confianza) y que el sentimiento neutro, en términos de polaridad, está cerca en los gráficos agregados. Es importante considerar, además, que el contexto del documento analizado, así como la parte concreta del mismo, influye en lo relativo a sentimientos. Tal y como se ha analizado en puntos anteriores, no es lo mismo usar la palabra “homicidio” en una lista de hechos probados, donde la influencia emocional sobre el juez es ínfima o nula, que utilizarlo en la parte argumentativa o del fallo, fuera del ámbito del relato de hechos probados.

Sin embargo, y considerando todo lo anterior, todo lo que no sea neutralidad en sentimientos, por leve que sea la polaridad, puede poner en peligro el derecho a un proceso justo.

4º.- El contexto en el que se usen las palabras debe ser tenido en cuenta a la hora de realizar una aplicación de *Sentiment Analysis*, pues como se demostraba en el bloque de aplicación práctica de los algoritmos, hay términos asociados a sentimientos positivos o negativos en cualquier lexicón que, dependiendo del texto, pueden no tener una connotación sentimental del mismo carácter, al ser utilizados como meros términos descriptivos.

5º.- El algoritmo LDA es muy eficaz en cuanto a clasificación de términos en *topics* concretos, y su aplicación práctica más evidente y eficaz es la de realizar resúmenes de textos, con las ventajas que ello conlleva y que han sido expuestas en puntos anteriores.

6º.- El algoritmo LDA, dados los buenos resultados para identificar *topics* en sentencias, es aplicable al mundo del Derecho no solo con el fin de resumir documentos.

7º.- Tal y como demuestra la literatura citada en este trabajo (y comentada en el punto 3.4.2), la clave para potenciar un motor de búsqueda es una mejora en la clasificación de los documentos que contenga. El proceso de clasificación se hace obteniendo *topics* mediante *Topic Modeling* y entrenándolos en un set pequeño, con el fin de sacar patrones comunes que aplicar a cantidades de datos más grandes posteriormente. Tal y como se ha demostrado con lo presentado en la parte práctica de este trabajo, el LDA ha obtenido buenos resultados en cuanto a extracción de *topics*, por lo que una potencial implementación de este algoritmo en los procesos de clasificación de documentos jurídicos puede llevar a una mejora en los motores de búsqueda de este tipo de documentos.

A raíz de las conclusiones extraídas, cabe terminar diciendo que este ámbito de aplicación del *Text* Mining, el Derecho, no es un ámbito estanco o cerrado, sino que puede seguir evolucionando. Viendo los resultados del algoritmo de *Sentiment Analysis*, cabría estudiar a futuro el impacto tangible de esa influencia psicológica sobre el derecho a un proceso judicial justo, y estudiar las formas de paliar el sesgo existente en caso de ser grave.

Por otro lado, en este trabajo se ha probado la eficacia del algoritmo LDA en estos documentos a nivel de obtención de *topics* y resúmenes, pero una línea de investigación

a futuro interesante sería el estudio de cuál es la mejor forma de implementarlo en un proceso de clasificación de documentos, es decir, el algoritmo más eficaz con el que podría funcionar mejor, así como estudiar la viabilidad de implementarlo concretamente en motores de búsqueda jurídicos. Según lo expuesto, muchos autores defienden que la mejora en la clasificación implica la mejora en los motores de búsqueda, aunque la implementación práctica de los algoritmos es algo distinto, al entrar en juego variables como cantidad o tipos de documentos, y debería ser objeto de estudio en el futuro.

6. REFERENCIAS BIBLIOGRÁFICAS

BLEI, D.M., GRIFFITHS, T.L., JORDAN, M.I. (2010). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57 (2), 1-30.

BLEI, D.M., LAFFERTY, J.D. (2006). Dynamic topic models. *Proceedings of the 23rd International Conference on Machine Learning*, junio 2006, 113-120.

BLEI, D.M., NG, A.Y., JORDAN, M.I. (2003). Latent Direchlet Allocation. *Journal of Machine Learning Research*, nº3, 993-1022.

CHUNG, H.M., GRAY, P. (2015). Special Section: Data Mining. *Journal of Management Information Systems*, 16 (1), 11-16.

CONSEJO GENERAL DEL PODER JUDICIAL. (15 de abril de 2021). *Buscador CENDOJ,* Nombre de la página: https://www.poderjudicial.es/search/indexAN.jsp

CRUZ CASTILLO, A.L. (2012). La razón de las emociones. Formación Social, política y cultural de las emociones. *Eléuthera*, 6, 64-82.

D'ANDREA, A., FERRI, F., GRIFONI, P., GUZZO, T. (2015). Approaches, tools and applications for sentiment analysis implementatios. *Internation Journal of Computer Applications*, 125 (3), 26-33.

DEERWESTER, S., DUMAIS, S.T., FURNAS, G.W. Y LANDAUER, T.K. (1990). Indexing by Latent Semantic Analysis. *Journal of the American society for information science*, 41 (6), 391-407.

EL NAQA, I., MURPHY, M.J. (2015). Machine Learning in Radiation Oncology. Nueva York: Springer.

FEI-FEI, L., PERONA, P. (2005). A bayesian hierarchical model for learning natural scene categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 524-531.

GOMEZ, H.F., BENITEX, J., GUAMAN, F.O., JÁCOME LAGARZA, L.R. (2016). Semantic Analysis of Judicial Sentences base don Text Polarity. *Iberian Conference on Information Systemsand Technologies*, 2016, 1-4.

HEARST, M. (2003). What is Text Mining? Berkely: University of California.

HERNÁNDEZ BLANCO, A. (2018). El papel de los sentimientos y las emociones en la toma de decisiones dentro de las organizaciones. *Revista Neumann Business School*, 4 (1), 57-77.

HOFFMAN, T. (1999). Probabilistic Latent Semantic Analysis. Berkeley: University of California.

KEITH NORAMBUENA, B., FUENTES LETTURA, E., MENESES VILLEGAS, C. (2019). Sentiment analysis and opinion mining applied to scientific paper reviews. *Inteligent Data Analysis*, nº 23, 191-214.

KUMAR, R., RAGHUVEER, K. (2012). Legal Document Summarization using Latent Dirichlet Allocation. *International Journal of Computer Science and Telecommunications*, 3(7), 8-23.

LI, L.J., WANG, C., LIM, Y., BLEI, D.M., FEI-FEI, L. (2010). Building and using a semantical image hierarchy. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, junio 2010, 3336-3343.

LIDDY, E.D. (2001). Natural Processing Language. Siracusa: Universidad de Siracusa.

MOHAMMAD, S.M. (2020). Practical and ethical considerations in the effective use of emotion and sentiment lexicons. Ottawa: National Research Council of Canada.

MOHAMMAD, S.M. (2010). *NRC word emotion lexicon*. Nombre de la página: https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm.

MUÑOZ ARANGUREN, A. (2011). La influencia de los sesgos cognitivos en las decisiones jurisdiccionales: El factor humano. Una aproximación. *Indret*, 2, 1-39.

PANESSI, W., BORDIGNON, F.R. (2001). Procesamiento de variables morfológicas en búsquedas de textos en castellano. *Revista Interamericana de Bibliotecología,* 24 (1), 69-88.

PLISSON, J., LAVRAC, N., MLADENIC, D. (2004). A rule based approach to word lemmatization. *Proceedings of IS*, 3, 83-86.

PLUTCHIK, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984, 197-219.

PUYOL MORENO, J. (2014). Una aproximación a Big Data. *Revista de Derecho UNED*, nº14, 471-505.

RHODY, L.M. (2012). Topic Modeling and Figurative Language. Nueva York: CUNY Graduate Center.

RICHTER, M. (2015). La protección de datos de carácter personal como derecho humano. *Revista Auctoritas Prudentium,* nº 12, 18-29.

SARIOGLU, E., YADAV, K., CHOI, H.A. (2013). Topic Modeling Based Classiffication of Clinical Reports. *51st Annual Meeting of the Association for*

Computational Linguistics Proceedings of the Student Research Workshop, 67-73.

SHARMA, A.K. Y GREAT LEARNING. (16 de octubre de 2020). *Understanding Latent Dirichlet Allocation (LDA).* Nombre de la página: https://www.mygreatlearning.com/blog/understanding-latent-dirichlet-allocation/

SHMUELI, G., BRUCE, P.C., YAHAV, I., PATEL, N.R., LICHTENDAHL, K.C. (2018). Data Mining for Business Analytics. Concepts, Techniques and Applications in R. Hoboken: Wiley.

SILGE, J., ROBINSON, D. (6 de abril de 2021). *Text Mining with R. A tidy approach.* Nombre de la página: https://www.tidytextmining.com/index.html.

STC 164/2008, de 15 de diciembre (BOE núm. 8, de 9 de enero de 2009).

WALLACH, H.M. (2006). Topic Modeling: Beyond the bag of words. *Proceedings of the 23rd International Conference on Machine Learning*, junio 2006, 977-984.

ZHU, Z. (2011). Improving search engines via Classification. London: University of London.

7. ANEXOS

Anexo I: Código completo utilizado en RStudio relativo al algoritmo de *Sentiment Analysis*

```
#instalación de paquetes
install.packages("pdftools")
install.packages("tm")
install.packages("NLP")
install.packages("syuzhet")
install.packages("ggplot2")

# carga de librerías
library(pdftools)
library(tm)
library(NLP)
library(syuzhet)
library(ggplot2)
library(stringr)

# lectura de documentos
documentos <- list.files(pattern = "pdf$")

#creación del corpus
corp <- Corpus(URISource(documentos),
               readerControl = list(reader = readPDF))

#listado de stopwords en español
stopwords(kind='es')

#limpieza de corpus
corp <- tm_map(corp, content_transformer(tolower))
corp <- tm_map(corp, removeWords, stopwords("spanish"))
corp <- tm_map(corp, removeWords, stopwords(kind = "es"))
```

```
p.stopwords <- c("jurisprudencia")
corp <- tm_map(corp, removeWords, p.stopwords)
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, removeNumbers)
corp <- tm_map(corp, stripWhitespace)
corp <- tm_map(corp, stemDocument)

#transformación de lista a vectores
corp_ch <- unlist(corp)

#asociación de documentos a sentimientos
corp_sent <- get_nrc_sentiment(corp_ch, cl = NULL, language = "spanish", lowercase = TRUE)

#graficación de los datos
h.sent<-data.frame(t(corp_sent))
h.sent_1 <- data.frame(rowSums(h.sent[2:18]))
names(h.sent_1)[1] <- "Cantidad"
h.sent_1 <- cbind("Emociones" = rownames(h.sent_1), h.sent_1)
rownames(h.sent_1) <- NULL
h.sent_2<-h.sent_1[1:8,]
quickplot(Emociones, data=h.sent_2, weight=Cantidad, geom="bar", fill=Emociones, ylab="Contador")+ggtitle("Gráfico emociones")

h.sent<-data.frame(t(corp_sent))
h.sent_1 <- data.frame(rowSums(h.sent[2:18]))
names(h.sent_1)[1] <- "Cantidad"
h.sent_1 <- cbind("Sentimientos" = rownames(h.sent_1), h.sent_1)
rownames(h.sent_1) <- NULL
h.sent_2<-h.sent_1[9:10,]
quickplot(Sentimientos, data=h.sent_2, weight=Cantidad, geom="bar", fill=Sentimientos, ylab="Contador")+ggtitle("Gráfico sentimientos")
```

Anexo II: Código completo utilizado en RStudio relativo al algoritmo de *LDA*

```
#instalación de paquetes
install.packages("pdftools")
install.packages("topicmodels")
install.packages("tm")
install.packages("NLP")
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")

# carga de librerías
library(pdftools)
library(topicmodels)
library(tm)
library(NLP)
library(ggplot2)
library(dplyr)
library(tidyr)
library(tidytext)

# lectura de documentos
documentos <- list.files(pattern = "pdf$")

#creación del corpus
corp <- Corpus(URISource(documentos),
               readerControl = list(reader = readPDF))

#listado de stopwords en español
stopwords(kind='es')

#limpieza de corpus
corp <- tm_map(corp, content_transformer(tolower))
corp <- tm_map(corp, removeWords, stopwords("spanish"))
```

```
corp <- tm_map(corp, removeWords, stopwords(kind = "es"))
p.stopwords <- c("artículo", "art", "articulo", "derecho", "recurso", "casación", "casacion", "prueba", "motivo", "sala", "costa", "sentencia", "penal", "civil", "jurisprudencia", "jurado", "defensa", "víctima", "acusado", "tribunal", "tribun", "delito", "demanda", "infracción", "francisco", "serafina", "hecho", "proce", "fecha", "hilario", "año", "part", "fernando", "proces", "procesado","serafin")
corp <- tm_map(corp, removeWords, p.stopwords)
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, removeNumbers)
corp <- tm_map(corp, stripWhitespace)
corp <- tm_map(corp, stemDocument)

#construcción de la matriz documentos-términos
corp.tdm <- DocumentTermMatrix(corp)
inspect(corp.tdm)

#construcción del modelo LDA
corp.lda <- LDA(corp.tdm, k = 5, method = "VEM", control = list(seed = 1234))
corp.lda

#estudio de los topics con la probabilidad por topic por palabra (beta), es decir, de cada palabra con respecto a cada topic
corp_topics <- tidy(corp.lda, matrix = "beta")
corp_topics

#graficación de lo anterior
corp_topics_term <- corp_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)

corp_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()

#estudio de los topics con la probabilidad por topic por documento (gamma)
corp_doc <- tidy(corp.lda, matrix = "gamma")
corp_doc

pesos.gamma = colSums (select (corp_doc, contains ("gamma")))
pesos.gamma
```