ESTUDIO DE IDENTIFICACIÓN DE AUTORÍA

Ciencias de la Computación e Inteligencia Artificial

12 DE MARZO DE 2019

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Tutores: Luis de Campos Ibáñez y Juan F. Miguel Huete

Índice

1. INTRODUCCIÓN 3

1.1 Propósito 3

1.2 Partes 4

1.3 Justificación 4

1.4 Ámbito 5

1.5 Relevancia teórica y práctica de la investigación 6

1.6 Estado del arte 6

1.7 Breve descripción de la metodología de la investigación 7

1.8 Esquema 7

2 PROFUNDIZAR SOBRE LA AUTORÍA 8

3 OBJETIVO 8

4 DATOS 9

5 FUNDAMENTOS TEÓRICOS 13

5.1 Extracción de características 13
- **5.1.1 Bolsa de Palabras** 13
- **5.1.2 Modelos *N-gram*** 14
- **5.1.3 Vectorización de documentos** 15
- **5.1.4 TF-IDF** 16

5.2 Selección de características 17
- **5.2.1 ANOVA** 18
- **5.2.2 Kendall's** 18

5.3 Reducción de la dimensión 19
- **5.3.1 LSA** 19
- **5.3.2 NMF** 20

5.4 Modelos 20
- **5.4.1 Red Neuronal** 20
  - **5.4.1.1 Densa** 23
  - **5.4.1.2 Dropout** 23
  - **5.4.1.3 Pooling** 23
  - **5.4.1.4 Embedding** 23
  - **5.4.1.5 Convolucionales 1D** 23

5.4.1.6 LSTM 23
5.4.1.7 GRU 23
5.4.2 Máquina de Vectores Soporte 23

6 MÉTRICAS 25

6.1 Exactitud o *accuracy* **25**

6.2 Precisión **26**

6.3 Exhaustividad o *recall* **26**

6.4 Validación cruzada **26**

7 EXPERIMENTOS 26

7.1 Modelo TF-IDF ANOVA MLP **26**
7.1.1 Parámetros 30
7.1.2 Resultados 30

7.2 Modelo TF-IDF ANOVA SVM Lineal **30**
7.2.1 Parámetros 31
7.2.2 Resultados 31

7.3 Modelo TF-IDF LSA MLP **31**
7.3.1 Parámetros 31
7.3.2 Resultados 31

7.4 Modelo TF-IDF NMF MLP **31**
7.4.1 Parámetros 31
7.4.2 Resultados 32

7.5 Modelo Embeddings LSTM **32**
7.5.1 Parámetros 32
7.5.2 Resultados 32

7.6 Modelo Embeddings sepCNN **32**
7.6.1 Parámetros 32
7.6.2 Resultados 32

8 COMPARATIVA 32

9 TECNOLOGÍA 32

10 ESTUDIO 33

11 CONCLUSIÓN 33

12 BIBLIOGRAFÍA 33

1. Introducción

1.1 Propósito

Vivimos un contexto tecnológico sin precedentes conocido como la era de la **información**. Cada vez somos más conscientes del poder que nos proporciona adquirirla y usarla. Por contraposición, al mismo tiempo vivimos una realidad en la que se nos hace más difícil responder de forma eficiente al volumen de datos que se crea. Cabe destacar que hay estudios que afirman que sólo se almacena menos de un 0,4% de la información que se produce y de esta, más del 75% se encuentra desestructurada. Llamamos desestructura a aquella información que no puede ser procesada directamente por modelo, como documentos de texto o imágenes. Es frecuente que se nos presenten los datos cada vez más complejos e interconectados, incluso requiriendo de un preprocesamiento no trivial y en última instancia datos que deben ser inferidos a partir de otros.

Es en este último punto donde el Aprendizaje Automático, más conocido en inglés como *Machine Learning* adquiere un papel clave al enlazarse y apoyarse con otras áreas del conocimiento. Nos permite responder a preguntas que o, por un lado, requerían de la supervisión de un humano o ni siquiera se sabía una respuesta.

Es indiscutible que muchos modelos de *Machine Learning* no tienen competidores actualmente que se asemejen en términos de eficiencia y eficacia. Algunos de los ejemplos más conocidos de aplicación son los siguientes:

- Diagnósticos médicos
- Procesamiento del lenguaje natural o *Natural Language Processing (NPL)*.
- Búsqueda online
- Coches inteligentes
- …

A medida que ha pasado el tiempo, la lista de aplicaciones se ha hecho interminable al mismo tiempo que la lista de publicaciones científicas o *papers* relacionadas con el tema ha crecido considerablemente. Este fenómeno, principalmente, se debe a dos motivos:

- Existe un componente **económico** que ha decido apostar fuertemente por una industria relacionada con modelos predictivos. La necesidad de automatizar trabajos y explotar la información que se posee siempre ha tenido un rol fundamental en una empresa. En 2016 solo el 8,6% de las empresas realizaba análisis masivos de datos, actualmente es un aspecto diferencial en la industria.
- El acceso a la **tecnología** para implementar, entrenar y validar modelos está prácticamente al alcance de todo el primer mundo. Esto se debe a la apuesta por igual que se ha hecho por la nube o más conocida como la *cloud*, que permite

no disponer en local de los recursos en *hardware* necesarios para realizar estas tareas. Existe también un componente de desarrollo *software*, que ha posibilitado esta situación, dejando atrás la barrera técnica que encontraban muchos investigadores para testear sus soluciones.

Nos encontramos en una etapa dorada para la aplicación de todos estos conceptos que se han venido desarrollando en el último siglo de forma teórica y que empiezan a ver sus primeros frutos en el presente.

Existen infinidad de formas en las que nos podemos encontrar la información. Es sabido, que la dificultad de la predicción o respuesta a la pregunta que plantee el problema estará sumamente relacionada con el perfil de información que poseamos. Pues aun siendo la misma pregunta, el problema es distinto si el conjunto de características viene dato en forma de imágenes o textos. Para cada una de estas posibilidades se proponen diferentes estrategias que han conformado subramas dentro del Aprendizaje Automático.

En el caso que nos atañe en este trabajo, la información está representada por texto etiquetado por un autor. De las muestras disponibles que hay para un autor se debe inferir las características que lo definen. De este modo, llegado un texto nuevo el modelo predictivo debe discernir en base a las características extraídas a que autor pertenece. Este problema es conocido como identificación de la autoría o en inglés *authorship attribution problem*.

A lo largo de la historia se han producido numerosos debates sobre la autoría de obras transcendentales para el conocimiento humano. El hecho de conocer el autor de un contenido da un peso conceptual extra a sus palabras que, apoyado por su biografía y su circunstancia crean un marco decisivo desde el que poder abarcar cualquier estudio. Esto es debido a que a veces podría ser más conclusivo responder a la pregunta, quién desarrolló un contenido, que el contenido mismo.

El propósito de este trabajo es automatizar la identificación del autor de un documento sobre un conjunto de autores previamente definido. Se desarrollará y argumentará un estudio completo sobre que es diferencial y que no para etiquetar el autor de una obra textual.

1.2 Partes

El trabajo constará de 12 secciones definidas en el índice inicial y fácilmente consultables. El formato electrónico permite navegabilidad sobre el mismo gracias a la inserción de vínculos sobre las entradas. A lo largo del trabajo se pueden encontrar referencias en forma de vínculos al apartado bibliográfico que se encuentra en la última sección de este documento.

1.3 Justificación

He decido abordar este trabajo tras la lectura de una Encuesta de Métodos Modernos de Atribución de Autoría (*A Survey of Modern Authorship Atribution Methods*, Universidad de Aegean) y el efecto del tamaño del conjunto de autores y el tamaño de

los datos en la atribución de autoría (*The effect of autor set size and data size in Authorship Attribution*). Ambas investigaciones fueron sugeridas por los tutores del proyecto. La lectura del capítulo V y VI del libro Inteligencia Artificial un Enfoque Moderno (Artificial Intelligence A Modern Approach por Stuart Russell y Peter Norvig) Ha supuesto una notable aportación en este trabajo.

También existe un componente previo personal que me ha llevado a aceptar esta temática. Mi afición por la lectura ha ocupado gran parte de mi vida, haciendo un especial énfasis en las obras filosóficas. Esto unido a mi interés por la psicología humana me ha permitido incorporar algunas pincelas que considero interesantes en el desarrollo del trabajo.

Desde un punto de vista laboral, mi trabajo actual como Científico de Datos para la Prevención del Fraude y el Crimen Organizado en Deloitte comparte muchas áreas de conocimiento con el tema que nos ocupa. Trato desde la identificación de nombres sobre listas sancionadas (Watch List Filtering) haciendo uso de emparejamiento por lógica difusa, hasta la identificación y clasificación de alertas sospechosas en los conceptos de las transferencias bancarias. Como ya se ha mencionado antes, las aplicaciones de *Machine Learning* inundan el mercado actual, danto la posibilidad de mantener arquitecturas de modelos semejantes en problemas aparentemente distintos. *ML* ha dado un paso de gigante en lo que a la abstracción de problemas se refiere.

Desde el punto de vista de las ciencias de la computación y la inteligencia artificial rama del conocimiento que estudio, este problema hace uso de muchos de los conceptos y herramientas que en ellas se explican. Debido a que ha sido una decisión propia el estudio de esta especialidad, es razonable la elección del proyecto.

1.4 Ámbito

El problema de la Identificación de Autoría o la Autoría de Documentos es una cuestión interdisciplinar que comparte actualmente lingüística, Recuperación de Información o *Information Retrieval* e Inteligencia Artificial. Siendo estos dos últimos desde los que se va abordar el trabajo. Nos adentrándonos en la subrama Aprendizaje Automático perteneciente a la Inteligencia Artificial.

Podemos decir que los tres ámbitos específicos que predominan en este trabajo son el procesamiento del lenguaje natural, la recuperación de información y el aprendizaje automático. A continuación, daremos una definición básica de ambos conceptos:

El procesamiento del lenguaje natural (NLP) es un rango teóricamente motivado de técnicas computacionales para analizar y representar textos que ocurren naturalmente en uno o más niveles de análisis lingüístico con el fin de lograr el procesamiento del lenguaje humano en una variedad de tareas o aplicaciones.

La Recuperación de Información es la conversión de grandes volúmenes de texto en estructuras simplificadas y comprensibles para su uso posterior.

El Aprendizaje Automático es el estudio de algoritmos de computación que mejoran automáticamente por medio de la experiencia.

Si quisiéramos profundizar más, existen numerosas subcategorías dentro del problema al que nos enfrentamos, dependiendo del registro del lenguaje, permisibilidad de faltas

de ortografía, tamaño de los textos, número de autores, número de muestras… Más adelante concretaremos estos conceptos en el apartado 2.

1.5 Relevancia teórica y práctica de la investigación

[Resumen del artículo de la encuesta y algo más]

Relevancia teórica

{Explicar los posibles avances dentro del campo}

A continuación, se presentan algunas aplicaciones que tiene el problema:

- Desde la literatura nos encontramos con debates actuales sobre la identificación de autores de obras anónimas o puesta en duda de obras que fueron atribuidas sin estudio previo. Pero la realidad es que muchas obras continúan anónimas actualmente ya sea por deterioro de la misma o falta de candidatos.
    - Un ejemplo de la primera es la conocida obra del Lazarillo de Tormes que tras una lista de candidatos se resolvió su autoría gracias a un estudio lingüista reconociendo a Sebastián de Horozco como escritor.
    - Un ejemplo de la segunda es el debate que hubo en torno a las obras de Shakespeare debido a su estilo impropio de la cuna del autor y a las lagunas de su biografía.

- Desde la criminología, derecho, psicología y psiquiatría nos encontramos considerable número de problemas.
    - Peritaje de conversaciones electrónicas.
    - Autoría de cartas de suicidio.
    - Falsificación en las relaciones laborales.
    - Falsificación de estudios (TFGs y TFMs).
    - Identificación de trastornos.
    - Psicología evolutiva y del aprendizaje.

- Desde la automatización de tareas, como es nuestro caso, tenemos algunos ejemplos.
    - Documentación automática (nuestro caso)
    - Detección de suplantaciones de identidad

1.6 Estado del arte

El primer estudio que se realizó sobre la materia fue en 1887 sobre las obras de Shakespeare, publicado por Thomas Corwin Mendenhall. Seguido medio siglo después por los trabajos estadísticos de Tule (1938:1944) and Zipf (1932). Aunque sin duda el trabajo más notorio y reconocido es el estudio realizado por Mosteller and Wallace

(1964) construido a partir de 'The Federalist Papers', un conjunto de 146 documentos de longitud variada escritos por tres autores diferentes. El método usado en este último estudio fue un modelo Bayesiano estadístico centrado sobre un grupo de palabras comunes en inglés.

Antes de este estudio, la capacidad de diferenciar autores se veía desde un punto de vista lingüístico dependiente del estilo literario de cada uno. Se definió un conjunto heterogéneo de mediciones, aproximadamente llegaron a ser unas 1000, sobre propiedades concretas y triviales del uso del lenguaje. Algunas de éstas fueron la frecuencia de palabras por frase, la frecuencia de caracteres por frase o el uso de palabras poco frecuentes.

La metodología de trabajo estaba bastante limitada tanto por los medios disponibles en el momento como por los problemas que se planteaban. La mayor parte de estos problemas cumplían las siguientes características:

- El documento a analizar usualmente se trataba de una obra completa o libro.
- El número de autores sobre los que se realizaba el estudio era pequeño (aproximadamente 2 o 3 autores).
- Existía un alto componente subjetivo en la evaluación de los métodos propuestos.
- La decisión de que método era el más apropiado para un problema partía de una ausencia notable de un banco de problemas resueltos.

A partir de la revolución de Internet, la metodología tradicional aplicada sobre los nuevos problemas se quedó obsoleta debido a la diversificación y volumetría de estos. Es en este punto, cuando el procesamiento del lenguaje natural (natural language processing) en conjunto con el Aprendizaje Automático y Recuperación de Información se imponen como áreas del conocimiento y modelos de trabajo para la identificación de la autoría.

Desde la recuperación de información se desarrollaron técnicas eficientes para la representación y clasificación de grandes volúmenes de información.

Desde el aprendizaje automático con el desarrollo de algoritmo capaces de trabajar con problemas de dimensión. En 1992, con la publicación realizada por E. Boser, Isabelle M. Guyon y Vladimir N. Vapnik en la que sugirieron la aplicación del modelo 'máquina de vectores soporte' (SVM) sobre 'kernels' no lineales.

Desde el procesamiento del lenguaje natural con el desarrollo de herramientas eficientes que analizaran características del lenguaje.

1.7 Breve descripción de la metodología de la investigación

1.8 Esquema

Que es el problema de la Autoría de Documentos (Autorships).

La iniciativa parlamentaria permite a un diputado o a un senador presentar al parlamento un proyecto de artículo constitucional, de ley o de decreto. Este proyecto

puede ser redactado de modo completo o formulado en términos generales. La comisión de la cámara donde ha sido depositada la iniciativa decide si es necesario darle curso. Por ejemplo, una iniciativa sobre un tema que ya está en discusión en el parlamento no será declarada válida. Si la comisión considera que la iniciativa puede ser acogida, el proyecto sigue el itinerario legislativo clásico (examen por parte de la comisión de la otra cámara, procedimiento de consulta, cámaras del parlamento, etc.)

Desde 1978 se recogen las iniciativas parlamentarias, pero nosotros nos centraremos 2008.

Fuente de las iniciativas parlamentarias:

Lenguaje parlamentario, corpus

Lenguaje oral cuidado

Lenguaje político de controversia, trufado por citas generales (con caracteres atractivos) función de alusión al otro. Estructura: argumentación, muy estructurado.

Extracción de características:

1. Características generales
2. Características lingüísticas
3. Características temáticas
4. Características a nivel de palabras

Explicar matriz dispersa

Explicar función logarítmica para la

2 Profundizar sobre la autoría

3 Objetivo

Estos son los objetivos del proyecto:

1. Realizar un **preprocesamiento** correcto de los **datos** para que puedan ser utilizados como entrada en un modelo de aprendizaje automático.
2. Proponer diversos **modelos** entrenados de clasificación multietiqueta que presenten buenos resultados.
3. Validar los modelos y compararlos entre sí.
4. Definir un criterio justificado para elegir el mejor modelo y presentarlo en un ambiente de producción.

4 Datos

La fuente de datos utilizada son la Iniciativas Parlamentarias del Parlamento de Andalucía durante el año 2008. Estas iniciativas están recogidas en 5.260 ficheros en formato XML y codificación UTF8. El formato XML presenta la siguiente estructura:

```
< iniciativa_completa >
        ...
        < iniciativa >
                ...
                < intervencion >
                        < interviniente > Nombre </interviniente >
                        < discurso >
                                < parrafo > Texto </parrafo >
                                ...
                        </discurso >
                </intervencion >
        </iniciativa >
</iniciativa_completa >
```

Donde el nombre se corresponde con la columna de las etiquetas y el texto formado a partir de la unión de los párrafos con la columna de las características.

La extracción del nombre requiere de una complejidad mayor. Esto es debido a que no es exactamente el nombre como tal, sino la presentación que se le realiza al interviniente. Alguna de las dificultades para la extracción son las siguientes:

- El nombre viene acompañado del cargo político o puesto laboral que ocupa el interviniente.
- Sólo se aporta un apellido o un nombre en lugar del nombre completo del interviniente.
- Sólo se alude al cargo del interviniente. Por ejemplo: “el diputado”.
- Frases del manejo del debate. Por ejemplo: “toma la palabra”.
- Diferencia entre nombres en mayúsculas o minúsculas.

Algunas de estas dificultades se pueden resolver de forma automática, como es el caso de las mayúsculas y minúsculas, otras se han tenido que resolver manualmente con un proceso iterativo de visualización de los datos. Para ello, se ha definido un fichero de remplazamiento de cadenas de caracteres que se encuentra en la raíz del proyecto, llamándose “replace.txt”. El fichero tiene la siguiente estructura:

```
cadena1
cadena2
cadena3
...
```

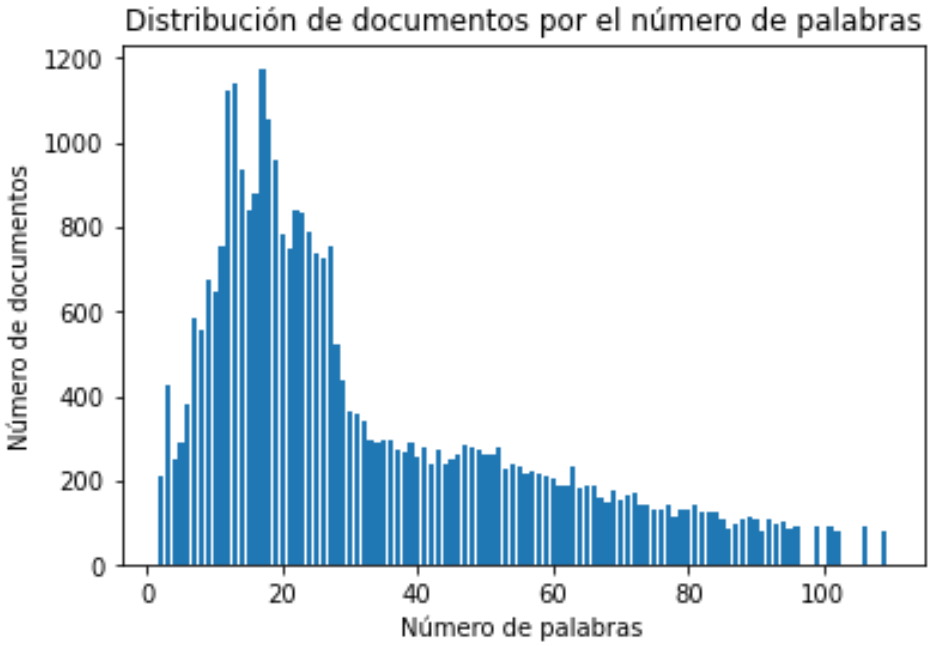
Donde cada una de las cadenas que se lee del fichero se remplaza por la cadena vacía. En conclusión, una etiqueta de conjunto de datos seguirá la siguiente expresión:

$$\forall i \in (0, N)\ nombre.replace(cad_i,)$$

Donde N es el número de cadenas contenidas en el fichero de remplazamiento que actualmente son 343.

Tras realizar la lectura de todos los ficheros y aplicar la limpieza sobre el campo nombre extraemos los siguientes datos: tenemos un total de 596 nombres o etiquetas diferentes con un total de 58.338 documentos de los cuales no vacíos o con más de una palabra hay 57.987.

A continuación, debemos responder a la pregunta de si todos esos documentos son válidos para un modelo en el que se realice una extracción de características sobre texto. Para ello nos fijaremos en el número de palabras por documento. En el siguiente grafico de barras se puede ver la distribución de los documentos frente al número de palabras:

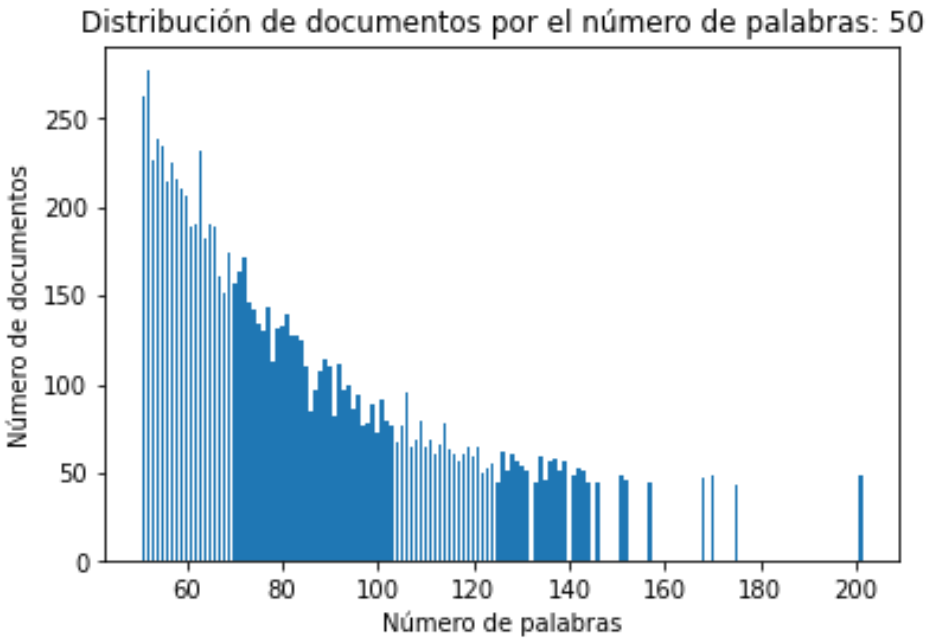


El gráfico está truncado para mostrar solo los 50 primeros valores sobre el eje x.

Para poder abordar un aprendizaje lógico y factible se ha definido el siguiente filtro sobre el conjunto de datos:

> "Se considerará como documento todo texto que contenga más de 50 palabras"

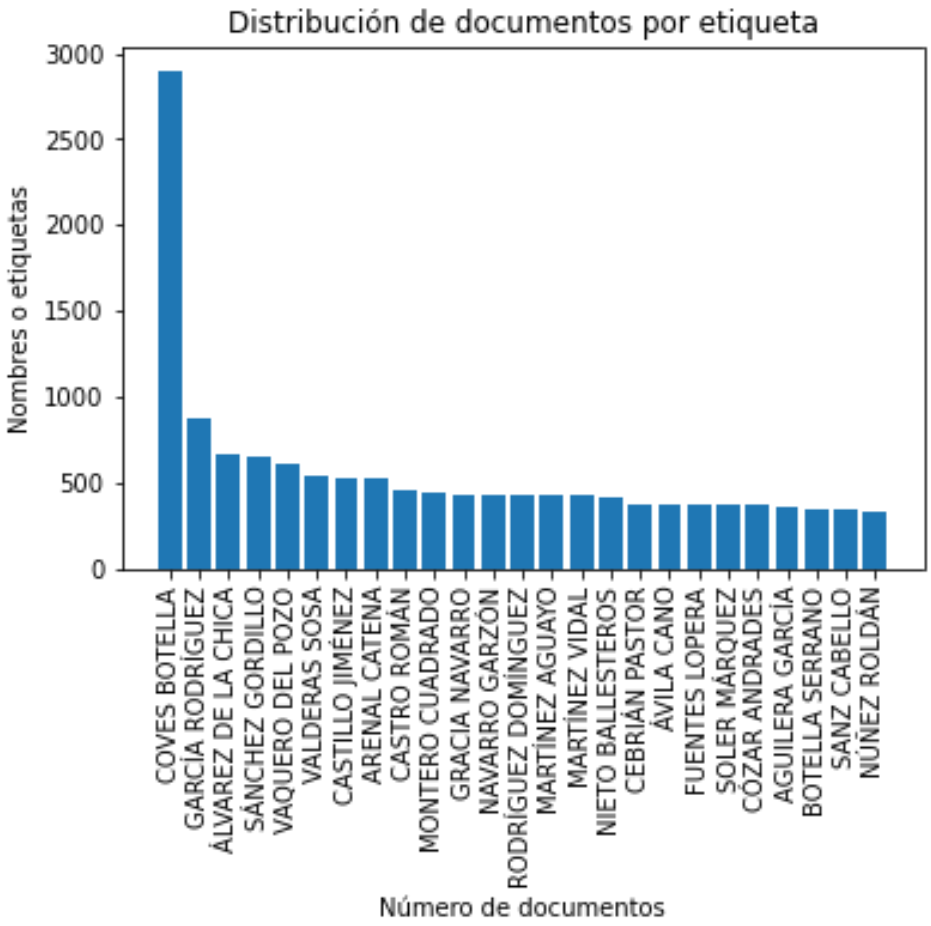
Por lo tanto, el gráfico de barras anterior quedaría actualizado así:



El gráfico está truncado para mostrar solo los 50 primeros valores sobre el eje x.

Los datos que nos quedan después de la aplicación del filtro son: 31.906 documentos y 332 etiquetas.

A continuación, vamos a realizar un conteo del número de documentos por nombre y etiqueta. Con ello perseguimos eliminar casos absurdos de aprendizaje sobre etiquetas que no tienen apenas documentos. Realizando el conteo de nombres por número de documentos.



Del gráfico se puede concluir que un subconjunto minoritario acumula prácticamente la totalidad de documentos del conjunto de datos. Es por ello, que debemos aplicar un segundo filtro para eliminar aquellas etiquetas con una frecuencia escasa. Observando los datos vemos que 272 de los 332 restantes tienen menos de 200 documentos. Por tanto, aplicamos el segundo filtro:

"Se considerará etiqueta a partir de los 200 documentos etiquetados a la misma fuente."

Los datos que nos quedan después de la aplicación del segundo filtro son: 23.701 documentos y 60 etiquetas. A partir de este momento, cuando nos referimos al conjunto de datos daremos por sentado que se han aplicado estos dos filtros para la implementación y ajuste de los modelos.

Los valores estadísticos a nivel de número de palabras resultantes son los siguientes:

	Valor
Conteo	23.701
Media	752,32
Desviación	1.060,69
Mínimo	51
Percentil 25%	94
Percentil 50%	309
Percentil 75%	975
Máximo	15.745

Con el objetivo de realizar pruebas justas entre todos los modelos se ha fijado la semilla para la división entre entrenamiento y test. El tamaño de estos dos conjuntos se ha elegido siguiendo el estándar 80% entrenamiento y 20% test o validación. Resumiendo, quedan estos dos conjuntos:

- Entrenamiento: 18.960 entradas.
- Validación: 4.741 entradas.

5 Fundamentos teóricos

5.1 Extracción de características

La extracción de características es el proceso de transformación de datos arbitrarios como texto o imágenes en características numéricas. Las características resultantes serán combinación de una o más característica de la matriz de entrada.

5.1.1 Bolsa de Palabras

La estructura de datos o representación Bolsa de Palabras o en inglés Bag of Words simplifica de forma eficiente el uso de texto en Aprendizaje Automático. Simplemente se base en realizar un conteo de las apariciones de una palabra a lo largo del corpus. Esto descarta gran parte de la complejidad que aportan las estructuras del lenguaje como capítulos o párrafos. Por lo tanto, llamaremos bolsa a la distribución de las frecuencias absolutas de las palabras en el corpus.

$P(c_1 : N)$ probabilidad de la secuencia de N caracteres en desde c_1 hasta c_N

Para computar la bolsa de palabras se siguen los siguientes pasos:

1. Token: definir la expresión que reconocerá una palabra para nuestro corpus. Ejemplo: una palabra es un conjunto de caracteres rodeados por espacios.
2. Tokenización: dividir cada documento en función del criterio que se haya definido en el token.
3. Construcción del vocabulario: realizar la unión algebraica de todos los conjuntos extraídos por la tokenización de cada documento.
4. Codificación: realizar el conteo de las apariciones de las palabras del vocabulario sobre los documentos.

Usualmente para aumentar la probabilidad de emparejamiento de palabras y reducir la bolsa de palabras se siguen algunas pautas:

- Unificación de mayúsculas minúsculas: se realiza una transformación sobre los caracteres del corpus pasándolos todos a minúscula, por ejemplo. En algunos corpus en los que abunden muchos nombres propios o de organizaciones puede llevar a un encarecimiento de la información extraía. El apellido “Pino” y el árbol “pino” pasarían a ser el mismo token dentro de la bolsa.
- Lematizador: mediante un algoritmo se extrae la palabra de la que deriva la original. Algoritmos como *WordNetLematizer* o *Lancaster,* entre otros, nos

ayudan a realizar esa función. Este tipo de algoritmos parten de la idea de que existen agrupaciones de palabras que describen grafos dirigidos con grandes sumideros. Aprovechar esos sumideros a costa de perder cierto grado de información puede ser interesante si aumenta el número de emparejamientos.
- Derivación: es una operación con el mismo objetivo que el lematizador. La principal deferencia, es que en este caso se realiza una extracción de la raíz.
- Eliminar acentuación: es una práctica muy común en los idiomas que provienen del latín.

Con el objetivo de eliminar palabras extremadamente poco frecuentes y muy frecuentes se definen dos umbrales:

- Frecuencia de documento mínima: se trata de la frecuencia relativa mínima que admitimos para considerar una palabra dentro de la bolsa de palabras.
- Frecuencia de documento máxima: se trata de la frecuencia relativa máxima que admitimos para considerar una palabra dentro de la bolsa de palabras.

Un tipo de umbral con el mismo propósito es el que limita el tamaño del vocabulario. Usualmente se seleccionan las k tokens más frecuentes encontrados.

Muchas de las librerías de procesamiento del lenguaje natural ofrecen conjuntos de palabras por idioma llamados **stopwords**. Estos grupos recogen las palabras más frecuentes que sirven como nexos entre ideas. Por lo general se sigue el criterio de que si se quitaran de una frase no se debería de ver mermada la información que se refleja. Dependiendo del conjunto de datos y problema, la eliminación de las stopwords puede aportar mejoras significativas en los resultados de la predicción.

Una de las principales desventajas que tiene esta estructura es que perdemos la información de la posición que tiene cada palabra con respecto al resto. Esta propiedad tiene una relación directa con el lenguaje, ya que puede no tener el mismo significado una palabra al principio de una frase, en medio o al final. Este significado dependiente de la posición debemos trasladárselo al modelo. Una de las formas más frecuente de hacerlo es partir del parámetro *N-gram*. Consideras un conjunto de palabras extra que cumple las mismas propiedades anteriormente descritas pero que define su token como unión de dos o más palabras dependiendo del rango que fijemos. Los valores frecuentes para el rango de *N-gram* van desde *1-gram* hasta *3-gram* inclusive. Esto se debe a que se experimenta una fuerte disminución de la frecuencia cuando se usan valores mayores de tres.

Este último punto ha adquirido tal importancia que ha conformado un concepto de modelos dentro de la práctica.

5.1.2 Modelos *N-gram*

Un *N-gram* se puede definir como una cadena de Markov de orden $n-1$, puesto que la probabilidad sobre el carácter c_i depende exclusivamente de los caracteres que le preceden. Por lo tanto, para un *3-gram*:

$$P(c_i|c_{1:i-1}) = P(c_i|c_{i-2:i-1})$$

Los modelos *N-gram* son usualmente aplicados de tres formas dependiendo la tokenización que se realice sobre el corpus:

- Realizado a nivel de **palabras** que se dividen por espacios o signos de puntuación. Los valores usuales no sobrepasan *3-gram*.
- Realizado a nivel de **caracteres**. En estos casos el rango está más abierto dependiendo de la aplicación que se le dé al modelo.
- Realizado a nivel de **caracteres dentro de una misma palabra**. Este caso es más frecuente y común que el anterior, ya que no suele tener mucho sentido que la cadena del vocabulario pertenezca a dos palabras diferentes de un texto.

Normalmente, dentro de estos modelos basados en bolsas de palabras si se da el caso de un *N-gram*, los *(N-i)-gram* que lo contienen también forman parte del vocabulario, dando la posibilidad de detectar tokens concretos de forma separada, pero sabiendo que no se presentan juntos.

(Russell & Norvig, N-gram character models, 2010)

5.1.3 Vectorización de documentos

El proceso de vectorización de un conjunto de documentos se resume en los siguientes puntos:

1. Construcción de un vocabulario a partir del corpus.
2. Siendo el número de componentes del vocabulario N, para cada documento d_i extraemos su vector correspondiente v_i que recoge las frecuencias de cada uno de los tokens en dicho documento.

Un ejemplo de aplicación de este proceso sobre la siguiente frase:

$$d_i = \text{"}que\ es\ lo\ que\ quieres\text{"}$$

Construcción del vocabulario:

Palabra	Frecuencia
que	2
es	1
lo	1
quieres	1

Vectorización del documento:

$$v_1 = (2,1,1,1)$$

Cuando se realiza este proceso sobre un número ingente de documentos el vocabulario que se obtiene genera vectores individuales con una alta frecuencia de valores 0. El conjunto de estos vectores se conforma en una matriz $M_{n\,x\,m}$ siendo n el número de documentos y m el tamaño del vocabulario. Esta matriz para ser almacenable en memoria requiere de una estructura de datos especial llamada **matriz dispersa**. Las matrices dispersas son un conjunto de ternas (i, j, k) donde i, j son los índices de posición de la matriz y k el valor que toma esa posición. El resto de combinación de índices no recogida, que se consideran 0.

El uso de matrices dispersas tiene como ventajas:

- Disminución en el uso de memoria
- Disminución del tiempo consumido de cómputo para ciertas operaciones

Pero también presenta una desventaja notable:

- Muchos de los modelos de Aprendizaje Automático no aceptan representaciones dispersas de las características, obligándonos a realizar entrenamientos parciales o incrementales tras una posterior transformación a matrices densas. Algunos modelos tampoco aceptan entrenamientos parciales.

5.1.4 TF-IDF

Es un uso más que extendido dentro del proceso de extracción de características sobre texto la aplicación de TF-IDF tras la vectorización de nuestro corpus.

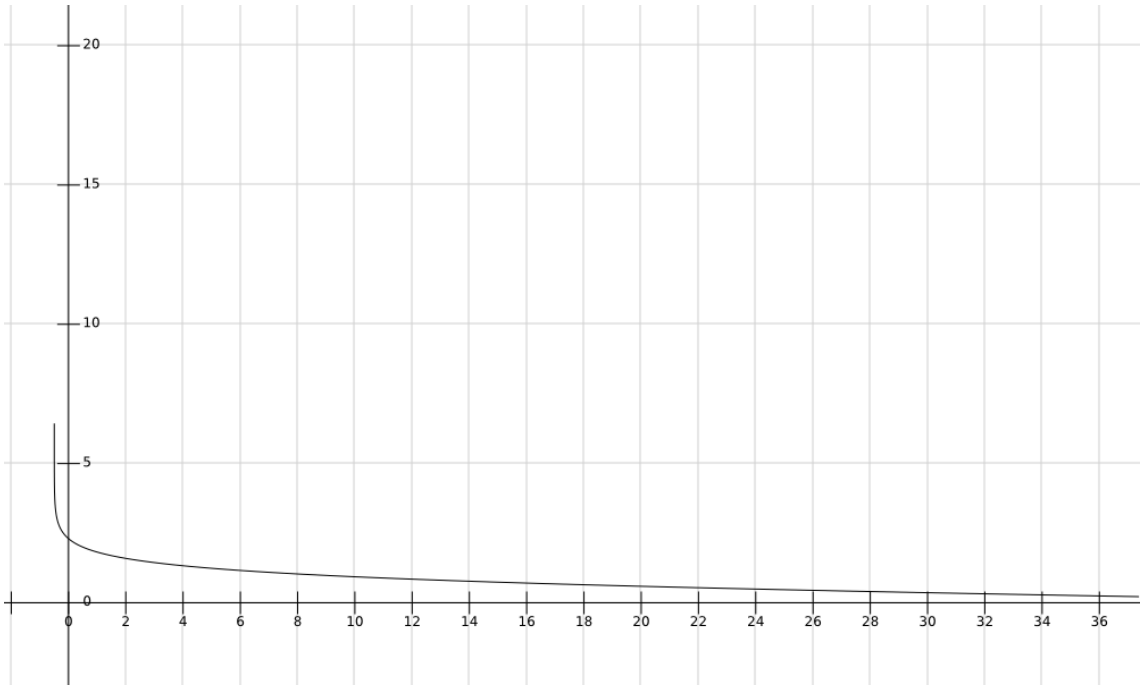
TF hace alusión a la frecuencia de un término dentro del corpus mientas que IDF refleja la inversa de la frecuencia sobre el documento. Se trata de una función de puntuación llamada **BM25**. Fue propuesta por Stephen Robertson y Karen Sparck Jones.

El objetivo TF-IDF es mezclar en una misma coordenada del documento la importancia del token sobre el corpus y sobre el documento. Hasta ahora el proceso de vectorización sólo recogía propiedades elegidas por el corpus con información exclusivamente del documento.

Definición de IDF:

$$IDF(q_i) = \log \frac{N - DF(q_i) + 0.5}{DF(q_i) + 0.5}$$

Donde N es el número de documentos en el corpus y $DF(q_i)$ el número de documento que contienen q_i. Analicemos el comportamiento de esta función mediante la siguiente gráfica para un valor de $N = 100$:



Es observable que conforme aumenta el número de documentos que contienen q_i el valor de IDF cae logarítmicamente decayendo su importancia.

La función TF-IDF es una función en dos variables que se aplica sobre el documento d_j el token q_i y que tiene la siguiente expresión:

$$TF - IDF(d_i, q_{i:N}) = \sum_{i=1}^{N} IDF(q_i) \cdot \frac{TF(q_i, d_j) \cdot (k+1)}{TF(q_i, d_j) + k \cdot (1 - b + b \cdot \frac{|d_j|}{L}}$$

Donde $TF(q_i, d_j)$ es el número de veces que aparece el token q_i en el documento d_j, L es la media de tokens que aparecen por documento, $k = 2.0$ y $b = 0.75$ usualmente.

Un punto interesante de esta función es asignar el valor 0 donde el proceso de vectorización asigne un 0. De otro modo, la eficiencia en memoria de la representación dispersa de la matriz se volvería en nuestra contra. Esto es fácilmente comprobable gracias a la multiplicación del numerador de la fracción.

5.2 Selección de características

El proceso de selección de características es aquel que elige un subgrupo de las características existentes en función de un criterio definido con el objetivo de discernir que características son influyentes en el problema.

Existe muchos algoritmos de selección de características. A continuación, mediante la siguiente tabla se resume la utilidad de algunos dependiendo de las características de entrada y del tipo de etiqueta.

Input \ Output	Numérica	Categórica
Numérica	Pearson's, Spearman's	ANOVA, Kendall's
Categórica	ANOVA, Kendall's	Chi-Squared, Mutual Information

En nuestro caso, teniendo en cuenta que la salida de una extracción de características sobre texto siempre es numérica, como se ha mencionado en el apartado anterior, nos limitaremos a explicar ANOVA y Kendall's.

5.2.1 ANOVA

El test ANOVA tiene las siguientes asunciones que deben satisfacerse para poder computar un $p - value$ valido.

- Las muestras deben ser independientes
- Cada muestra debe seguir una distribución normal
- La desviación estándar de la población de los grupos debe ser la misma. Esta es la propiedad conocida como homocedasticidad.

Si dice que un modelo es homocedástico si en todos los grupos de datos de una observación, la varianza del modelo respecto de las variables explicativas se mantiene constante.

El computo del F-Valor viene dado por la siguiente expresión:

$$F = \frac{\sum_{i=1}^{K} \frac{n_i(\overline{Y_i} - \overline{Y})^2}{K-1}}{\sum_{i=0}^{K}\sum_{j=1}^{n_i}\left(Y_{ij} - \bar{Y}_i\right)^2/(N-K)}$$

Donde K el numero de grupos, $\bar{Y}_i$ es la media de la columna y n_i el número de elementos en la columna. $K-1$, $N-K$ se corresponde con los grados de liberta de la distribución de Fisher.

Este test se realiza para cada una de las características extraídas por el modelo *N-gram* obteniendo las k que presente mejores resultados en el test.

5.2.2 Kendall's

En estadística, el coeficiente de correlación de Kendall es usado para medir la asociación ordinaria entre 2 mediciones cuantitativas.

Este coeficiente se puede expresar mediante la siguiente fórmula:

$$\tau = \frac{n_c - n_d}{\binom{n}{2}}$$

Donde n_c son el número de pares de puntos sobre las dos muestras que son concordantes. Es decir, que son mayores o menores en sus dos coordenadas:

$$(x_i, y_i)\ concuerda\ con\ \left(x_j, y_j\right)\ si\ (x_i < x_j\ \&\ y_i < y_j)\ |\ (x_i > x_j\ \&\ y_i > y_j)$$

Donde n_d son el número de pares de puntos sobre las dos muestras que con discordantes. Es decir, que son mayores en una coordenada y menores en otra.

Los puntos que son iguales en ambas coordenadas no se consideran ni concordantes ni discordantes.

Donde $\binom{n}{2}$ es el coeficiente binomial para la cantidad de formar de elegir 2 elementos de n elementos.

Definido el coeficiente de correlación se calcula la matriz de correlación de todas las características. Este proceso se suele aplicar de dos formas:

- Seleccionar aquellas características que no presenten correlación superior a un umbral con ninguna característica.
- Seleccionar las n características que presente menor correlación con el conjunto.

5.3 Reducción de la dimensión

El proceso de reducción de la dimensión de un problema pretende realizar una simplificación de la matriz de características perdiendo la mínima información posible del problema. En algunas referencias este proceso se incluye en la extracción de características debido a su semejanza. Este proceso no se debe aplicar necesariamente cuando nos encontremos con problemas de alta dimensión, ya que existe modelos que tienen comportamientos aceptables en estos casos. La reducción de la dimensionalidad suele requerir en muchos casos un tiempo de cómputo demasiado grande. Es por esto que generalmente se han implementado métodos iterativos que aproximen a valores teóricos de publicaciones.

Sin duda el método más conocido dentro de esta área es el Análisis de Componentes Principales o por sus siglas en inglés PCA. En este trabajo no se va a tratar este debido a que no se puede aplicar sobre representaciones de datos dispersos. La representación de datos dispersa elimina la posibilidad de aplicar gran parte de los métodos disponibles como puede ser el Análisis de Independiente de Componentes o por sus siglas en inglés ICA. Los tres métodos más conocidos bajo estas restricciones son LDA, LSA y NMF. Nos centraremos en explicar estos dos últimos por sus buenos resultados.

5.3.1 LSA

El Análisis Semántico Latente es un proceso de reducción de la dimensionalidad estrechamente ligado a la extracción de características sobre texto. LSA es una técnica de Procesamiento de Lenguaje Natural al mismo tiempo que una técnica del aprendizaje no supervisado. Por tanto, como su nombre indica, está buscando conocimiento latente o inherente en los datos de por sí solos, por medio de representaciones del texto en términos de temáticas y palabras clave. La aplicación de LSA suele ir después de la vectorización y la transformación TF-IDF de nuestro corpus.

El primer paso para computo de LSA es la Descomposición en Valores Singulares, o por sus siglas en inglés SVD, de la matriz de características. Resumiendo brevemente este proceso, expresamos la matriz original a partir de tres matrices:

$$\mathrm{M} = \mathrm{U}\Sigma\mathrm{V}^* = \sum_{i=1}^{m} \sigma_i u_i v_i^*$$

Supongamos que $M_{n\,x\,m}$ tiene n filas y m columnas, entonces las matrices tendrán las siguientes dimensiones $U_{n\,x\,n}, \Sigma_{n\,x\,m} V^*_{m\,x\,m}$ donde la diagonal de Σ refleja los valores singulares de M ordenados de mayor a menor. Aplicándolo a un problema de ML, m representa el número de características y n el número de datos.

La forma más común de proceder es fijar un número máximo de compontes a calcular sustituyendo en la fórmula anterior m por un k:

$$\mathrm{M}_k = \sum_{i=1}^{k} \sigma_i u_i v_i^*$$

A este cambio sobre el cálculo de la SVD se le conoce como SVD truncada.

https://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf

https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf

5.3.2 NMF

5.4 Modelos

5.4.1 Red Neuronal

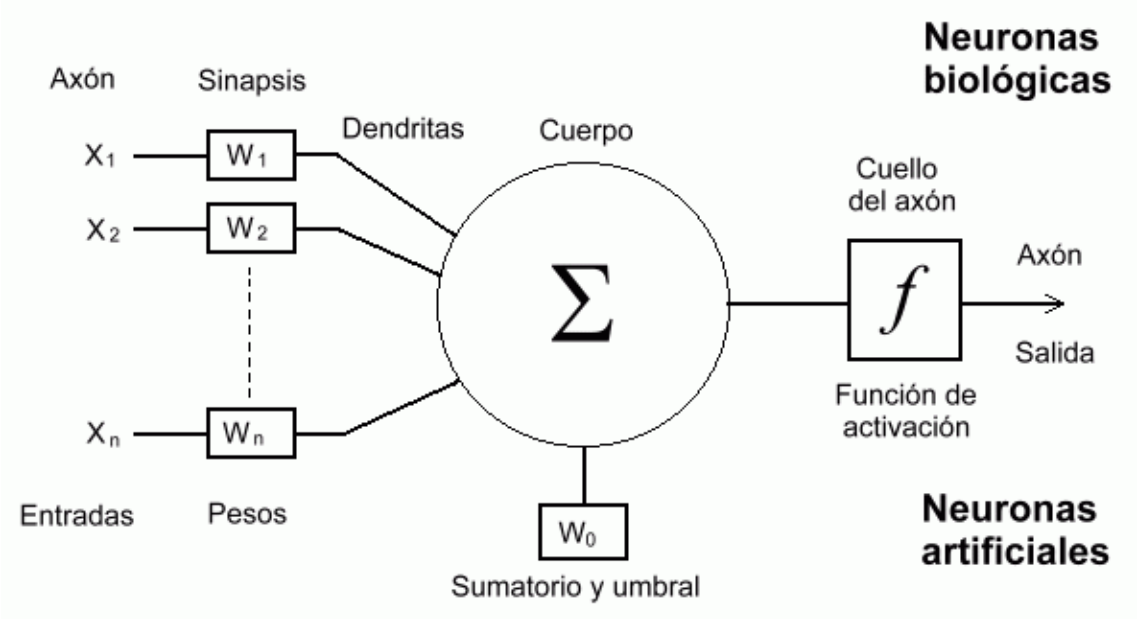
Uno de los modelos más conocidos dentro de Aprendizaje Automático son las redes neuronales artificiales. Un modelo inspirado en el comportamiento del cerebro humano que pretenden simular el intercambio de información entre las **neuronas**.

Las redes neuronales se estructuran en capas que forman un grafo dirigido que parte desde los inputs de las características hasta los outputs del etiquetado. Las capas intermedias que no son ni inputs ni outputs se les llama capas ocultas. Las capas están formadas por neuronas. Cada capa puede tener un número deferente de neuronas.

Usualmente a los nodos del grafo se les llama unidades. Al enlace que une la unidad u_i con la unidad u_j se le asocia un peso w_{ij}.

Los modelos Multicapa Perceptrón son un subconjunto de los modelos basados en redes neuronales. Fueron los primero en aparecer debido a la simplicidad de su arquitectura.

A continuación, se muestra la estructura que tendría una neurona o unidad dentro de estos modelos haciendo una comparativa bilógica:



- Entradas: se corresponde con información exterior del modelo o de neuronas conectadas ella.
- Pesos: como hemos descrito antes son los parámetros (usualmente números reales) que debe ajustar el modelo en su aprendizaje.
- Sumatorio: se produce tras multiplicar cada entrada por su peso. De esta forma pasamos de una entrada de dimensión N a una entrada de dimensión 1.
- Función de activación: se aplica al resultado de la sumatoria cerrado el proceso de transformación de los datos.
- Salida: puede ser tanto el resultado final del modelo o las posibles conexiones a otra capa de neuronas.

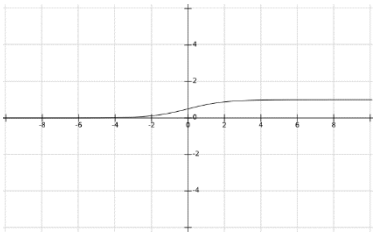
La expresión matemática que resume este proceso es la siguiente:

$$f(\sum_{i=1}^{K} X_i W_i + b)$$

Siendo f la función de activación y b usualmente llamada bias una constante de entrada a la neurona.

Existen varias propuestas de funciones de activación. A continuación, se explican las tres más conocidas y sus casos de aplicación.

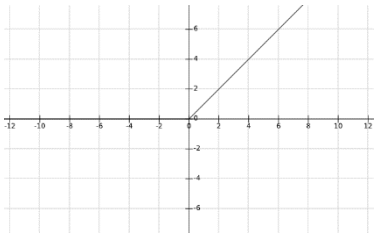
Función Sigmoide



La característica principal de esta función es su recorrido (0,1). Usada en la capa de salida para problemas de clasificación binaria. La expresión matemática que la define es la siguiente:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Función Relu



Función que posibilitó las redes neuronales profundas. Propuesta por Hahnoloser. Suele aplicarse en capas ocultas. La expresión matemática que la define es la siguiente:

$$f(x) = \max(0, x)$$

Función SoftMax

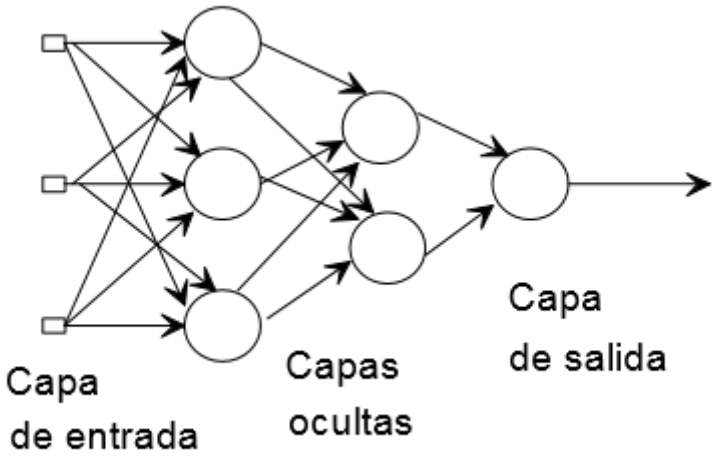
Función Softmax es comúnmente utilizada para clasificación de problemas multietiqueta dentro de entrenamiento de redes neuronales. Solemos encontrarla en la última capa de la red. La función devuelve un vector de K dimensiones, donde K es el número de clases de nuestro problema y cada competente la probabilidad de pertenencia a la clase.

$$f: \mathbb{R}^K \to \mathbb{R}^K$$

$$f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \; para\; i = 1, \dots, K \; y \; x = (x_1, \dots, x_K) \in \mathbb{R}^K$$

Otras funciones derivadas o con las mismas propiedades que estas se han usado como pueden ser Selu, Elu o tangente.

Cuando muchas neuronas pertenecen a un mismo modelo formando capas forman estructuras visuales que comúnmente se representan mediante esquemas de grafos dirigidos:



- Capa de entrada: se corresponde con los datos de entrada del problema.
- Capas ocultas: capas que reciben datos calculados por campas anteriores y conectan con capas posteriores.
- Capa de salida: se corresponde con la predicción del algoritmo.

Para realizar el proceso de entrenamiento se emplea el algoritmo de Propagación Hacia Atrás o en inglés **Back Propagation**, elaborado y popularizado por Rumelhart, Hinton y Williams. Resumidamente el algoritmo computa el gradiente de la función de perdida para cada capa, iterando hacia atrás reutilizando los datos. Este esquema de reutilización es una estrategia de programación dinámica.

Uno de los problemas más difíciles de resolver dentro de estos modelos es el sobreajuste o overfit. Llamamos sobreajuste al efecto de sobreentrenar un modelo de tal forma que abandone su función de aprendizaje por la de memorizar. La técnica sin duda más usada es **dropout**. Un método de regularización que elimina unidades tanto de capas ocultas como visibles. Al eliminarlas un porcentaje de estas durante un proceso de entrenamiento reducimos notablemente la complejidad

Para realizar la búsqueda del mejor modelo de MLP se han simplificado los parámetros a 5 que se explican a continuación:

- Número de capas ocultas: se corresponde con el número de capas internas con las que se genera el modelo.
- Número de unidades: se corresponde con el número de neuronas que tendrá cada capa interna.
- Tasa droupout: es la tasa de enlaces entre dos capas que el algoritmo debe quitar.
- Dimensión input: dimensión de la primera capa.
- Dimensión output: dimensión de la última capa que debe concordar con el número de clases del problema.

5.4.1.1 Densa

5.4.1.2 Dropout

5.4.1.3 Pooling

5.4.1.4 Embedding

5.4.1.5 Convolucionales 1D

5.4.1.6 LSTM

5.4.1.7 GRU

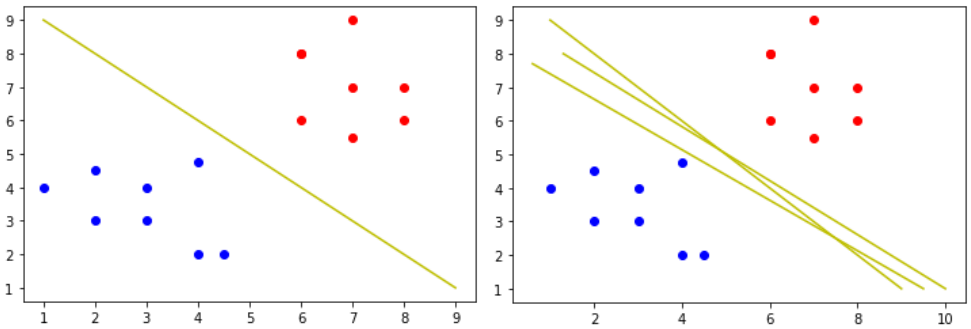
5.4.2 Máquina de Vectores Soporte

La Máquina de Vectores Soporte es uno de los modelos más usados dentro del Aprendizaje Automático supervisado. Se puede encontrar implementado en la mayor parte bibliotecas genéricas de ML. Hay cinco propiedades que lo hacen diferenciarse al resto de modelos:

1. Pretende **maximizar el margen de separación** del conjunto de datos con el objetivo de tener un modelo más genérico y robusto.
2. Originalmente muchos problemas presentan datos que no son linealmente separables para las dimensiones originarles. SVM haciendo uso de **kernels** aumenta las dimensiones del problema llevándolo a un espacio donde los datos son linealmente separables por un hiperplano.
3. SVM es un **modelo no paramétrico**. Ser no paramétrico quiere decir que la complejidad de su espacio de hipótesis crece según lo hace el número de datos de entrada. Esto representa una ventaja frente al sobreajuste y una desventaja frente a uso de los recursos. Está desaconsejado el uso de SVM para conjuntos de datos grandes.
4. SVM permite realizar entrenamientos con representaciones de datos dispersas, lo que lo hace un modelo ideal en combinación con una extracción de características sobre texto.
5. SVM debido a su característica no paramétrica **no** permite realizar **entrenamientos parciales o incrementales**. El uso actual para conjuntos de datos que no son posibles alojar en memoria requiere usar plataformas de entrenamiento más sofisticadas. En la publicación científica "Incremental and Decremental Support Vector Machine Learning" apuntan una modificación del algoritmo que permite el entrenamiento parcial. Por otro lado, la mayor parte de las guías concluyen en cambiar de modelo si se produjera esta situación.

Margen

Imaginemos la aplicación de un modelo SVM sobre un problema de clasificación binario linealmente separable, donde el número de hiperplanos posibles que dividan el espacio de las etiquetas sea superior a 1. Por lo tanto, existan infinitos hiperplanos posibles que realicen la separación.



En términos relativos, podríamos definir una "seguridad" en la predicción realizada diciendo que puntos más alejados de la frontera de decisión representan una mayor seguridad sobre datos más cercanos. En la gráfica de la izquierda se puede observar que la coordenada $(7,9)$ se encuentra más lejana de la frontera de decisión que la $(7,5)$.

Refiriéndonos a la segunda gráfica, cómo podemos saber cuál de los hiperplanos o separadores es el que mejor explica nuestro conjunto de datos:

- ¿El qué mayor número de puntos clasifique?
- ¿El que deje mayor distancia entre la frontera de decisión y el conjunto de puntos?

Generalizar estas dos ideas para todos los posibles hiperplanos que separan un conjunto de datos requiere una notación previa.

Notación

Consideraremos las etiquetas $y \in \{-1,1\}$ en lugar de $\{0,1\}$.

El parámetro independiente del clasificador se demorará de la siguiente forma:

$$h_{w,b}(x) = g(w^T x + b)$$

Donde $g(z) = 1$ si $z \geq 0$, si no $g(z) = -1$.

Problema de minimización

Resolución por Lagrange

Kernels

Regularización

Concepto de margen

Definamos el siguiente problema de se

Enlace: http://cs229.stanford.edu/notes/cs229-notes3.pdf

6 Métricas

6.1 Exactitud o *accuracy*

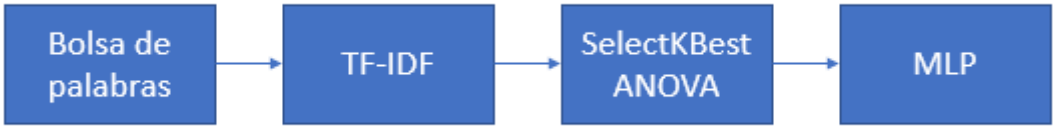
6.2 Precisión

6.3 Exhaustividad o *recall*

6.4 Validación cruzada

7 Experimentos

7.1 Modelo TF-IDF ANOVA MLP



Modelo compuesto de 4 componentes:

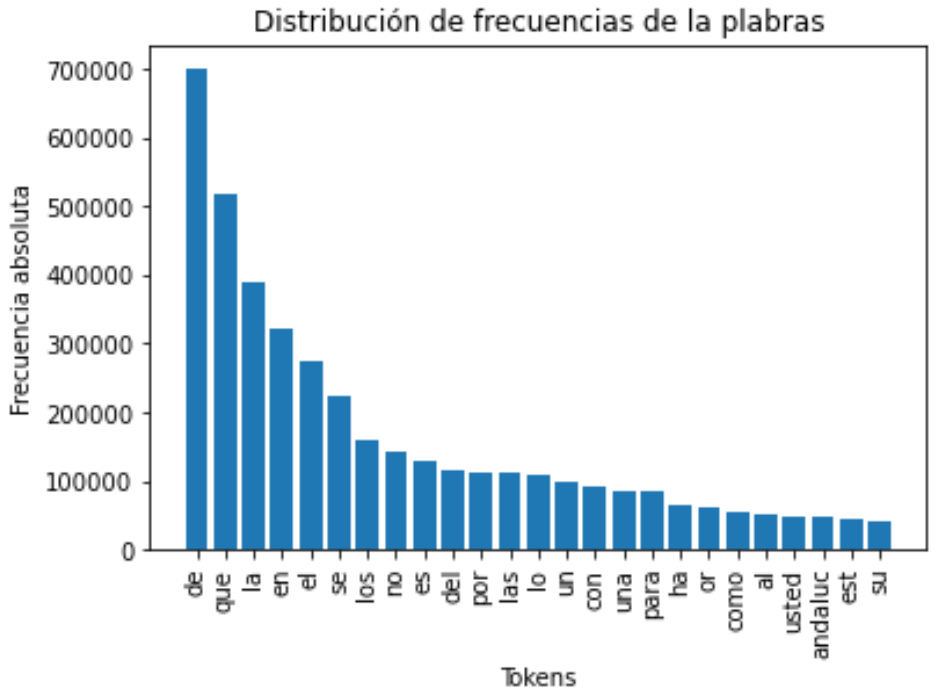
1. Bolsa de palabras para realizar una extracción de características sobre texto.
2. TF-IDF transformación que mantiene las dimensiones del conjunto de características, pero aporta información extra.
3. Selección de las mejores características en función del test ANOVA.
4. Red neuronal multicapa perceptrón para la clasificación de las etiquetas a partir de las características extraídas y seleccionadas.

Bolsa de palabras

Para el conjunto de datos que nos atañe se ha seguido un análisis o tokenización a nivel de palabras. En la siguiente tabla se puede ver las 10 más frecuentes de todo el corpus:

Palabra	Frecuencia absoluta
de	700.842
que	518.938
la	389.484
en	321.489
el	272.735
se	223.706
los	158.813
no	141.399
es	128.702
del	116.744

Para este análisis sólo se han eliminado los acentos y se han pasado todas las palabras a minúscula, dando un total de 64.259 palabras o características *1-gram*. Se puede ver a continuación un diagrama de barras de las 25 primeras ordenadas por frecuencia:



Sobre nuestro conjunto de datos obtenemos las siguientes características para valores de *N-gram*:

N-gram	Número de características
(1,1)	64.259
(2.2)	1.360.480
(3,3)	4.451.936

Para la extracción de características de este modelo se ha usado el rango de *N-gram* (1,2) teniendo un total de 1.424.739 características. Ha este resultado se le ha aplicado tres filtros sobre la frecuencia:

- Frecuencia relativa mínima del documento: $\frac{1.0}{1000.0}$.
- Frecuencia relativa máxima del documento: $\frac{999.0}{1000.0}$.
- Número máximo de características: 100.000.

Tras la aplicación de estos tres filtros, el conjunto de características se reduce a 63.475.

Un último filtro mencionado en el apartado de bolsa de palabras es el conjunto de ***stopwords***. Un conjunto predefinido de tokens para los cuales sabemos que no suelen aportar información útil al modelo. Usualmente se suele incluir dos conjuntos:

- Signos de puntuación.
- Palabras comunes del lenguaje usado en el corpus.

Estas palabras se eliminan como posibles características del modelo reduciendo el ruido que provocan en la clasificación. En este caso hemos utilizado 345 de las cuales 32 son

de puntuación y 313 comunes del lenguaje (castellano). Algunos ejemplos de palabras comunes son:

- Preposiciones: por, con, para…
- Determinantes: la, el…
- Conjunciones: y, o, ni…
- Formas verbales: estoy, está, habrán, tuviera…

Tras la aplicación del filtro por *stopwords* reducimos el conjunto de características a 34.344. Si no tuviéramos una representación en memoria dispersa nos sería imposible trabajar con ese volumen de datos.

Para la aplicación de la vectorización, uno de los valores de referencia para la comparación de procesos de extracción de características sobre texto es la dispersión de los datos sobre la matriz vectorizada del corpus. En nuestro caso ese valor es el siguiente:

$$dispersión = 0{,}64\%$$

El valor se puede calcular fácilmente dividiendo el número de ternas de la matriz dispersa entre la multiplicación de las dimensiones de la matriz. La interpretación de estos datos es que el 0.64% de los valores de la matriz es distinto de 0. Esto no debe sorprendernos y debe ser interpretado positivamente. Los documentos no comparten un gran número de características, pero tampoco un número muy pequeño como para que no podamos sacar ningún patrón común. Algunas publicaciones apuntan que sobrepasar el 1% de dispersión en este tipo de extracción de características se consideraría una mala extracción.

Para finalizar este apartado una de las representaciones gráficas más comunes de las bolsas de palabras es la nube de palabras. El tamaño de la palabra refleja su frecuencia en el texto.

TF-IDF

Aplicamos la transformación TF-IDF sobre los datos ponderando como hemos apuntado antes a la baja a palabras muy comunes dentro del conjunto de datos. Esta transformación no altera la dimensión de nuestra matriz de características.

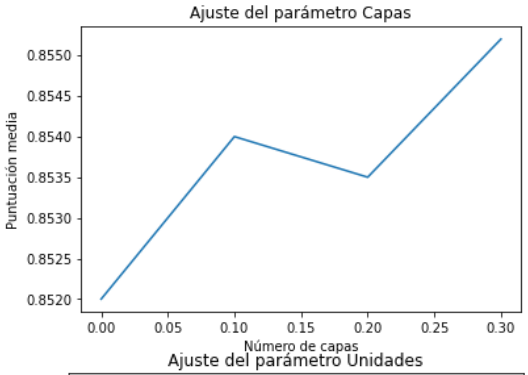
SelectKBest ANOVA

Llamamos al proceso de selección de características. Es de sobra conocido que, en la extracción de características sobre texto, al tener un gran número de columnas, en nuestro caso 34.344, muchas de ellas solo aporten ruido y confusión al problema. Para ello se han seleccionas las 20.000 mejores. El criterio de este número es el siguiente:

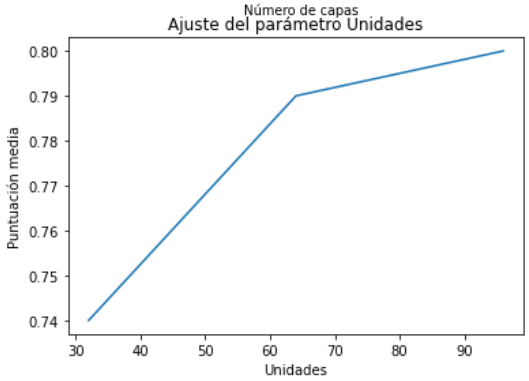
- Recomendación de la guía de implementación de este modelo.
- Pruebas realizadas con otros valores de K.

MLP

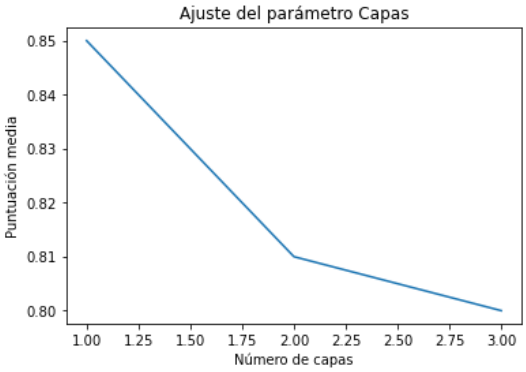
Los modelos MLP por defecto no admiten entrenamientos con matrices dispersas. Para resolver esta problemática se usa otra propiedad que, sí que tiene MLP, el entrenamiento parcial o incremental. Es tipo de entrenamiento permite pasar en lotes nuestro conjunto de datos por el modelo. El parámetro que define el tamaño del lote se llama $batch\_size$ y en nuestro caso toma el valor 1024.



Valor	Puntuación
0,0	0,85
0,1	0,85
0,2	0,85
0,3	0,86



Valor	Puntuación
32	0,79
64	0,84
96	0,86



Valor	Puntuación
1	0,85
2	0,81
3	0,73

7.1.1 Parámetros

Componente	Parámetro	Valor
Bolsa de palabras	Frecuencia relativa mínima	0,001
	Frecuencia relativa máxima	0,999
	Máximo de características	100.000
	Stopwords	Spain+Puntuación
	N-Gram	(1,2)
BestKBest ANOVA	K	20.000
MLP	Número de capas ocultas	1
	Número de unidades por capa	96
	Tasa dropout	0,3
	Dimensión input	20.000
	Dimensión output	60

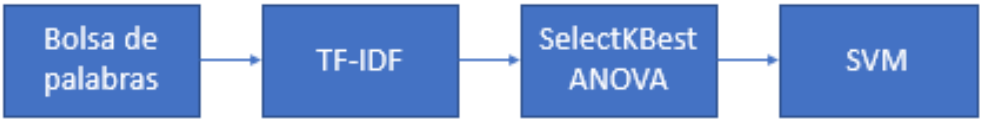
7.1.2 Resultados

División	1	2	3	4	5	Media	Desviación
Puntuación	0,85	0,85	0,86	0,86	0,86	0,86	0,0025

	Entrenamiento	Validación
Puntuación	1,00	0,87

Tiempo total de entrenamiento 1,3 minutos.

7.2 Modelo TF-IDF ANOVA SVM Lineal



7.2.1 Parámetros

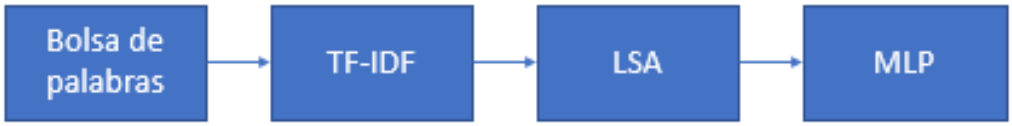
Componente	Parámetro	Valor
Bolsa de palabras	**Frecuencia relativa mínima**	0,001
	Frecuencia relativa máxima	0,999
	Máximo de características	100.000
	Stopwords	Spain+Puntuación
	N-Gram	(1,2)
BestKBest ANOVA	**K**	20.000
SVM	**C**	1

7.2.2 Resultados

División	1	2	3	4	5	Media	Desviación
Puntuación	0,85	0,85	0,86	0,86	0,86	0,86	0,0025

	Entrenamiento	Validación
Puntuación	1,00	0,89

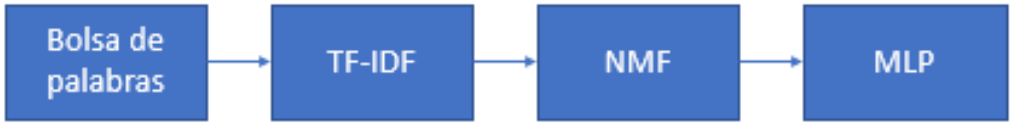
7.3 Modelo TF-IDF LSA MLP



7.3.1 Parámetros

7.3.2 Resultados

7.4 Modelo TF-IDF NMF MLP



7.4.1 Parámetros

7.4.2 Resultados

7.5 Modelo Embeddings LSTM

7.5.1 Parámetros

7.5.2 Resultados

7.6 Modelo Embeddings sepCNN

7.6.1 Parámetros

7.6.2 Resultados

8 Comparativa

Balanceados y no balanceado

Ajuste de hiperparamentros

Graficas sobre hiperparametros

9 Tecnología

Python

Sklearn

Tensorflow

Cuda

Pandas

Matplotlib

10 Estudio

11 Conclusión

12 Bibliografía

C. Müller, A., & Guido, S. (2016). *Introduction to Machine Learning with Python.* Sebastopol: O'REILLY.

Russell, S., & Norvig, P. (2010). Artificial Neural Networks. En S. Russell, & P. Norvig, *Artificial Intelligence A Modern Approach* (págs. 727-736). Harlow: PEARSON.

Russell, S., & Norvig, P. (2010). N-gram character models. In S. Russell, & P. Norvig, *Artificial Intelligence A Modern Approach* (pp. 861-862). Harlow: PEARSON.

Russell, S., & Norvig, P. (2010). Support Vector Machines. En S. Russell, & P. Norvig, *Artificial Intelligence A Modern Approach* (págs. 744-747). Harlow: PEARSON.

Singh, P., & Manure, A. (2020). *Learn TensorFlow 2.0.* Bangalore: APRESS.