Análisis de sentimiento de textos basado en opiniones de películas usando algoritmos de aprendizaje computacional

Jorge Chulilla Alcalde
Grado de Informática - Inteligencia Artificial

David Isern Alarcón
Carles Ventura Royo

Junio'21

1. Contexto y objetivos

- Hoy en día la maximización de venta de productos está fuertemente ligada al conocimiento y los datos relacionados. Para las grandes compañías, especialmente las plataformas online, hay un interés muy evidente en recoger información de su producto y determinar si gusta o no a sus clientes, y adaptar en caso necesario.

  - **Caso de uso: En el contexto de la industria audiovisual, analizar y clasificar las opiniones de los clientes respecto a sus contenidos.**

- Objetivos marcados en el TFG:

Objetivo principal: Análisis y evaluación de **algoritmos clasificadores de sentimientos** dedicados al análisis de textos relacionados con opiniones de películas o series.

Objetivo final: Proveer de una herramienta software para la prueba de los clasificadores obtenidos en el objetivo principal.

2. Alcance y premisas

❑ Alcance:

- ➢ Análisis de sistemas actuales NLP ("*Natural Language Processing*") y algoritmos de aprendizaje computacional.
- ➢ Desarrollo de un sistema NLP **utilizando tres algoritmos distintos**, siguiendo una metodología de trabajo basada en: definición de datos → preparación de datos → modelado → evaluación de los modelos.
- ➢ Desarrollo y generación de un aplicativo *front-end* web para usuario. (Objetivo final deseado)

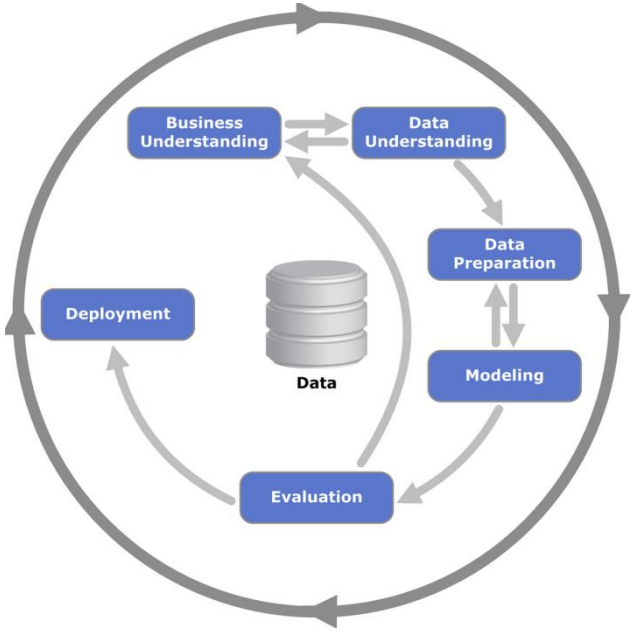


Diagrama de proceso basado en metodología de trabajo CRISP-DM

❑ Premisas iniciales:

- ✓ Disponer de las herramientas hardware/software necesarias, así como de un *dataset* confiable.
- ✓ No hay automatización de procesos.
- ✓ Estudio de opiniones para **textos en idioma inglés**.
- ✓ Categorización de opiniones: **positivas o negativas**.

3. Análisis del estado del arte – Análisis de sentimiento

❑ ¿Qué es el análisis de sentimiento?

"Sentiment analysis or opinion mining is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes." (Liu B. & Zhang L., 2012)

[El análisis de sentimiento o minería de opinión es el estudio computacional de opiniones de personas, evaluaciones, actitudes y emociones respecto a entidades, individuos, asuntos, eventos, temas y sus atributos.]

❑ Clasificación de opiniones

- Regulares Directas → *"Este coche es bonito."*
- Regulares Indirectas → *"Después de la revisión mi coche funciona bien."*
- Comparativas → "*Mi nuevo coche gusta más que el de la mayoría."*

4. Análisis del estado del arte – Problemáticas

A considerar

Subjetividad. Una sentencia subjetiva puede expresar algún tipo de sensación personal o creencia, pero no necesariamente un sentimiento.

Emociones. Odio, venganza, tristeza, amor,…, serian ejemplos de emociones que influirían en el sentimiento sobre una entidad.

Afectan

Multilenguaje. Connotaciones distintas según el idioma.

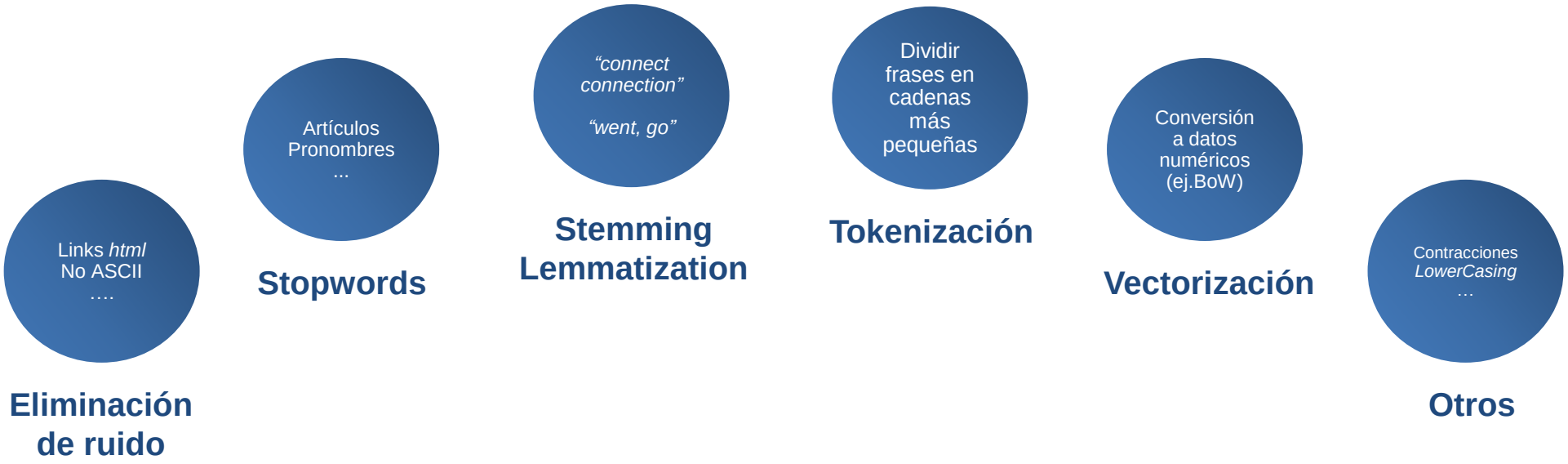
Agrupación de referencias. Conceptos iguales pero con palabras distintas.

Detección de spam y su influencia en las opiniones de los demás.

Sarcasmo e ironía.

5. Análisis del estado del arte – Técnicas: preprocesamiento

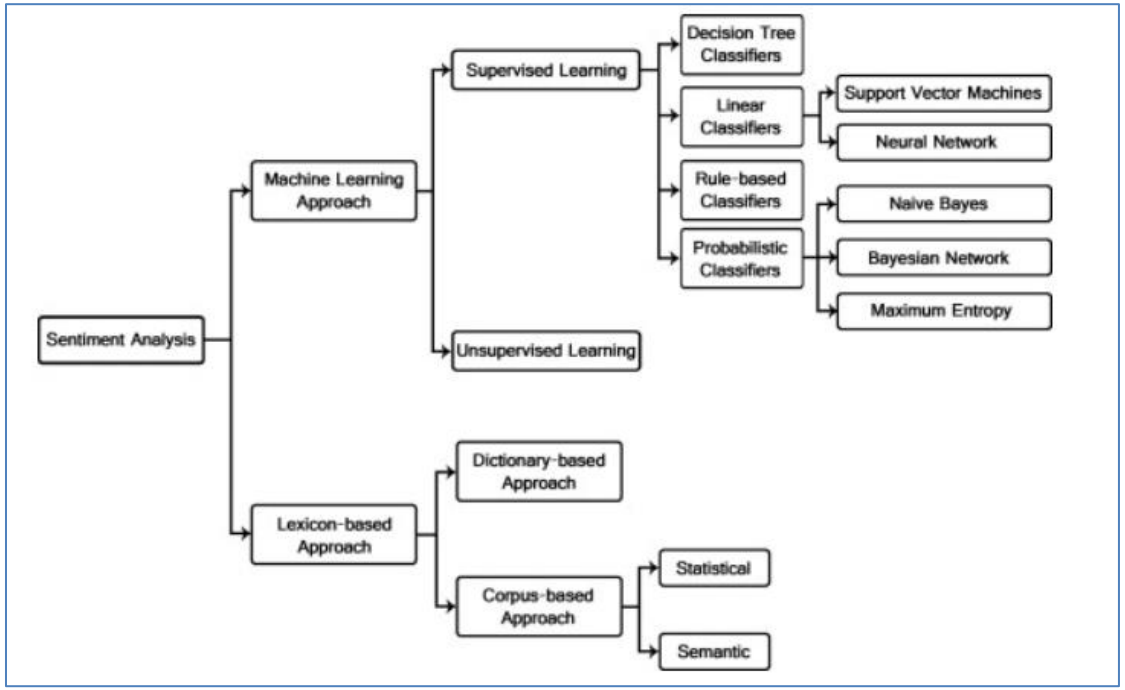
En este punto se busca refinar la calidad de la información origen, eliminando aquello que no sea necesario o transformando aquello que pueda ayudar al análisis.



6. Análisis del estado del arte – Técnicas: clasificación

La clasificación de objetos se basa en la utilización de una herramienta o algoritmo que permita generar los modelos de clasificación.

- **Técnicas utilizadas en el tratamiento NLP y clasificación de textos:**
  - ***Lexicon-based*.** Basados en la existencia de una base de datos léxica (Sentiword, Wordnet,..).
  - ***Machine Learning* no supervisado.** Clasifica objetos según sus características y una métrica que mida sus similitudes. Ejs.: Dendrogramas, K-Means,...
  - ***Machine Learning* supervisado.** Clasifica los objetos a partir de casos cuyo resultado ya es conocido. Ejs.: Logistic Regression, K-NN, árboles de decisión, redes neuronales,…



7. Análisis del estado del arte – Técnicas: evaluación

Una vez generados los modelos de clasificación será necesario su evaluación, determinando el grado de confianza de los mismos.

- **Validación simple.** Se trataría de dividir los datos de entrada en dos: datos de entrenamiento y de test. Los primeros son los utilizados para generar el modelo y una vez obtenido se comprueba contra los datos de test.
- **Validación cruzada.** Se basa en realizar *k* pruebas utilizando en cada una un subconjunto como test y el resto como entrenamiento. A partir de ahí se calcula la media y la desviación estándar de los resultados.

Y todo esto se expresa finalmente en una…matriz de confusión.

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

- **VP** es la cantidad de *positivos* que fueron *clasificados correctamente* como positivos por el modelo.
- **VN** es la cantidad de *negativos* que fueron *clasificados correctamente* como negativos por el modelo.
- **FN** es la cantidad de *positivos* que fueron *clasificados incorrectamente* como negativos.
- **FP** es la cantidad de *negativos* que fueron *clasificados incorrectamente* como positivos.

Medidas

- **Error.** Número de ejemplos mal clasificados respecto al total. $error = \frac{fp+fn}{tp+fp+tn+fn}$
- **Exactitud (“*Accuracy*”).** Número de ejemplos bien clasificados respecto al total. $exactitud = \frac{tp+tn}{tp+fp+tn+fn}$
- **Precisión.** Número de ejemplos bien clasificados respecto al total con predicción positiva. $precisión = \frac{tp}{tp+fp}$
- **Sensibilidad (“*Recall*”).** Número de ejemplos bien clasificados respecto al total de ejemplos positivos. $sensibilidad = \frac{tp}{tp+fn}$
- **F1.** Basado en precisión y sensibilidad, descartando los elementos negativos $F1 = 2 \times \frac{precisió \times sensibilidad}{precisión + sensibilidad} = \frac{2 \times tp}{2 \times tp + fn + fp}$

8. Solución propuesta – Diseño y arquitectura

- **Interfaz de usuario.** *Front-end* web para usuarios, permite la realización de pruebas.
- **Analítica avanzada.** Utilización de lenguaje Python en entornos locales y *cloud* (“*Google Colab*”) para el desarrollo y generación de los modelos.
- **Control de versiones.** Github, nos permite no solo eso sino también la compartición de código fuente.
- **Despliegue.** Imagen Docker, permite la ejecución del *front-end* con independencia de la máquina local.
- **Origen de datos.** Kaggle como repositorio de *datasets*.

9. Solución propuesta – Premisas de implementación

Definición del *dataset*

- Nuestro *dataset* seleccionado para el estudio se encuentra en un repositorio web de Kaggle específico para el análisis de sentimiento de opiniones de películas: https://www.kaggle.com/columbine/imdb-dataset-sentiment-analysis-in-csv-format

- ***Dataset* formado por 2 ficheros:**
  - **"*Train*" (40k filas)→ Entrenamiento**
  - **"*Test*" (5k filas) → Prueba**

- Estos ficheros se componen de 2 campos separados por ",":
  - "*Text*" → Opinión del contenido.
  - "*Label*" → Etiquetado. Indicador de sentimiento (0-negativo, 1-positivo).

Análisis y preparación de los datos

- Las opiniones de los usuarios rondan en su mayoría entre 250 y 10k caracteres.

- La distribución entre positivos y negativos está balanceada entorno al 50% tanto en entrenamiento como en *test*.

- Preprocesado utilizado:
  - Eliminación de links y marcas *html*, expresiones entre corchetes, caracteres no ASCII, guiones entre caracteres, espacios en blanco y caracteres especiales ("\", …), palabras de longitud igual a 1, *stopwords*.
  - *LowerCasing*, sustitución de contracciones, sustitución de códigos numéricos por texto.

Modelado y pruebas

- Se utilizan los siguientes algoritmos ML supervisados:
  - **MNB (Multinomial Naïve Bayes)**
  - **LR (Logistic Regression)**
  - **ULMFiT (Universal Language Model Fine-Tuning)**

- Pruebas a realizar:
  - ✓ *Stemming* vs *Lemmatization*.
  - ✓ Parametrización de la vectorización de tipo Tf-idf.
  - ✓ Parametrización de los algoritmos utilizados.
  - ✓ ULMFiT: Activación y desactivación de capas + ejecución de *epochs*.

Evaluación de resultados

- Generación de matriz de confusión de resultados.

- Métrica de validación: **exactitud** (*"accuracy"*).

10. Modelado (*Multinomial Naïve Bayes*)

- **Naïve Bayes** es un clasificador que **asume que las características son independientes de la clase dada**. Su funcionamiento se basa en el teorema de Bayes:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

, donde:

- P(H) es la probabilidad a priori, la forma de introducir conocimiento previo sobre los valores que puede tomar la hipótesis.
- P(D|H) es la probabilidad de obtener D dado que H es verdadera.
- P(D) es la probabilidad de observar los datos D promediado sobre todas las posibles hipótesis H.
- P(H|D) es la probabilidad a posteriori, o sea, la distribución de probabilidad final para la hipótesis.

Es un clasificador que funciona bien con datos continuos (p.e. la altura de una persona), pero no tanto con datos discretos (fijos). En este caso la preferencia es utilizar **Multinomial Naïve Bayes (MNB)**.

11. Resultados (*Multinomial Naïve Bayes*)

Stem/Lemmat. (S/L)	***Vectorization***	***Parameters***	***Accuracy(%)***
L	-	alpha=0.3	86,90%
S	-	alpha=0.3	86,30%
L	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	alpha=0.3	85,58%
S	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	alpha=0.3	85,24%
L	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(2,3)	alpha=0.3	65,78%
S	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(2,3)	alpha=0.3	67,32%
L	-	alpha=0.5	86,92%
S	-	alpha=0.5	86,48%
L	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	alpha=0.5	85,60%
S	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	alpha=0.5	85,26%
L	-	alpha=0.7	85,60%
S	-	alpha=0.7	85,30%

Parametrizaciones

Vectorization

- ***Max_df**:* Máximo porcentaje de palabras a no tratar. Las palabras que estén por encima de ese porcentaje en cuanto a frecuencia no se tratarían.
- ***Min_df**:* Ídem que el anterior, pero en el caso de las palabras que menos aparezcan.
- ***Ngram_range**:* Tratamiento de palabras o binomio de palabras.

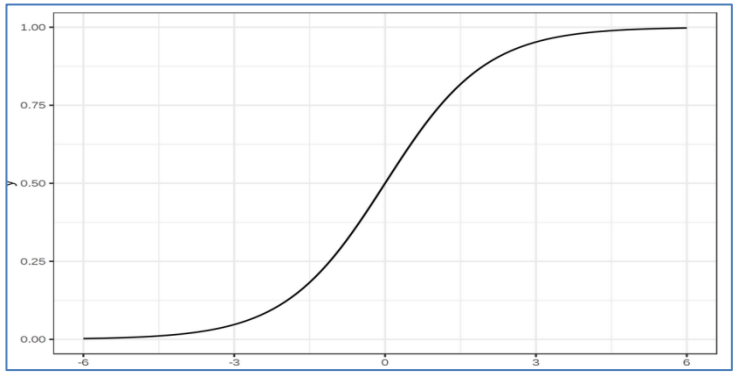
Parameters

- ***Alpha:*** "*Smoothing*" (suavizado) de valores categóricos. La idea es que ninguna probabilidad sea cero, regularizando el funcionamiento del algoritmo.

12. Modelado (*Logistic Regression*)

- **Logistic Regression (LR)** es un clasificador que mejora las capacidades de Linear Regression, y que **se aplica a búsqueda de valores binarios (0,1)**. Utiliza la función logística para representar los valores entre 0 y 1.

$$\text{logistic}(\eta) = \frac{1}{1 + exp(-\eta)}$$



Nos hemos decidido por este clasificador por dos razones concretas:

- ✓ Existen **diferentes estudios que prueban sus buenos resultados**, como el realizado por J.Lin y A.Kolcz en 2012 "*Large-Scale Machine Learning at Twitter",* utilizando *tweets.*
- ✓ LR no es solo un clasificador simple, sino que también **retorna probabilidades**, lo que es muy útil para valorar si una instancia concreta tiene mayor o menor probabilidad de pertenecer a una clase.

13. Resultados (*Logistic Regression*)

Stem/Lemmat. (S/L)	***Vectorization***	***Parameters***	***Accuracy(%)***
L	-	multi_class='multinomial'	90,02%
S	-	multi_class='multinomial'	89,74%
L	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	multi_class='multinomial'	87,94%
S	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	multi_class='multinomial'	87,48%
L	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(2,3)	multi_class='multinomial'	67,90%
S	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(2,3)	multi_class='multinomial'	69,32%
L	-	C = 0.5	89,34%
S	-	C = 0.5	89,14%
L	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	multi_class='multinomial' C = 0.5	87,96%
S	analyzer = "word", max_df=0.9,min_df=0.01, ngram_range=(1,3)	multi_class='multinomial' C = 0.5	87,76%
L	-	multi_class='multinomial' C = 0.3	87,90%
S	-	multi_class='multinomial' C = 0.3	87,84%

Parametrizaciones

Vectorization

- ***Max_df**:* Máximo porcentaje de palabras a no tratar. Las palabras que estén por encima de ese porcentaje en cuanto a frecuencia no se tratarían.
- ***Min_df**:* Ídem que el anterior, pero en el caso de las palabras que menos aparezcan.
- ***Ngram_range**:* Tratamiento de palabras o binomio de palabras.

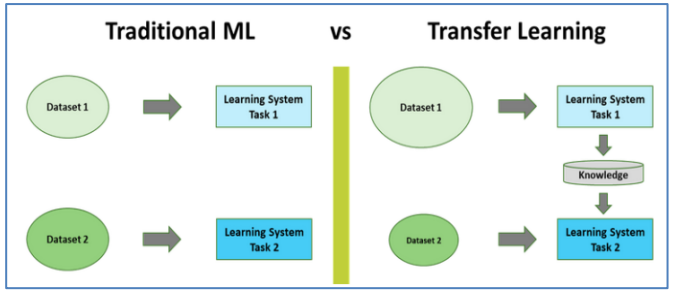
Parameters

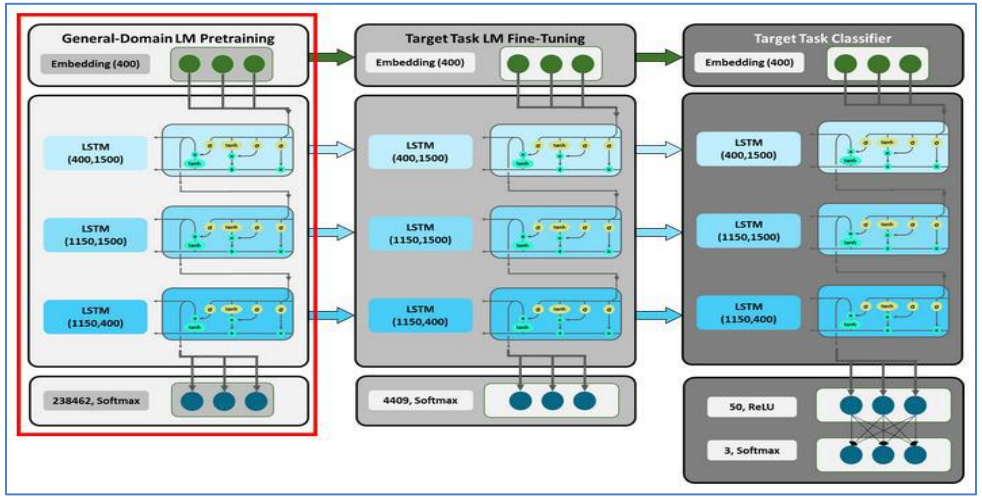
- ***C***: Es un parámetro que controla la "regularización", por la cual se intenta controlar el sobre entrenamiento, y que también se aplica en otros algoritmos basados en regresión.
- ***Multi_class***: Si la opción es "ovr" se aplica un algoritmo para encontrar valores binarios. Aplicando sin embargo la opción "multinomial" se aplica una distribución entre los valores posibles, aunque el resultado final sea 0 ó 1.

14. Modelado (*ULMFiT*)

- **ULMFiT** es un clasificador que se engloba dentro de la nueva ola de algoritmos basado en redes neuronales, y en este caso más concretamente en **"*Transfer Learning*"**. Este tipo de algoritmos se basa en la reutilización de conocimiento adquirido por otros modelos pre-entrenados.

  Algunas ventajas de este tipo de algoritmos:
  - ✓ Reaprovecha conocimiento.
  - ✓ Robustez. Se parte de unos modelos que ya funcionan.
  - ✓ Generalizan mejor. Parten de modelos ya ajustados que enriquecen los textos del contexto a aplicar.





Language Model pre-training. Pre-entrenamiento con un *dataset* muy grande pero genérico. A partir de aquí el modelo es capaz de predecir, con un grado de acierto, la siguiente palabra en una secuencia.

Language Model Fine-Tuning. Sería parecido al paso anterior pero enfocado al texto de estudio. En este caso se "congelan" y "descongelan" capas de la red a medida que los datos convergen.

Text classifier Fine-tuning. Se realiza la clasificación del texto. En nuestro caso concreto, clasificación de análisis de sentimiento. El "tuning" seria parecido al del paso anterior por capas.

15. Resultados (*ULMFiT*)

	Test#	Learning Rate	Momentum	Layers unfroz./Epochs	Accuracy(%)
Language Model	#1	6,31E-07	0,95-0,85	0/1	20,91%
	#2	2,29E-06	0,8-0,7	0/1	21,03%
		3,31E-04	0,9-0,8	*All*/2	27,11% 27,76%
	#3	1,58E-02	0,8-0,7	0/1	25,19%
		3,98E-06	0,8-0,7	*All*/1	25,48%
	#4	1.45E-05	0,8-0,7	0/1	21,67%
		1.74E-03	0,8-0,7	*All*/1	26,87%
Classifier	#1	1,45E-03	0,9-0,8	1/1	89,60%
		6,31E-07	0,9-0,8	2/3	89,66% 89,26% 89,20%
		6,31E-07	0,9-0,8	3/1	89,70%
		6,31E-07	0,8-0,7	*All*/1	92,42%
	#2	1.32E-02	0,8-0,7	1/1	88,36%
		1,32E-06	0,8-0,7	2/3	89,10% 89,34% 89,70%
		1,58E-06	0,8-0,7	3/1	91,32%
	#3	*slice*(1E-02/(2,6**4), 1E-02)	0,8-0,7	1/1	87,70%
		slice(1E-02/(2,6**4), 1E-02)	0,8-0,7	2/1	91,48%
		slice(5E-03/(2,6**4), 5E-03)	0,8-0,7	3/1	92,56%
		slice(1E-03/(2,6**4), 1E-03)	0,8-0,7	*All*/1	92,58%
	#4	6.92E-04	0,8-0,7	1/1	88,66%
		6.31E-07	0,8-0,7	2/1	89,26%
		slice(5E-03/(2.6**4), 5E-03)	0,8-0,7	3/1	92,54%
		slice(1E-03/(2.6**4), 1E-03)	0,9-0,8	*All*/1	92,86%

Parametrizaciones

Parameters

- ***Learning Rate***: Es un indicador basado en un numérico real que marca la tasa de aprendizaje de las capas. Se puede fijar o bien pasar un intervalo de valores utilizando la función “slide”.
- ***Momentum***: Está pensado para acelerar los procesos de aprendizaje, aplicando una tasa de porcentaje de cálculos. Se especifica un intervalo que en nuestro caso ha ido entre el 95 y el 70%, para no generar demasiadas inestabilidades.
- ***Layers unfroz. /Epochs***: Número de capas “descongeladas” (activas) / Número de pasadas en cada cálculo. Cuantas más pasadas más puede ajustar los resultados pero en más tiempo.

Universitat Oberta de Catalunya

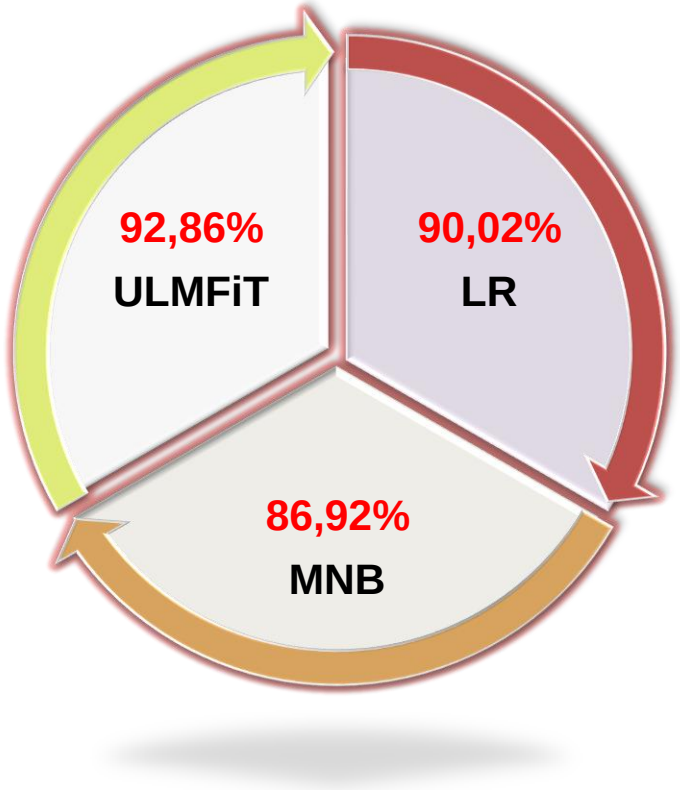
25 anys

¡TEST!

16. Conclusiones y líneas de trabajo a futuro

Conclusiones

- Obtención de resultados con **porcentajes de acierto entorno al 90%** en los tres algoritmos probados.
- En la comparación entre algoritmos más clásicos (LR y MNB) y de última generación (ULMFiT), no ha habido grandes diferencias, sin embargo **ULMFiT ha sido el que mejores resultados** ha obtenido. Esto, unido a su potencial, nos hace declararlo como **ganador** en esta comparativa, pese a que hoy en día requiere de más recursos.
- Lecciones aprendidas: Se ha observado una gran dependencia al *dataset* y al contexto de las opiniones al obtener los modelos. **Los resultados con texto libre no eran todo lo concluyentes que cabría esperar**.



Líneas de trabajo a futuro

- Utilización de diferentes *datasets* en contextos distintos.
- Utilización de textos en idioma castellano.

Universitat Oberta de Catalunya

25 anys

¡GRACIAS!