

Procesamiento de Lenguaje Natural

Apunte de Clases (Borrador)

Felipe Bravo Márquez

Felipe Bravo Márquez

Ilustración Portada por Paulette Filla

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN, UNIVERSIDAD DE CHILE

GITHUB.COM/DCCUCHILE/CC6205

Apuntes de clases del curso de Procesamiento de Lenguaje Natural de la Universidad de Chile.

El formato del apunte fue tomado del template de Jasmine Hao.

Borrador, 1 de abril de 2024



Índice general

0.1	Evaluación	3
0.1.1	Matriz de Confusión	4
0.1.2	Métricas de Desempeño	4
0.1.3	Una Medida Combinada: Medida F	5
0.1.4	Conjuntos de Prueba de Desarrollo ("Devsets")	6
0.1.5	Validación Cruzada: Múltiples Divisiones	6
0.2	Conjuntos de entrenamiento, prueba y validación	7
0.2.1	Matriz de Confusión para clasificación de 3 clases	8

0.1 Evaluación

La evaluación juega un papel crucial en la construcción de sistemas de PLN basados en aprendizaje automático. Al construir estos sistemas, es común llevar a cabo comparaciones y experimentos con diversos modelos durante el proceso de entrenamiento, con el fin de elegir el modelo más idóneo para su posterior implementación en producción. Generalmente, adoptamos un enfoque cuantitativo para evaluar modelos, contrastando las predicciones realizadas con las predicciones esperadas, y resumiendo estos resultados mediante métricas de evaluación. Este enfoque cuantitativo se prefiere debido a que un enfoque cualitativo o de inspección manual no resulta escalable. Ilustremos este proceso de evaluación con un ejemplo.

■ **Ejemplo 0.1** Supongamos un problema de clasificación binaria, para la cual queremos comparar dos modelos de clasificación m_1 y m_2 . Probamos los modelos sobre un conjunto de datos de prueba y obtenemos las predicciones señaladas en la Tabla 1.

Se puede observar que tanto el modelo m_1 como el modelo m_2 presentan errores en sus predicciones al compararse con las categorías reales de los ejemplos (columna clase), también conocidas como “gold labels” o “ground truth”. Sin embargo, ¿cómo determinamos cuál modelo es mejor? Para abordar esta pregunta, resulta conveniente calcular métricas de desempeño, las cuales se derivan al contrastar las predicciones de un modelo con sus salidas esperadas. ■

Texto	Clase	m_1	m_2
no estoy bien	negativo	negativo	positivo
al fin!	positivo	positivo	negativo
que pena	negativo	negativo	negativo
feliz	positivo	positivo	positivo
no quiero	negativo	positivo	negativo
jamás!	negativo	negativo	negativo
buenas ideas	positivo	positivo	negativo
chao contigo	negativo	positivo	negativo

Cuadro 1: Sentimiento de texto y predicciones realizadas por dos modelos distintos.

0.1.1 Matriz de Confusión

La mayoría de las métricas de evaluación para problemas de clasificación se derivan de una matriz de confusión (véase Tabla 2), la cual representa una tabla de contingencia entre las predicciones de un modelo y las salidas esperadas en un conjunto de prueba. Para problemas de clasificación binaria, esta matriz consta de 4 componentes:

1. **Verdadero Positivo (VP)**: cuando el modelo predice correctamente la clase positiva, coincidiendo con la salida esperada.
2. **Verdadero Negativo (VN)**: cuando el modelo predice correctamente la clase negativa, coincidiendo con la salida esperada.
3. **Falso Negativo (FN)**: cuando el modelo predice incorrectamente la clase negativa, siendo la salida esperada positiva.
4. **Falso Positivo (FP)**: cuando el modelo predice incorrectamente la clase positiva, siendo la salida esperada negativa.

Es importante destacar que la definición de qué constituye la clase positiva es arbitraria en un problema de clasificación; generalmente se refiere a la clase que se desea detectar. En el caso de la clasificación de sentimientos, es una coincidencia que la clase positiva se asocie con el sentimiento “positivo”. Esta consideración es fundamental, ya que el valor de algunas métricas de desempeño (como precisión, recall y F_1) depende de la elección de la clase positiva.

	Modelo Positivo	Modelo Negativo
Real Positivo	Verdadero Positivo (VP)	Falso Negativo (FN)
Real Negativo	Falso Positivo (FP)	Verdadero Negativo (VN)

Cuadro 2: Matriz de confusión para un problema de clasificación binaria.

■ **Ejemplo 0.2** Veamos como serían las matrices de confusión para los modelos m_1 y m_2 del ejemplo anterior. Primero procedemos a categorizar cada predicción de ambos modelos según una de las cuatro categorías señaladas (VP,VN,FP,FN).

Luego, agregamos dichos conteos en una matriz de confusión para cada modelo.

Clase esperada	m1	Resultado m1	m2	Resultado m2
negativo	negativo	VN	positivo	FP
positivo	positivo	VP	negativo	FN
negativo	negativo	VN	negativo	VN
positivo	positivo	VP	positivo	VP
negativo	positivo	FP	negativo	VN
negativo	negativo	VN	negativo	VN
positivo	positivo	VP	negativo	FN
negativo	positivo	FP	negativo	VN

Cuadro 3: Tipos de predicciones de m_1 y m_2 .

0.1.2 Métricas de Desempeño

Recall (exhaustividad) (también conocido como **Sensibilidad** o **Tasa de Verdaderos Positivos**):

$$\text{Recall} = \frac{VP}{VP + FN}$$

Precisión:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Accuracy (exactitud):

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + VN + FN}$$

¿Por qué no usamos la exactitud como nuestra métrica?

Imagina que vimos 1 millón de tweets:

- 100 de ellos hablaban sobre Delicious Pie Co.
- 999,900 hablaban de otra cosa.

Podríamos construir un clasificador tonto que simplemente etiquete todos los tweets como "no sobre pasteles":

- ¡¡¡Obtendría una exactitud del 99.99 %!!! ¡¡¡Wow!!!
- ¡Pero sería inútil! ¡No devuelve los comentarios que estamos buscando!

Por eso usamos precisión y recall en su lugar.

Precisión mide el porcentaje de elementos que el sistema detectó (es decir, los elementos que el sistema etiquetó como positivos) que son realmente positivos (según las etiquetas de oro humanas).

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

Recall mide el porcentaje de elementos que el sistema identificó correctamente de todos los elementos que deberían haber sido identificados.

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

Considera nuestro clasificador tonto de pasteles que simplemente etiqueta nada como "sobre pasteles".

- Exactitud = 99.99 % (etiqueta correctamente la mayoría de los tweets como no relacionados con pasteles)
 - Recall = 0 (no detecta ninguno de los 100 tweets relacionados con pasteles)
- La precisión y el recall, a diferencia de la exactitud, enfatizan los verdaderos positivos:
- Se centran en encontrar las cosas que se supone que debemos buscar.

0.1.3 Una Medida Combinada: Medida F

La medida F es un número único que combina la precisión (P) y el recall (R), definida como:

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

La medida F, definida con el parámetro β , pondera diferencialmente la importancia del recall y la precisión.

- $\beta > 1$ favorece al recall
- $\beta < 1$ favorece a la precisión

Cuando $\beta = 1$, la precisión y el recall son iguales, y tenemos la medida F_1 equilibrada:

$$F_1 = \frac{2PR}{P + R}$$

0.1.4 Conjuntos de Prueba de Desarrollo ("Devsets")

- Para evitar el sobreajuste y proporcionar una estimación más conservadora del rendimiento, comúnmente utilizamos un enfoque de tres conjuntos: conjunto de entrenamiento, conjunto de desarrollo y conjunto de prueba.



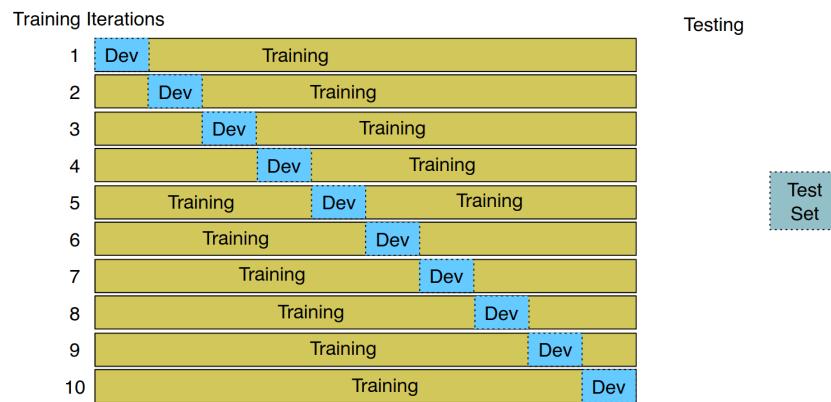
- **Conjunto de entrenamiento:** Se utiliza para entrenar el modelo.
 - **Conjunto de desarrollo:** Se utiliza para ajustar el modelo y seleccionar los mejores hiperparámetros.
 - **Conjunto de prueba:** Se utiliza para informar el rendimiento final del modelo.
- Este enfoque garantiza que el modelo no esté ajustado específicamente al conjunto de prueba, evitando el sobreajuste.
 - Sin embargo, crea una paradoja: queremos la mayor cantidad de datos posible para el entrenamiento, pero también para el conjunto de desarrollo.
 - ¿Cómo dividimos los datos?

0.1.5 Validación Cruzada: Múltiples Divisiones

La validación cruzada nos permite utilizar todos nuestros datos para el entrenamiento y la prueba sin tener un conjunto de entrenamiento, conjunto de desarrollo y conjunto de prueba fijos. Elegimos un número k y dividimos nuestros datos en k subconjuntos disjuntos llamados pliegues. En cada iteración, uno de los pliegues se selecciona como conjunto de prueba mientras que los $k - 1$ pliegues restantes se utilizan para entrenar el clasificador.

Calculamos la tasa de error en el conjunto de prueba y repetimos este proceso k veces. Finalmente, promediamos las tasas de error de estas k ejecuciones para obtener una tasa de error promedio.

Por ejemplo, la validación cruzada de 10 pliegues implica entrenar 10 modelos con el 90 % de los datos y probar cada modelo por separado. Las tasas de error resultantes se promedian para obtener la estimación final del rendimiento. Sin embargo, la validación cruzada requiere que todo el corpus sea ciego, lo que impide examinar los datos para sugerir características o comprender el comportamiento del sistema. Para abordar esto, se crea un conjunto de entrenamiento y un conjunto de prueba fijos, y se realiza la validación cruzada de 10 pliegues dentro del conjunto de entrenamiento. La tasa de error se calcula convencionalmente en el conjunto de prueba.



0.2 Conjuntos de entrenamiento, prueba y validación

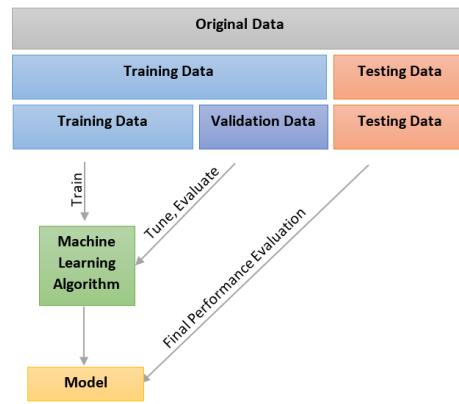
Cuando entrenamos un modelo, nuestro objetivo es producir una función $f(\vec{x})$ que mapee correctamente las entradas \vec{x} a las salidas \hat{y} según lo evidenciado por el conjunto de entrenamiento. La evaluación del rendimiento en los datos de entrenamiento puede ser engañosa, ya que nuestro objetivo es entrenar una función capaz de generalizar a ejemplos no vistos. Una forma común de abordar esto es dividir el conjunto de entrenamiento en subconjuntos de entrenamiento y prueba (80 % y 20 % respectivamente). Se entrena el modelo en el subconjunto de entrenamiento y se calcula la precisión en el subconjunto de prueba.

Sin embargo, este enfoque tiene una limitación. En la práctica, a menudo se entranan varios modelos, se comparan sus calidades y se selecciona el mejor. Si se selecciona el mejor modelo en función de la precisión en el subconjunto de prueba, se obtendrá una estimación excesivamente optimista de la calidad del modelo. No se sabe si la configuración elegida del clasificador final es buena en general o simplemente es buena para los ejemplos particulares en los subconjuntos de prueba.

La metodología aceptada es utilizar una división de tres vías de los datos en conjuntos de entrenamiento, validación (también llamado desarrollo) y prueba¹. Esto proporciona dos conjuntos apartados: un conjunto de validación (también llamado conjunto de desarrollo) y un conjunto de prueba. Todos los experimentos, ajustes, análisis de errores y selección de modelos deben realizarse

basados en el conjunto de validación. Luego, una única ejecución del modelo final sobre el conjunto de prueba proporcionará una buena estimación de su calidad esperada en ejemplos no vistos. Es importante mantener el conjunto de prueba lo más limpio posible, realizando la menor cantidad de experimentos posible en él. Incluso algunos defienden que no se deben mirar siquiera los ejemplos en el conjunto de prueba, para evitar sesgar el diseño del modelo.

¹Un enfoque alternativo es la validación cruzada, pero no se escala bien para entrenar redes neuronales profundas.



0.2.1 Matriz de Confusión para clasificación de 3 clases

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	precision _u = $\frac{8}{8+10+1}$
	normal	5	60	50	precision _n = $\frac{60}{5+60+50}$
	spam	3	30	200	precision _s = $\frac{200}{3+30+200}$
		recall _u = $\frac{8}{8+5+3}$	recall _n = $\frac{60}{10+60+30}$	recall _s = $\frac{200}{1+50+200}$	

Cómo combinar métricas binarias (Precisión, Recall, F_1) de más de 2 clases para obtener una métrica única:

- Macro-promedio:
 - Calcular las métricas de rendimiento (Precisión, Recall, F_1) para cada clase individualmente.
 - Promediar las métricas en todas las clases.
- Micro-promedio:
 - Recopilar las decisiones para todas las clases en una matriz de confusión.
 - Calcular la Precisión y el Recall a partir de la matriz de confusión.

Class 1: Urgent		Class 2: Normal		Class 3: Spam		Pooled									
true	true	true	true	true	true	true	true								
system	urgent	8	11	system	normal	60	55	system	spam	200	33	system	yes	268	99
system	not	8	340	system	not	40	212	system	not	51	83	system	no	99	635
		$\text{precision} = \frac{8}{8+11} = .42$		$\text{precision} = \frac{60}{60+55} = .52$		$\text{precision} = \frac{200}{200+33} = .86$		$\text{microaverage precision} = \frac{268}{268+99} = .73$							
		$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = .60$													

¹Fuente: <https://www.codeproject.com/KB/AI/1146582/validation.PNG>



[McCallum et al., 1998] McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI.

[Mosteller and Wallace, 1963] Mosteller, F. and Wallace, D. L. (1963). Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309.