INCO - FING

WebIR
Clasficación de noticias

Lucas Micol
Bruno Stecanella
Felipe Osimani
Gastón Rodríguez
21 de noviembre de 2016

1. Introducción

En el siguiente documento se describe una solución al clásico problema de clasificar noticias. Esta consiste en la utilización de una API otorgada por MonkeyLearn, la cual es el resultado de entrenar un modelo de aprendizaje automático con un gran conjunto de noticias ya clasificadas. Se han realizado mediciones sobre los resultados obtenidos y resultaron altamente satisfactorios.

2. Problema

Surge el problema frente a la necesidad de navegar por las noticias publicadas en la página web de El País `elpais.com.uy`, y encontrar que las mismas no se encuentran categorizadas de una forma amigable para el usuario. Las categorías son muy pocas y muy genéricas, haciendo que el usuario se pierda fácilmente y probablemente decida ir a otro portal de noticias más organizado.

3. Motivación

La motivación para llevar adelante este proyecto surge como consecuencia de intentar scrapear comentarios de Facebook (junto con sus respectivas respuestas y reacciones) de noticias categorizadas de la página de SubrayadoHD y entrenar un modelo utilizado por un algoritmo de aprendizaje automático para detectar "nivel de popularidad". Luego se scrapearían comentarios de noticias de la misma categoría sin respuestas ni votos asignados por otros usuarios de la página de El País, para medir su grado de popularidad.
Al realizar la etapa de scrapear comentarios de Facebook y entrenar el modelo aparece el problema de obtener noticias categorizadas de la página de El País, que en pocos minutos se hizo evidente la desorganización de ésta y la dificultad de obtener buenos resultados. Todo esto hizo desistir lo antes planteado y cambiar la dirección en la que iba encaminado el proyecto. Se decidió que frente al problema real de categorizar noticias de uno de los diarios uruguayos más importantes, se podría diseñar una solución.
Vale la pena mencionar que los resultados obtenidos luego de entrenar al modelo con los comentarios de Facebook no fueron alentadores, y fue éste el ultimo y decisivo motivo para pivotear el proyecto a pesar de haber tenido un alto grado de avance en la idea original. Esto se debe a que es muy difícil, incluso para un ser humano, predecir la popularidad de determinado comentario. Hay muchas variables presentes (grupo de personas que comenta, importancia de la noticia, forma de escribir del periodista) y además muchas veces un comentario popular no se diferencia significativamente de uno impopular.

4. Enfoque de la solución

La solución consiste en dos grandes etapas, la primera es obtener una gran cantidad de noticias de la página de Subrayado con sus categorías. La segunda es realizar el entrenamiento

del modelo utilizando MonkeyLearn: esto implica entrenar el modelo con los datos obtenidos, realizar las métricas de desempeño a partir de cross-validation. La última etapa es obtener métricas de desempeño con datos etiquetados a mano de noticias de la página de El País.

Con todo esto se realiza una presentación de métricas obtenidas con dos diferentes algoritmos de aprendizaje automático (Support Vector Machines y Naive Bayes), matriz de confusión y palabras clave por tipo de categoría de noticia

4.1. Noticias categorizadas y entrenamiento del modelo

Para el entrenamiento del modelo se utilizaron noticias clasificadas en 7 categorias, que son las que utiliza Subrayado en su página web, las mismas son: Policiales, Internacionales, Deportes, Sociedad, Espectáculo, Política y Curiosidades. Las noticias utilizadas fueron todas desde el 9 de Enero del 2013 hasta Octubre del 2016.

El entrenamiento del modelo fue prácticamente imperceptible en dificultad, dado que la Api que brinda MonkeyLearn es muy intuitiva y organizada. El mismo demoró aproximadamente 10 minutos para cada uno de los diferentes algoritmos de aprendizaje.

5. Herramientas

Como herramientas principales fueron utilizadas la ApiGraph publicada por Facebook para scrapear comentarios, un scraper utilizando la librería de Python Scrapy, y la API de MonkeyLearn.

5.1. ApiGraph - Facebook

Partiendo del primer problema que se decidió afrontar, fue necesario obtener información relacionada a los post de SubrayadoHD en Facebook junto con sus comentarios y ciertas características. Para ello se introdujo en los conceptos de la API que provee Facebook para acceder a sus recursos tal como información de usuarios, post, likes, comentarios, etc.

Dicha API se modela como un grafo, sus nodos pueden ser root o non-root y representan los objetos tal como usuarios, posts, comentarios. Por otro lado las aristas representan las conexiones entre los nodos, a modo de ejemplo, los post de un usuario tienen aristas a sus post, y los post a sus comentarios, además todos lo nodos tienen campos, que a la hora de hacer una solicitud a uno de estos es necesario conocerlos y especificarlos correctamente para obtener los datos deseados de cada objeto.

La interacción necesaria para realizar la extracción de los post junto con sus respectivos comentarios y reacciones de SubrayadoHD requirió realizar solicitudes de lectura a la arista *posts* del nodo *SubrayadoHD* con los siguientes campos: *id, story, message, link, comments, reactions, likes* y como metadato fue necesario especificar un limite de paginado y el formato json de los resultados.

Los resultados obtenidos fueron procesados con el fin de homogenizar los datos y poder almacenarlos de manera tal que sea posible emplear dicha información para entrenar el modelo de aprendizaje automático.

Luego que el proyecto cambio su rumbo, esta parte del mismo terminó no siendo utilizada.

5.2. SCRAPY

Scrapy es una librería open source de Python para scrapear información de la web de forma rápida y fácil.

Se utilizó en la creación de dos spiders para obtener 17.916 noticias ya categorizadas del archivo de la página de Subrayado. La solución resultó ser muy rápida, pudiendo terminar la descarga en menos de 15 minutos.

La navegación dentro del sitio resultó ser sencilla, pues el archivo de Subrayado expone todas las noticias con un ID incremental en la URL.

También se utilizó para obtener noticias de la página de El País para ser categorizadas a mano.

5.3. MONKEYLEARN

MonkeyLearn es una plataforma web de Machine Learning que provee facilidades de aprendizaje automático como servicio. Se pueden crear clasificadores de texto utilizando la interfaz web: el usuario define la taxonomía y sube el conjunto de entrenamiento, y MonkeyLearn entrena, produce métricas de evaluación del modelo, y expone una API para clasificar textos nunca vistos por el modelo.

Es posible elegir los parámetros de entrenamiento del modelo y el algoritmo de aprendizaje automático a utilizar. Las métricas que la herramienta calcula son obtenidas mediante cross-validation de tamaño 4: el conjunto de entrenamiento se parte en 4 conjuntos, utilizando 3 de ellos se entrena y con el restante se evalúa. Esto se hace para todas las combinaciones de conjuntos, y se promedian los resultados de evaluación.

Esta plataforma fue utilizada para entrenar un clasificador que, dado el texto de una noticia, la categoriza en una de las categorías del sitio de Subrayado como se describe más adelante. Para entrenar este modelo se utilizaron las noticias (y la categoría de cada noticia) obtenidas con Scrapy del sitio de Subrayado.

6. EVALUACIÓN Y RESULTADOS

Utilizando los artículos categorizados de la página de Subrayado fue creado un clasificador de MonkeyLearn que dado un texto le asigna una categoría. Esto se pudo hacer con un gran volumen de datos porque de la web fue obtenido directamente cada artículo con su categoría.

El módulo creado junto a todas sus métricas puede ser accedido en `https://app.monkeylearn.com/main/classifiers/cl_VCykxryo/`

La pestaña *Sandbox* tiene el módulo entrenado con Support Vector Machines, y *Live* tiene el módulo entrenado con Naive Bayes.

Fue elaborado también un conjunto de datos de prueba, independiente del conjunto de entrenamiento, en base a noticias de la página del diario El País. Estas noticias fueron categorizadas a mano por el equipo con las categorías de Subrayado que le corresponde a cada noticia. Luego, fueron clasificadas por el módulo de MonkeyLearn y se compararon las etiquetas. Con esto se obtuvieron más métricas para evaluar el desempeño del módulo.

6.1. Parámetros de entrenamiento

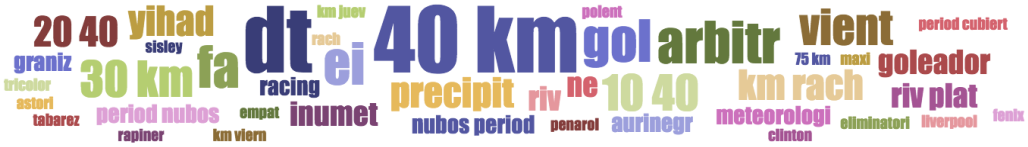
Para entrenar el modelo se utilizaron los siguientes parámetros:

- Idioma: Español. Esto afecta el parser que se usa en el texto, el stemmer, los stopwords, y demás cosas que sean dependientes del idioma.
- Rango de n-gramas: Unigramas y Bigramas. Esto quiere decir que el modelo tomará features que sean una palabra sola o dos palabras seguidas, pero no tres o más.
- Cantidad máxima de features: 15,000
- Stemming: Sí. Esto quiere decir que palabras como *partido* y *partidario* se convierten ambas en *part*
- Stopwords: Sí. Los stopwords son palabras como: *de, la, que, para, con, no, una, su, al, lo, como, más, pero, sus, le, ya, o.*
- Algoritmo: Se probó con Support Vector Machines y Naive Bayes, los resultados de cada uno se detallan más abajo.

6.2. Palabras clave

Un dato interesante de analizar en cuanto a las noticias uruguayas son las palabras clave de cada categoría, mostradas a continuación:
Observación: como el algoritmo usa stemming muestra los raíz de las palabras clave.

- Palabras clave de todas las noticias:



Observación: se puede observar que muchas de las palabras clave son de noticias del clima, ya que abarcan una gran cantidad del total y tienen un vocabulario muy específico.

- Policiales:

- Sociedad:

- Espectáculos:

- Curiosidades:

- Deportes:

- Internacionales:

- Política:

6.3. Algoritmo: Naive Bayes

6.3.1. Cross-Validation

Usando el cross-validation de MonkeyLearn (de tamaño 4) se obtuvo accuracy de 83 %. Las métricas de evaluación se detallan a continuación:

Métricas

Categoría	Precision	Recall
Curiosidades	0.64	0.73
Deportes	0.91	0.95
Espectaculos	0.75	0.79
Internacionales	0.83	0.80
Policiales	0.86	0.91
Politica	0.82	0.83
Sociedad	0.85	0.78

Matriz de confusión

	Cur.	Dep.	Esp.	Int.	Polic.	Polit.	Soc.
Curiosidades	495	29	20	118	12	5	20
Deportes	23	1265	4	24	13	2	12
Espectaculos	29	5	260	15	3	3	12
Internacionales	161	32	34	1397	51	34	37
Policiales	13	14	2	22	1697	17	65
Politica	6	6	4	27	25	1289	185
Sociedad	48	38	20	53	151	226	1977

6.3.2. TESTING SET

La accuracy que se obtuvo evaluando con el testing set es de 74 %

Métricas

Categoría	Precision	Recall	F-Score	Cantidad
Curiosidades	0.61	0.79	0.69	14
Deportes	0.71	1.00	0.83	17
Espectaculos	0.92	0.46	0.62	26
Internacionales	0.73	0.62	0.67	13
Policiales	0.90	0.82	0.86	11
Politica	0.74	0.71	0.72	35
Sociedad	0.72	0.83	0.77	41

Matriz de confusión

	Cur.	Dep.	Esp.	Int.	Polic.	Polit.	Soc.
Curiosidades	11	0	0	1	0	0	2
Deportes	0	17	0	0	0	0	0
Espectaculos	5	5	12	2	0	0	2
Internacionales	2	0	1	8	0	2	0
Policiales	0	1	0	0	9	1	0
Politica	0	0	0	0	1	25	9
Sociedad	0	1	0	0	0	6	34

6.4. OBSERVACIONES

- Las noticias de la categoría sociedad y política fueron las más erradas. Esto se desprende que hay muchas noticias que pueden ser considerados tanto de una como de la otra, ya que son dos áreas de la información fuertemente relacionadas.

6.5. ALGORITMO: SUPPORT VECTOR MACHINES

6.5.1. CROSS-VALIDATION

Usando el cross-validation de MonkeyLearn (de tamaño 4) se obtuvo accuracy de 84 %

Métricas

Categoría	Precision	Recall
Curiosidades	0.64	0.71
Deportes	0.91	0.94
Espectaculos	0.76	0.80
Internacionales	0.84	0.80
Policiales	0.87	0.93
Politica	0.82	0.84
Sociedad	0.86	0.79

Matriz de confusión

	Cur.	Dep.	Esp.	Int.	Polic.	Polit.	Soc.
Curiosidades	932	38	47	219	20	5	8
Deportes	08	2182	18	61	29	9	16
Espectaculos	64	8	487	27	2	3	8
Internacionales	337	37	43	2468	106	80	26
Policiales	22	21	4	30	3007	55	111
Politica	3	8	12	59	43	2413	217
Sociedad	134	49	83	146	327	589	3223

6.5.2. TESTING SET

La accuracy que se obtuvo evaluando con el testing set es de 75 %

Métricas

Categoría	Precision	Recall	F-Score	Cantidad
Curiosidades	0.63	0.86	0.73	14
Deportes	0.76	0.94	0.84	17
Espectaculos	0.88	0.58	0.70	26
Internacionales	0.70	0.54	0.61	13
Policiales	0.82	0.82	0.82	11
Politica	0.71	0.71	0.71	35
Sociedad	0.77	0.83	0.80	41

Matriz de confusión

	Cur.	Dep.	Esp.	Int.	Polic.	Polit.	Soc.
Curiosidades	12	0	0	1	0	0	1
Deportes	1	16	0	0	0	0	0
Espectaculos	6	3	15	1	0	0	1
Internacionales	0	1	1	7	0	3	1
Policiales	0	1	0	0	9	1	0
Politica	0	0	1	0	2	25	7
Sociedad	0	0	0	1	0	6	34

7. Conclusiones

Las métricas del conjunto de prueba son peores que las de cross-validation, pero son más similares al comportamiento del clasificador en un ambiente de producción, pues fue creado utilizando noticias de El País y no de Subrayado, que son las que se utilizarían en una eventual puesta en producción.

Tanto con cross-validation como con el conjunto de prueba independiente el modelo de SVM tiene mejores métricas que el de Naive Bayes, que era un resultado esperable.

Viendo la matriz de confusión, hay un solapamiento entre las categorías Política y Sociedad, lo cual era de esperar pues son similares en muchos aspectos. También hay confusión en Espectáculos, donde aparentemente se confunde con Curiosidades y Deportes.

Curiosidades es la categoría que peor funciona, con una precisión del 61 %. Una posible explicación de este resultado mirando los artículos es que es una categoría muy amplia con poca consistencia. Esto puede generar que el modelo no aprenda correctamente la categoría y luego clasifique incorrectamente.

Más allá de esos detalles, el clasificador funciona muy bien, con la mayoría de las noticias categorizadas correctamente. Más adelante, una página en producción que utilice este clasificador tendría muy buenos resultados.

8. Trabajo futuro

Gracias a los resultados obtenidos, sería posible desarrollar una aplicación web que permita categorizar cualquier tipo de noticias. En particular, un buscador de noticias por categorías de los diarios nacionales, donde un usuario puede priorizar de qué diario quiere obtener información y brindar alguna palabra clave para satisfacer su búsqueda. También se podría implementar un sistema de reaprendizaje"donde regularmente se toman noticias categorizadas y se reentrene el algoritmo categorizador para mantener el modelo al día, y obtener mejores resultados para artículos actuales.
Otro interesante proyecto sería un Addon para navegador que al entrar a una noticia del diario El País consulte a la API que se provee para clasificar y mostrar la categoría en tiempo real. De esta forma, un usuario podría agregar la información de categorías faltante al sitio web del diario.

Una último uso de todo este proyecto es su versatilidad para uso en cualquier otro proyecto de software. Al proveer una API que clasifica noticias uruguayas en tiempo real se puede utilizar para lo que el desarrollador necesite, teniendo un modelo de machine learning ya entrenado y listo para usar.

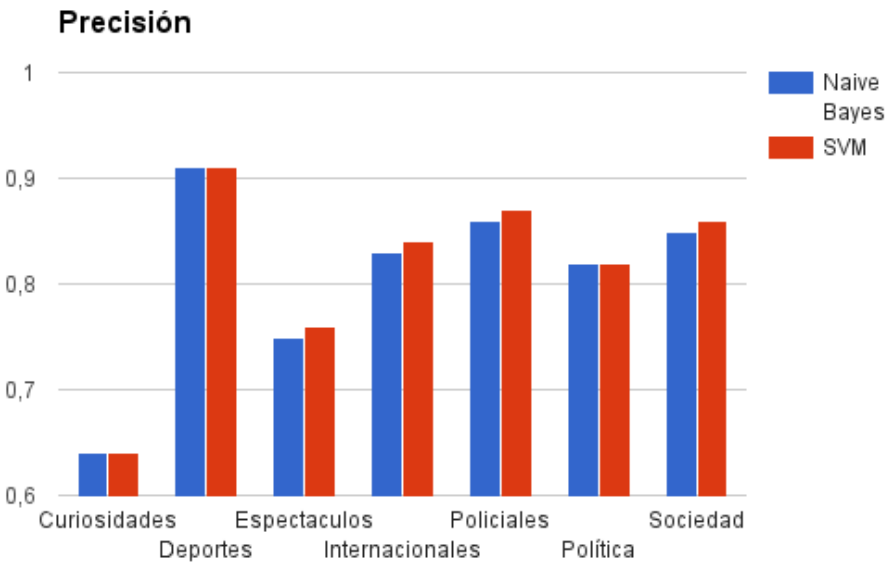
Como lección aprendida, este método, que es muy sencillo y fácil de replicar, se puede aplicar a cualquier lugar de la web donde haya páginas con contenidos similares donde una tenga categorías o metadatos que la otra no. Funciona tanto para categorización de artículos como para reviews de bienes y servicios de todo tipo. Se pueden analizar los resultados para obtener métricas que de otra forma serían imposibles de conocer.

9. Referencias

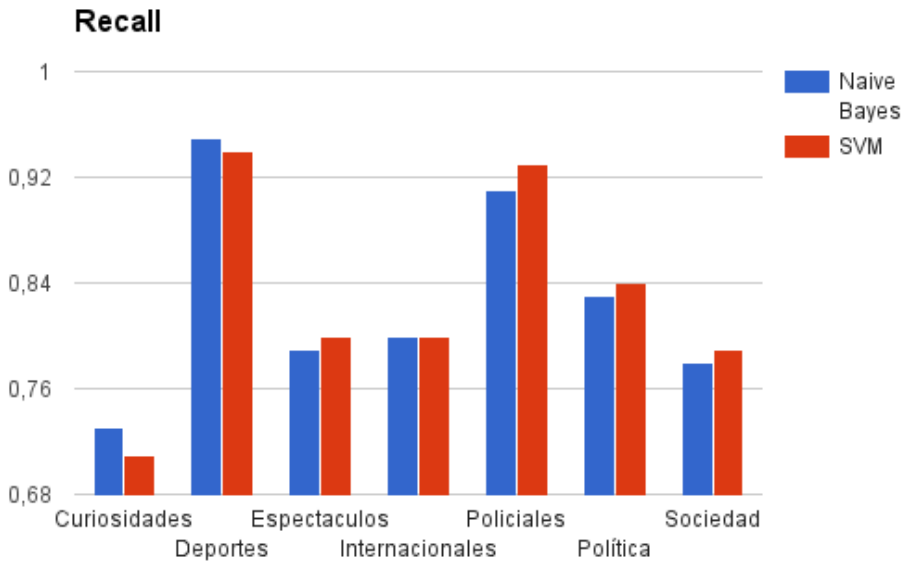
1. **Subrayado:** `https://subrayado.com.uy/`
2. **El País:** `https://www.elpais.com.uy/`
3. **ApiGraph Facebook:** `https://developers.facebook.com/docs/graph-api`
4. **Scrapy framework:** `https://scrapy.org/`
5. **MonkeyLearn:** `https://monkeylearn.com/`

10. Anexo

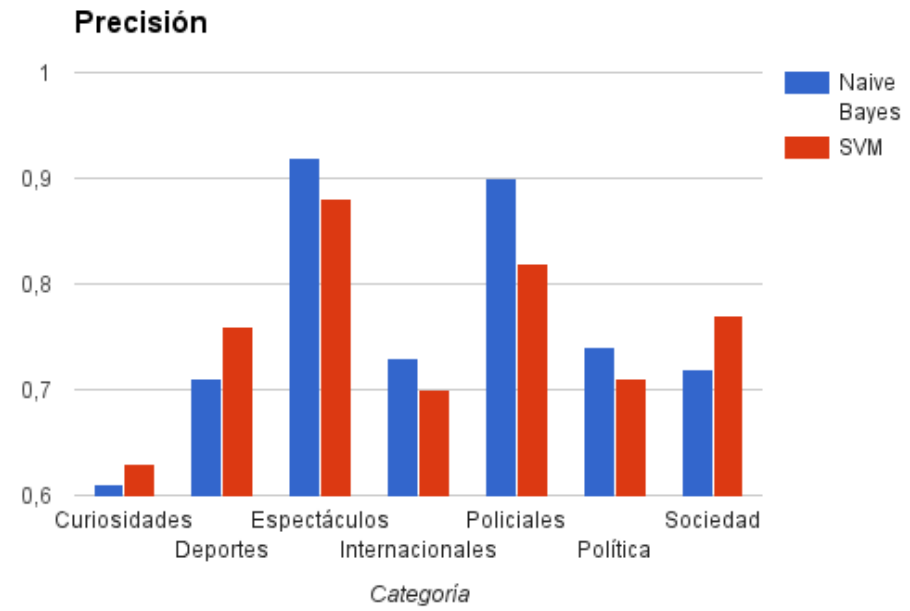
- Comparación de precisión Naive Bayes y SVM en Cross-Validation



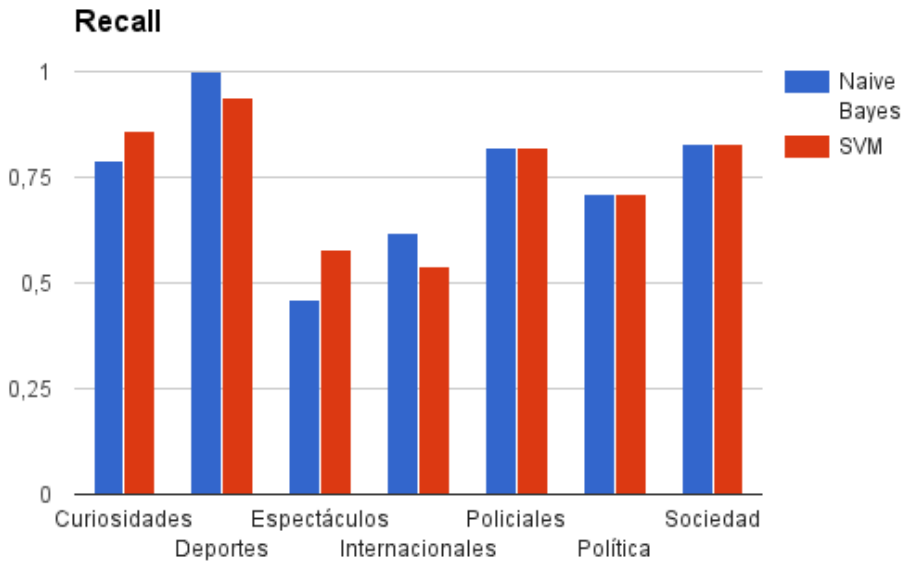
- Comparación de recall Naive Bayes y SVM en Cross-Validation



- Comparación de precisión Naive Bayes y SVM en Testing Set



- Comparación de recall Naive Bayes y SVM en Testing Set



- Comparación de F-Score Naive Bayes y SVM en Testing Set