

# Minería de Opinión en Twitter

---

Clasificación de Tweets y  
visualización de tendencias de  
opinión

# Objetivos

---

- Construir una herramienta que nos permita clasificar y visualizar en tiempo real tweets obtenidos utilizando Twitter Streaming API.
- Evaluar el modelo de clasificación obtenido utilizando las métricas Precision, Recall y F-Measure.
- Construir un clasificador simple que evalúe la polaridad de un tweet en términos de ocurrencias de palabras positivas y negativas.
- Comparar la performance de ambos modelos de clasificación.

# Sistema de Clasificación y Visualización en Tiempo Real

---

# Interfaz con el usuario

index.html x +

File | File:///home/leo/UNS/tests/classification\_interface/index.html

Visualización de clasificación de tweets según polaridad

Términos

Fecha Desde  
12/04/2018, 09:13 AM

Fecha Hasta  
12/06/2018, 09:13 AM

Real Time

Buscar Histórico

Referencias 33564

☒ Ninguno 938 de 13155

☒ Positivo 1342 de 13218

☒ Negativo 283 de 6780

☒ Neutral 13 de 411

Google

Map data ©2018 Google Terms of Use

Fecha	En Mapa	Sentimiento	Tweet	Tokens
12/4/2018, 10:13:25 AM	Si	positivo	Acaba de publicar una foto en Rosario, Santa Fe <a href="https://t.co/RX2yM20jbp">https://t.co/RX2yM20jbp</a>	acab,public,fot,rosari,sant,fe
12/4/2018, 10:13:34 AM	Si	ninguno	Un toque de misterio invade el ambiente con la luna en escorpio, las emociones llegan a su punto limite y estamos l... <a href="https://t.co/WEr1Ya4Ybf">https://t.co/WEr1Ya4Ybf</a>	toqu,misteri,invad,ambient,lun,escorpi,emocion,lleg,punt,limit,i,...
12/4/2018, 10:13:36 AM	Si	ninguno	#hermosos paisajes #naturalez en Paso de los Toros <a href="https://t.co/dD6AVv4Zai">https://t.co/dD6AVv4Zai</a>	hermos,paisaj,naturalez,pas,tor
12/4/2018, 10:14:06 AM	Si	positivo	Hermoso regalo de mi hermosa bvm87 Muchas gracias mi Amor. Te Amo. en Montevideo, Uruguay <a href="https://t.co/tncuclsnm0">https://t.co/tncuclsnm0</a>	hermos,regal,hermos,bvmm,much,graci,amor,amo,montevideo,uruguay
12/4/2018, 10:14:24 AM	Si	positivo	Acaba de publicar una foto en Tala, Uruguay <a href="https://t.co/bzue0697Uv">https://t.co/bzue0697Uv</a>	acab,public,fot,tal,uruguay
12/4/2018, 10:14:27 AM	Si	ninguno	Instagram challenge day 2: your breakfast! - como no especifica exactamente el desayuno de cual día me tomo la libe	instagram,challeng,day,your,breakfast,especif,exat,desayun,cual,dia

# Funcionalidades

---

- Búsqueda Histórico
  - Permite buscar tweets ya procesados en un rango de fechas.
- Búsqueda en Tiempo Real
  - Permite iniciar la lectura y procesamiento de tweets en tiempo real. La interfaz se actualiza periódicamente con los tweets procesados.
  - Búsqueda por términos. Antes de iniciar la búsqueda en tiempo real tenemos la posibilidad de ingresar términos para el filtrado de tweets.
    - Term1 Term2: filtra los tweets que contengan ambos términos sin importar el orden en el que aparezcan en el mensaje.
    - Term1, Term2: filtra los tweets que contengan al menos uno de los términos.

# Visualización de Tweets

---

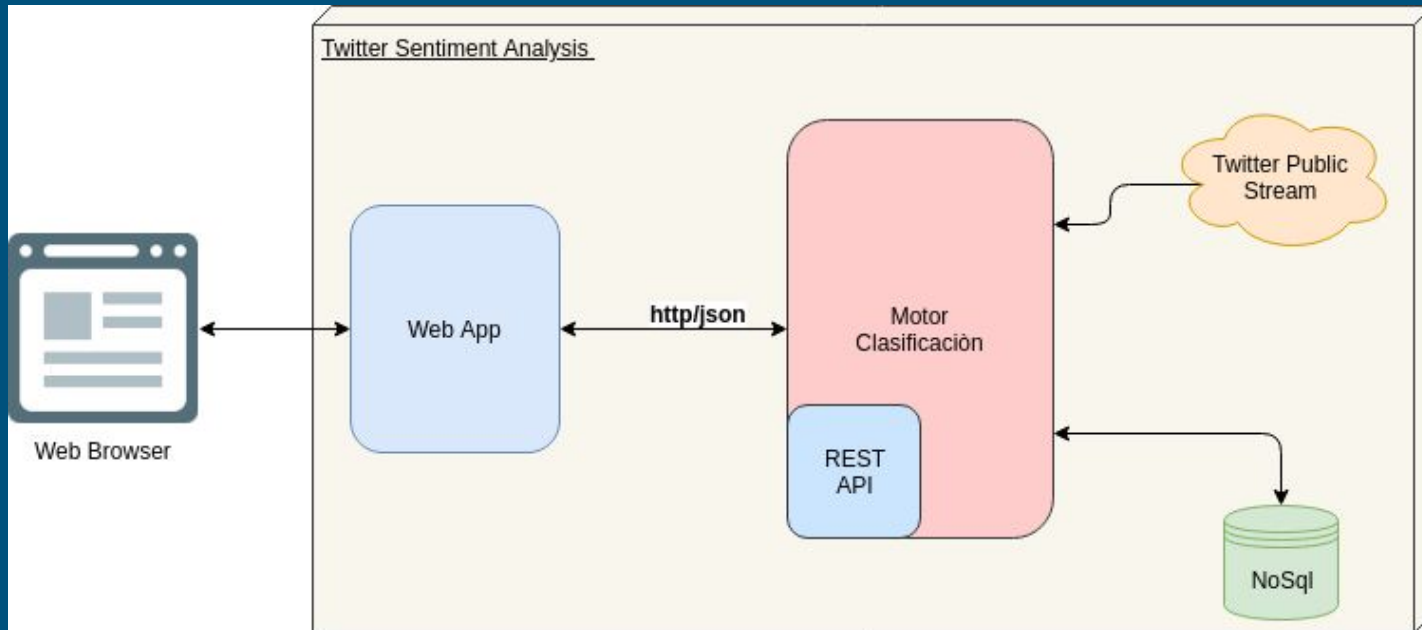
- Referencias

- El cuadro de referencias nos indica la cantidad de tweets procesados y la clase/polaridad a la que pertenecen.
- Por medio de los checkbox podemos filtrar los tweets que se despliegan como “círculos” en el mapa.

- Grilla de Tweets

- Nos deja ver los tweets procesados y al seleccionarlos resaltarlos en el mapa.

# Arquitectura del Sistema



# Tecnologías

---

- Base de datos NoSQL: **Mongodb** (<https://www.mongodb.com/>)
- Backend: escrito en Python (<https://www.python.org/>)
- Frontend: HTML + CSS + Javascript.
- Librería acceso a Twitter Streaming API: **Tweepy** (<http://www.tweepy.org/>)
- Machine Learning Library: **scikit-learn** (<https://scikit-learn.org>)
- REST API Framework: **falcon** (<https://falconframework.org/>)
- Web Application Libraries: **jquery** (<https://jquery.com/>) - **jsgrid** (<http://js-grid.com/>) - **Google Maps JavaScript API** (<https://cloud.google.com/maps-platform/>)



# Clasificación de Tweets

---

# TASS Corpus

---

- El modelo de clasificación generado fue entrenado utilizando este corpus.
- TASS (<http://www.sepln.org/workshops/tass/>): Taller de Análisis Semántico de la SEPLN.
- SEPLN (<http://www.sepln.org>): Sociedad Española para el Procesamiento del Lenguaje Natural.
  - La Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN) es una asociación científica sin ánimo de lucro con el objetivo de promover todo tipo de actividades relacionadas con el estudio del procesamiento de lenguaje natural.

# Formato de un Tweet en el Corpus

---

```
<tweet>
  <tweetid>142378325086715906</tweetid>
  <user>jesusmarana</user>
  <date>2011-12-02T00:03:32</date>
  <lang>es</lang>
  <sentiments>
    <polarity>
      <value>N</value>
    </polarity>
  </sentiments>
  <topics>
    <topic>politica</topic>
  </topics>
  <content>Portada 'Público', ....</content>
</tweet>
```

# Formato de un Tweet en el Corpus

---

- La polaridad asociada a cada tweet puede ser uno de los siguientes valores: N+ (muy negativo), N (negativo), NEU (neutral), P (positivo), P+ (muy positivo), NONE (no expresa sentimiento).
- En este trabajo se utilizan cuatro niveles de polaridad (ninguno, positivo, negativo y neutral).
- Mapeo de polaridades:
  - NONE → ninguno
  - P, P+ → positivo
  - N, N+ → negativo
  - NEU → neutral

# Pre-Procesamiento de Tweets

---

1. Stemming (lemmatization). Se utilizó SnowballStemmer incluído en NLTK.
2. Tratamiento de emoticones. Se consideran un token.
3. Tratamiento signos de puntuación. Se remueven los signos de puntuación.
4. Stop words. Se remueven stop words.
5. Lower case words. Se lleva a minúsculas todos los tokens.
6. Se eliminan las URLs y direcciones de E-Mails.
7. Referencias a usuarios, @user. Se eliminan las referencias a usuarios.
8. Hashtags. Se elimina el símbolo de hash, “#hashtag” se transforma en “hashtag”.
9. Reducción de longitud de las palabras. Los caracteres repetidos más de tres veces se contraen a tres, por ejemplo “hooooooooola” se transforma en “hoooola”.

# Pre-Procesamiento de Tweets

---

Ejemplo de Tweet pre-procesado:

```
|Tweet: Una de las mejores series que he visto!! #Merlí 🐼🐼🐼  
https://t.co/37...  
Tokens: mejor, seri, vist, merl, 🐼, 🐼, 🐼
```

# Generación de Feature Vectors

---

- Del pre-procesamiento de un tweet obtenemos los tokens que utilizamos para generar su vector de características, input al modelo de clasificación.
- Este vector tiene tantos componentes como tokens extraídos del Corpus.
- Se trabajaron con valores binarios  $[0, 1]$ . Si el token se encuentra presente en el tweet dicha componente se define con el valor 1 y de forma análoga, de no encontrarse, toma el valor 0.
- Se decidió este esquema bajo la hipótesis de que dada la naturaleza de los mensajes, 140 caracteres al momento en el que se generó el Corpus, no es importante la cantidad de ocurrencias de un término sino que ocurra.

# Generación del Modelo

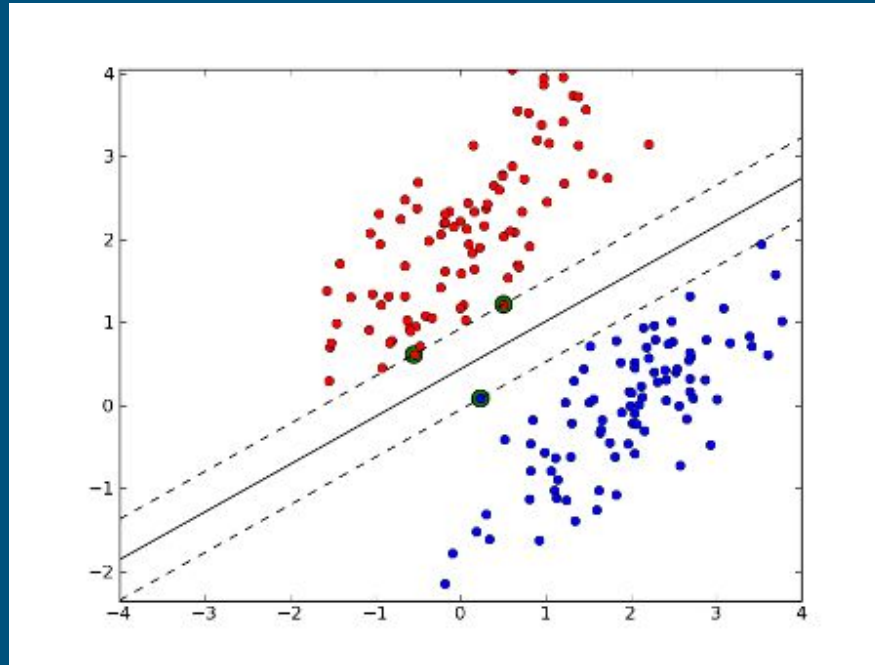
---

- El modelo de clasificación está basado en **Support Vector Machine**.
- SVM busca generar un Hiperplano que divida las Instancias en las clases correspondientes.
- El Hiperplano buscado es aquel que maximice su distancia a los vectores representativos de cada clase.
- Los vectores representativos (support vectors) de cada clase serán aquellos que estén más cerca del Hiperplano.



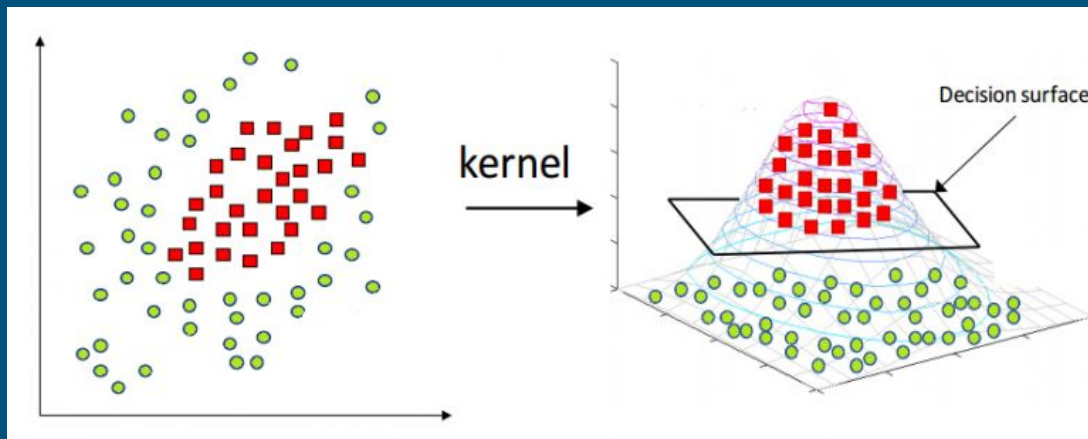
# Generación del Modelo

---



# SVM Kernels

- Cuando las instancias no son linealmente separables se utilizan funciones Kernel que buscan dimensionar los vectores en espacios superiores y así encontrar la separación buscada.



# Optimización de Parámetros

---

- SVM tiene como input diversos parámetros que modifican su comportamiento. Estos varían según la función Kernel que se esté usando.
- En este trabajo se utilizaron:
  - Linear Kernel
    - Parámetros: C
  - RBF Kernel
    - Parámetros: C, Gamma
- Para la optimización de estos parámetros se utilizó **Grid Search (scoring='f1\_weighted')** junto con **K-Fold Cross Validation (K = 5)**

# Evaluación de Desempeño

---

# Métricas

---

Para evaluar el desempeño de los modelos generados se utilizaron las métricas de Precision, Recall y F1-Measure. El 80 % del Corpus se utilizó para entrenar el clasificador y el 20 % restante para su validación.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

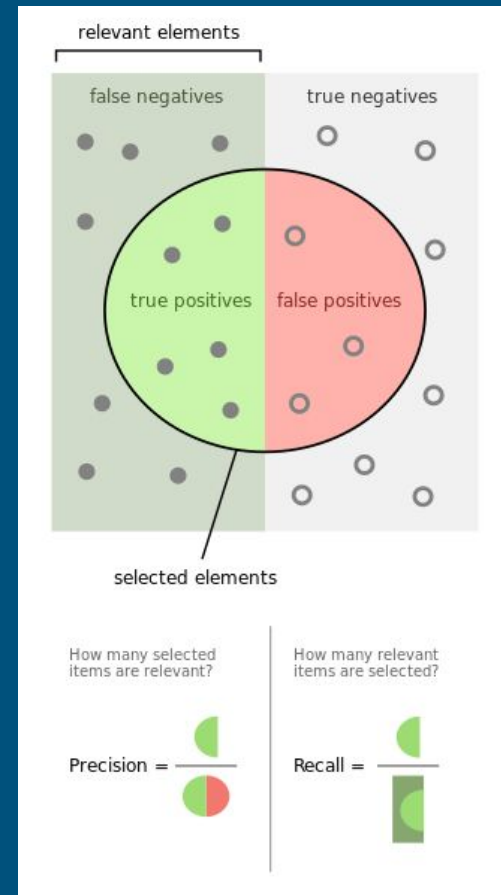
# Métricas

## Precision

Nos dice qué porcentaje de las instancias predichas son relevantes, es decir, pertenecen a la clase.

## Recall

Nos dice qué porcentaje de las instancias relevantes fueron clasificadas en la clase.



# Linear Kernel - $C = 0.25$

---

	Precision	Recall	F1-Measure	Instancias
Ninguno	0.68	0.76	0.72	4288
Positivo	0.82	0.79	0.80	4430
Negativo	0.72	0.70	0.71	3180
Neutral	0.16	0.02	0.03	262
AVG/Total	0.73	0.74	0.73	12160

# Linear Kernel - C = 0.25

---

	Real Ninguno	Real Positivo	Real Negativo	Real Neutral
Predicho Ninguno	3265	743	736	46
Predicho Positivo	466	3482	197	84
Predicho Negativo	553	194	2235	127
Predicho Neutral	4	11	12	5



# RBF Kernel - $C = 2^{1.5}$ - $\text{Gamma} = 2^{-3.5}$

---

	Precision	Recall	F1-Measure	Instancias
Ninguno	0.69	0.76	0.72	4288
Positivo	0.83	0.79	0.81	4430
Negativo	0.71	0.72	0.71	3180
Neutral	0.37	0.03	0.05	262
AVG/Total	0.74	0.74	0.74	12160

# RBF Kernel - $C = 2^{1.5}$ - $\text{Gamma} = 2^{-3.5}$

---

	Real Ninguno	Real Positivo	Real Negativo	Real Neutral
Predicho Ninguno	3260	700	711	39
Predicho Positivo	447	3512	186	79
Predicho Negativo	579	214	2277	137
Predicho Neutral	2	4	6	7


# Simple Classifier

---

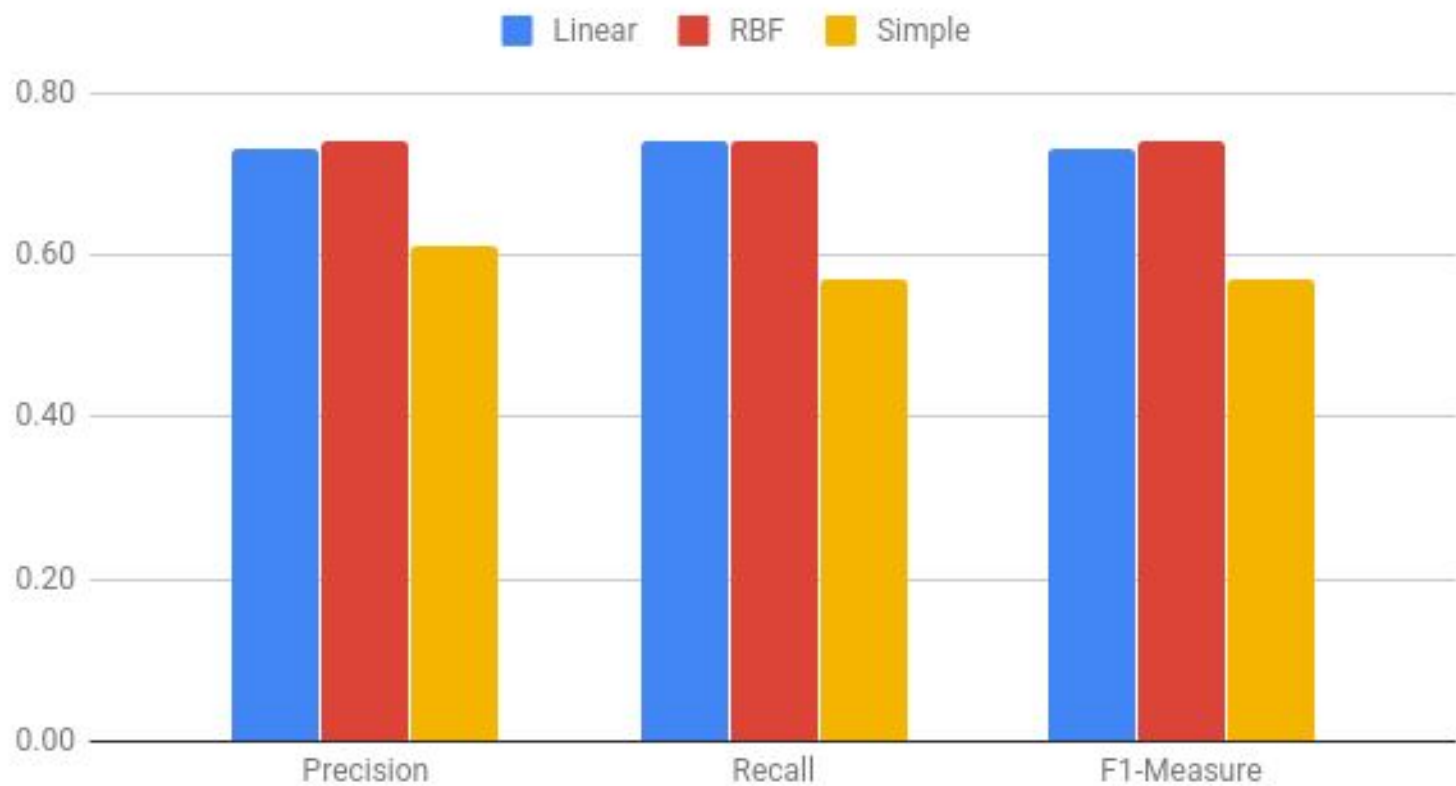
	Precision	Recall	F1-Measure	Instancias
Ninguno	0.55	0.66	0.60	21416
Positivo	0.64	0.65	0.64	22233
Negativo	0.68	0.37	0.48	15844
Neutral	0.06	0.19	0.10	1305
AVG/Total	0.61	0.57	0.57	60798

# Simple Classifier

---

	Real Ninguno	Real Positivo	Real Negativo	Real Neutral
Predicho Ninguno 	14161	5981	5199	241
Predicho Positivo	4609	14473	3078	525
Predicho Negativo	1802	682	5837	287
Predicho Neutral	844	1097	1730	252

## AVG/Total



# Código Fuente e Informes

---

Todo el material relativo a esta tesis se encuentra en GitHub:

- <https://github.com/nietol/tesis>

En el repositorio, archivo README.md, se describe el proceso de setup del ambiente para ejecutar los proyectos.

Fin

