

II Jornadas Internacionales de Estadística Aplicada 5 y 6 de diciembre de 2019

¿Qué pasó en las PASO? Descifrando el sentimiento político con Machine Learning

Autores: Angeli, Francesco*; Benites, Franco*; Tabuenca, Juan Manuel*; Taljuk, Lain**

Institución: *Facultad de Ciencias Económicas, Universidad Nacional de Tucumán

**Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán

Datos de contacto: +543815053940 , 1franco.benites@gmail.com

RESUMEN

A la luz de las cada vez más grandes imprecisiones que presentan las tradicionales encuestas a la hora de predecir elecciones, este trabajo se propone explicar los resultados de las PASO del 11/08/2019 e intentar dar una predicción de los resultados de las elecciones presidenciales del 27/10/2019, utilizando la técnica de sentiment analysis en tweets.

El análisis de sentimientos es una tarea de clasificación masiva de datos de manera automática, que busca extraer información subjetiva para determinar la polaridad de un documento o la actitud de un usuario respecto a algún tema. El análisis de posts en la web de microblogging Twitter, resulta de especial interés debido a la naturaleza subjetiva de los mensajes, la limitación de caracteres y su fácil acceso, aspectos que facilitan su análisis y procesamiento.

En este trabajo se aplicaron técnicas de aprendizaje automático (Machine Learning) en combinación con NLP (Natural Language Processing) a un conjunto de tweets recolectados previamente a las fechas de las elecciones del 11/08/2019 y del 27/10/2019 , con el fin de categorizarlos de modo automático bajo las etiquetas positivo, negativo o ambiguo y, de este modo, intentar brindar una respuesta a la inquietud de si los mensajes intercambiados en esta plataforma online reflejan verosíblemente sentimientos acerca de la política manifestados de manera offline.

Quedaría planteada la interrogante en vistas a futuras investigaciones, si un análisis como el de este trabajo, representa una alternativa costo-eficiente y más precisa a las encuestadoras tradicionales

Palabras Clave: Machine Learning; pronósticos; Análisis de Sentimiento; clasificación lineal; aprendizaje supervisado.

INTRODUCCIÓN

En la actualidad el poder de las redes sociales para predecir resultados de campañas electorales sigue siendo muy discutido y no se lo considera un método válido para la predicción de elecciones, por lo cual se sigue optando por invertir millones en las encuestadoras tradicionales, ignorando el poder que ofrecen el sentiment analysis y el machine learning para extraer valor tangible y directo de cualquier escrito. Por ej. en solo segundos se podría conocer la opinión miles de consumidores sobre un determinado producto, detectar comentarios tóxicos en periódicos online con millones de lectores, clasificar textos muy extensos según su contenido, etc.

Un caso muy conocido fue el que llevó a cabo la administración de Obama, utilizando el análisis de sentimiento para sondear la opinión pública sobre sus políticas y mensajes de campaña antes de las elecciones presidenciales del 2012 .

El gran desafío de este trabajo fue adaptarse a las herramientas existentes para procesamiento de texto, ya que la mayoría de los Frameworks para análisis de texto en Python (lenguaje de programación que se utilizó en todo el proyecto) vienen preparados para análisis de textos en inglés, con modelos adaptados para determinar el sentimiento o polaridad de un nuevo texto, lo cual en español no ocurre, ya que las herramientas existentes sólo permiten un procesamiento básico de texto, por lo que un análisis complejo de la semántica textual dependerá de las habilidades del programador y de su capacidad para adaptar estas herramientas al problema dado.

Otra barrera fue la inexistencia de un corpus de política argentina (ver subsección 2.A.3) adaptado para nuestro lenguaje (Español-Argentina), por lo que fue necesario hallar las guías apropiadas para armar un Corpus lingüístico propio, compuesto con más de 3000 mensajes clasificados manualmente por los 4 integrantes de este trabajo (más adelante se explicará en detalle su grado de importancia).

No tenemos connotación de otros trabajos de este estilo realizados hasta el día de la fecha (enfocados a política argentina), aunque no se descarta la posibilidad de que existan investigaciones de carácter privado que no estén disponibles para el público general.

Por este motivo resultan interesantes los enfoques que se llevaron a cabo en este trabajo y los resultados finales obtenidos.

METODOLOGÍA

- Sección A: Obtención de los datos
- Sección B: Preprocesamiento de los datos (limpieza del texto).
- Sección C: Machine learning pipeline
- Sección D: Extracción de Insights

2.A-Obtención de datos

2.A.1 Scraping de tweets

Como ya se mencionó anteriormente, los datos fueron extraídos de Twitter. Para esta tarea se utilizó el framework GetOldTweets3 por un motivo importante: la API oficial de Twitter para extraer tweets de los usuarios tiene una restricción de tiempo que no permite obtener tweets con más de una semana de antigüedad desde la fecha actual (hay que tener en cuenta que esta investigación se comenzó en septiembre, y las PASO 2019 se llevaron a cabo el 11 de agosto).

GetOldTweets3 permite buscar tweets más antiguos y profundos, sin limitarnos a solo una semana de búsquedas.

Se utilizaron como filtro de búsqueda seis palabras claves que representen a cada uno de los diferentes partidos (también se incluyeron algunos hashtags que fueron trending topic en esa semana previa). Por ejemplo, para el partido **“Juntos por el Cambio”** los tweets se filtraron con las siguientes palabras:

- macri
- pichetto
- #JUNTOSPORELCAMBIO
- @MAURICIOMACRI
- @MIGUELPICHETTO
- @JUNTOSCAMBIOAR

Este procedimiento se repitió para los 6 partidos considerados con mayor captación de votos (*Frente de Todos, Juntos por el Cambio, Consenso Federal, Frente Despertar, Fit Unidad, Frente NOS*).

Además de los filtros por palabras clave, se impuso un límite a la cantidad total de tweets scrapeados por cada palabra, ya que no se contaba con buenos recursos de hardware para poder captar la totalidad de tweets deseados.

Otro filtro importante a destacar es que cada tweet obtenido en el scraping sea catalogado como “popular” en la red social, con el fin de buscar representatividad en la muestra y optimizar los recursos disponibles. Por ejemplo los tweets con cero retweets o cero favoritos no fueron tomados en cuenta.

El dataset final quedó construido aproximadamente con 109 mil tweets del día previo a las PASO 2019, ya que según el enfoque dado en [1] se obtienen mejores resultados solo con información del día previo a las elecciones. Estos tweets quedaron repartidos de manera no equitativa entre los 6 candidatos (hay que tener en cuenta que algunos candidatos son menos populares que otros, por lo que la cantidad de tweets scrapeados para ellos es menor).

En la subsección 2.A.2 se muestra la cantidad de tweets finales por cada partido, ya que primero se deben eliminar las menciones conjuntas (esto también se explica en la subsección 2.A.2).

El dataset final es un archivo de excel que se compone de 6 sheets, una por cada partido.

2.A.2 Menciones conjuntas

Para que los resultados presenten el menor sesgo posible, fue necesario eliminar lo que se conoce como “menciones conjuntas”, es decir, la mención de un candidato dentro de un dataset ajeno a su partido.

Por ej. mencionar de forma negativa al candidato Lavagna dentro del dataset de “Juntos por el Cambio” provocaría que se sume un voto negativo al candidato Mauricio Macri, lo cual no sería correcto (una mejora a futuro será que el algoritmo sea capaz de determinar hacia qué partido está dirigido el tweet).

Para eliminar menciones conjuntas se escribió un algoritmo que busque todas las posibles formas en las que se pueda nombrar a alguno de los candidatos y/o partidos dentro de un dataset que no le corresponda.

Luego de llevar a cabo esto, la cantidad final de tweets por cada partido fue:

Frente de Todos : 17338 tweets

Juntos por el Cambio: 29736 tweets

Consenso Federal : 1016 tweets

Frente Despertar : 5174 tweets

Fit Unidad : 2254 tweets

Frente NOS: 1032 tweets

2.A.3 Corpus de entrenamiento

En la mayoría de los casos, los datos que se utilizan para entrenar un algoritmo de machine learning pueden significar el éxito o el fracaso del modelo final (y del proyecto en sí).

Como el objetivo de este trabajo consiste clasificar de manera automática los tweets scrapeados según su polaridad (positiva, negativa o ambigua), es necesario contar con un corpus lingüístico adaptado a política argentina para poder entrenar el modelo de machine learning.

En resumidas palabras, un corpus de entrenamiento es un conjunto de textos relativamente grande, creado independientemente de sus posibles formas o usos, con el objetivo de que sean utilizados para hacer análisis estadísticos y contrastar hipótesis sobre el área que estudian.

Ante la necesidad de adaptar el algoritmo a las expresiones y modismos que utilizan los argentinos al momento de hablar sobre política, se decidió armar un corpus propio de entrenamiento y no hacer uso de corpus lingüísticos internacionales, como los que provee la sociedad española de Natural Language Processing (corpus TASS) .

Para armar este corpus político se clasificó manualmente cerca de 3 mil tweets del total de tweets scrapeados, con las leyendas positivo, negativo y ambiguo, siempre buscando que las 3 clases se encuentren balanceadas (mil tweets clasificados por cada sentimiento). Este corpus de entrenamiento permitió que la precisión final del algoritmo aumente considerablemente (pasando del 56% al 94%), como era de esperarse, ya que es un corpus mucho más adaptado al problema planteado.

2.B-Preprocesamiento de los datos

2.B.1 Reducción del contenido del tweet

Otro paso fundamental para este proyecto consiste en la limpieza de los datos. El objetivo de la limpieza de datos es reducir el contenido de cada tweet a su mínima expresión, eliminando todo lo que no contribuya a su polaridad (ya sea positiva, negativa o ambigua). Teniendo en cuenta lo anterior, en cada tweet el algoritmo de preprocesamiento aplicó las siguientes acciones:

- Convirtió todo el texto del tweet a minúscula.
- Eliminó caracteres no alfabéticos.
- Eliminó todo tipo de URL's.
- Eliminó nombres de usuario, emojis, etc.
- En cada tweet, el algoritmo detecta todo tipo de insulto o palabra despectiva y lo transforma a la etiqueta "toxicword", con el objetivo de unificar en una sola palabra muchísimas expresiones de carácter despectivo, sin importar si este insulto está escrito en singular o plural, con caracteres repetidos, o dirigido hacia un hombre o una mujer.

- Detectó la mención de algún candidato ya sea por nombre, apellido o apodo y lo transformó a la etiqueta "somecandidate", ya que el nombre de un candidato no interviene en el sentimiento general de un tweet (aunque sería útil para proyectos más avanzados que impliquen reconocer hacia quien está dirigido el mensaje).
- Eliminó "stopwords" (palabras que se repiten con mucha frecuencia pero que no aportan demasiado valor sintáctico, como por ej: de, por, con,)
- Unificó todo tipo de risas :
'ajajajajajaj' ----> 'jaja', 'jojojo'----> jaja
- Llevó a cabo un spell-checking, es decir, corrigió las palabras mal escritas debido a modismos usados en redes sociales.
Ej: 'xq'--> 'porque', 'q' ---> 'que', etc.
- Eliminó frases que hagan referencia a cosas dichas por otras personas.
Por ej: El candidato dijo que "tendremos un futuro mejor" ---> 'El candidato dijo que'

Entonces, con este preprocesamiento, se sabe lo que se necesita mantener en cada tweet y lo que se necesita sacar. Este preprocesamiento se aplica a los conjuntos de entrenamiento y prueba.

A continuación un ejemplo de un tweet luego de ser procesado (fue elegido al azar, no existe intención de mostrar ningún tipo de inclinación política):

Tweet: *Lavagna YO te #apoyo en estas elecciones!!!!. Todos los hipócritas que no te voten pueden ver el siguiente enlace www.link.com*

→ *somecandidate apoyo elecciones toxicword no voten pueden ver enlace.*

El algoritmo también es capaz de llevar a cabo un método conocido como "stemming", un proceso por el cual se transforma cada palabra en su raíz.

Por ej. las palabras encantado, encantada o encantados comparten la misma raíz y se consideran la misma palabra tras el stemming. En este trabajo se desactivó la acción de stemming del algoritmo, ya que resultó ser contraproducente para la precisión final del modelo de machine learning.

2.C Machine Learning pipeline

2.C.1 Vector de características

Para poder analizar los tweets con un modelo de ML, es necesario extraer y estructurar la información contenida en el texto de forma numérica. Existen muchas maneras de realizar esto, pero en este trabajo se utilizó el tipo de vectorización TF*IDF, que sirve para analizar la importancia que tienen ciertas palabras en comparación con todas las disponibles en el documento.

TF determina la frecuencia relativa de una palabra específica en un documento. Este valor se compara con la frecuencia relativa de todos los demás términos de un texto (el logaritmo evita que un aumento sustancial del uso de una palabra específica no afecte al valor final del cálculo).

$$TF(i) = \frac{\log_2(Freq(i,j)+1)}{\log_2(L)}$$

IDF es la frecuencia inversa de un texto. Es muy importante ya que incluye en el cálculo la frecuencia de el texto de términos específicos: compara el número de todas la palabras disponibles con el número de textos que contienen el término. Entonces lo que hace IDF es determinar la relevancia de un texto con respecto a una palabra específica.

$$IDF_t = \log \left(1 + \frac{N_D}{f_t} \right)$$

2.C.2 Selección del algoritmo de ML

Como el objetivo de este problema es clasificar los tweets bajo una etiqueta previamente conocida (positivo, negativo o ambiguo), se decidió utilizar un modelo de aprendizaje supervisado.

El aprendizaje supervisado se refiere al subconjunto de ML donde se generan modelos para predecir el resultado de salida en base a ejemplos históricos de esa variable de salida. En este caso, el modelo de ML se entrenará con el CORPUS preprocesado, y utilizará un 80% de el para entrenarse y, un 20% para testear sus predicciones.

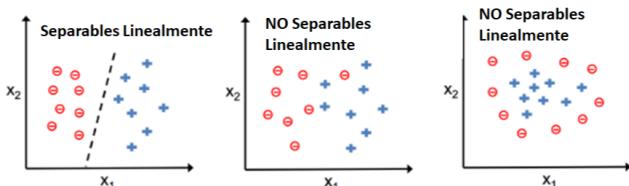
La tarea de clasificación es una subcategoría del aprendizaje supervisado en la que el objetivo es predecir las etiquetas de clase categóricas de las nuevas instancias, basadas en observaciones pasadas. Existen dos tipos de clasificaciones: la Binaria y la Multi-clase (una observación puede ser asignada a una de múltiples categorías, en este caso, un tweet puede ser positivo, negativo o ambiguo).

Es importante indicar que no todos los modelos de clasificación son útiles para separar adecuadamente las diferentes clases de un conjunto de datos.

Algunos algoritmos no convergerá al aprender los pesos del modelo si las clases no pueden separarse por una frontera de decisión lineal. Por ejemplo algunos casos típicos:

Luego de ser evaluados distintos algoritmos, (que cumplan con los requisitos antes explicados) entre ellos: Multinomial Naïve Bayes, OneVsRest Random Forest y OneVsRest Linear SVC , se

eligió el último mencionado.



Las métricas típicas que se utilizan en problemas multiclase son las mismas que se utilizan en el caso de la clasificación binaria, con la diferencia que se calculan para cada clase al procesarla como un problema de clasificación binaria luego

de agrupar todas las otras clases como pertenecientes a la segunda clase.

En la tabla.1 adjuntan los valores de las métricas de evaluación usadas, en donde se puede observar una precisión del algoritmo de clasificación por arriba del 89% para cada clase.

	PRECISION	RECALL	F1-SCORE
POSITIVOS	0,894	0,733	0,806
NEGATIVOS	0,957	0,976	0,967
AMBIGUOS	0,949	0,941	0,945

Tabla.1: Resultados de métricas de evaluación

DESARROLLO

2.D Extracción de insights

2.D.1 Resultados reales vs Sentiment Analysis

La idea principal en este punto es determinar si realmente existe algún tipo de relación entre el comportamiento político de las personas en redes sociales y su comportamiento político en la vida real.

Para esto, es necesario encontrar un modo de relacionar el porcentaje de tweets clasificados como positivos, negativos o ambiguos por el modelo de ML, con el resultado real que ocurrió en las elecciones provisionales en Argentina, las PASO 2019.

Siguiendo los estudios expuestos por [1] y [2], se pueden hallar dos métodos interesantes de predicción:

- 1) El primer método de predicción consiste solamente en contar el número de tweets que mencionan a cada candidato. Según estos estudios, la proporción de tweets que mencionan a cada candidato debería reflejar de cerca la participación real de los votos en las elecciones.
- 2) El segundo método de predicción aprovecha la naturaleza bipartidista de un proceso electoral (como es el caso de Argentina), y propone un método conocido como "vote-share", donde se considera que la mayor parte de los votos de la elección estarán dirigidos solamente hacia los dos candidatos más populares:

$$vote_{share}(c1) = \frac{P(c_1) + N(c_2)}{P(c_1) + P(c_2) + N(c_1) + N(c_2)}$$

$$vote_{share}(c2) = \frac{P(c_2) + N(c_1)}{P(c_1) + P(c_2) + N(c_1) + N(c_2)}$$

Algunas cuestiones sobre las fórmulas anteriores:

- Se etiquetan los 2 candidatos más populares como C1 y C2 respectivamente.
- $vote_share(C1)$ es la fórmula que computa los votos para el candidato C1.
- $vote_share(C2)$ es la fórmula que computa los votos para el candidato C2.
- $Pos(C1)$ y $Neg(C1)$ son, respectivamente, el número de tweets positivos y negativos que mencionan al candidato C1.
- $Pos(C2)$ y $Neg(C2)$ son, respectivamente, el número de tweets positivos y negativos que mencionan al candidato C2.

Particularmente para este trabajo, el segundo método entregó mejores resultados, ya que refleja mejor el clima electoral argentino.

Hay que tener en cuenta que comúnmente en una carrera electoral participan más de dos candidatos, por lo que es necesario normalizar los resultados de las PASO 2019 para que puedan alcanzar el 100%. Finalmente se compara este resultado normalizado con aquél del vote-share para sacar conclusiones.

En la tabla.2 se muestra el porcentaje real de votos de las PASO 2019 hacia cada candidato:

Candidato	%total de votos
Macri	31.8%
Fernandez	47.79%
Espert	2.16%
Del caño	2.83%
Lavagna	8.15%
Centurión	2.62%
TOTAL	95.35%

Tabla.2.Resultados reales de las PASO 2019

Claramente se observa que Mauricio Macri y Alberto Fernández fueron los dos candidatos más populares de las elecciones primarias. Aunque Roberto Lavagna también obtuvo una buena cantidad de votos, para este trabajo se consideró despreciable su porcentaje con respecto a los dos primeros candidatos, ya que la idea es utilizar el método del vote-share, el cual resulta válido para elecciones bi-partidistas o muy polarizadas.

Normalizando los votos que obtuvieron los dos candidatos más populares se obtiene:

$$\%normalizado_{Alberto.F} = 60.05\%$$

$$\%normalizado_{Mauricio.M} = 39,95\%$$

Ahora es necesario estudiar los resultados que arrojó el algoritmo de ML con la técnica de sentiment analysis. Los mismos se adjuntan en la tabla.3, indicando la cantidad de votos positivos y negativos hacia cada candidato (los tweets ambiguos fueron descartados ya que no representan una intención de voto definida):

Candidato	Tweets Positivos	Tweets Negativos	Tweets Totales
Macri	3.406,00	22.245,00	29.736,00
Fernandez	2.511,00	11.436,00	17.338,00
Espert	717,00	3.275,00	5.174,00
Del Caño	318,00	1.432,00	2.254,00
Lavagna	193,00	555,00	1.016,00
Centurion	117,00	717,00	1.032,00
TOTAL	7.262,00	39.660,00	56.550,00

Tabla.3: Resultados obtenidos aplicando sentiment analysis

Ahora se aplica el vote share para ambos candidatos (la sigla AF representa al candidato Alberto Fernández y la sigla MM al candidato Mauricio Macri):

$$vot_{share}(AF) = \left(\frac{2511 + 22245}{3406 + 22245 + 2511 + 11436} \right)$$

$$\Rightarrow vot_{share}(AF) = 62,52\%$$

$$vot_{share}(MM) = \left(\frac{3406 + 11436}{3406 + 22245 + 2511 + 11436} \right)$$

$$\Rightarrow vot_{share}(MM) = 37,48\%$$

En resumen se tiene:

Resultado normalizado de las PASO 2019	Resultado con sentiment analysis y vote-share
$\%norm_{Alberto.F} = 60.05\%$	$vot_{share}(AF) = 62,52\%$
$\%norm_{Macri} = 39,95\%$	$vot_{share}(MM) = 37,48\%$

Es posible observar una gran similitud entre el porcentaje de votos normalizados de las PASO

y el porcentaje de votos calculados por el vote-share.

Por lo tanto, el análisis de sentimiento logró explicar con un margen de error absoluto de aproximadamente % 2,5 el resultado normalizado de las PASO.

Si bien el candidato Mauricio Macri obtuvo mayor cantidad de tweets positivos que Alberto Fernández, la cantidad de tweets negativos que recibió fue muchísimo mayor en comparación con la del candidato del Frente de Todos.

Resulta interesante como una red social como Twitter pudo reflejar tan bien los resultados de las PASO 2019, por lo que en la siguiente sección se aplicará el mismo método con el objetivo de dar una predicción para las elecciones del 27 de octubre.

2.D.1 Predicción de las elecciones 27/10/2019

Para esta última sección del trabajo se aplicó todo lo visto anteriormente, con el objetivo de dar una predicción de las elecciones del 27/10/2019 haciendo uso solamente de Twitter en combinación con sentiment analysis y machine learning.

Primero se hizo scraping de Tweets con 48 hs previas a las elecciones (con el fin de intentar captar más tweets), siguiendo la misma metodología de la subsección 2.A.1. Luego se eliminaron menciones conjuntas, se aplicó el preprocesamiento a todos los tweets y se utilizó el modelo de ML para etiquetar cada tweet según su polaridad, obteniendo los resultados de la tabla.3 (sólo se muestran los valores para los dos candidatos con mayor captación de votos) :

Candidato	Tweets positivos	Tweets negativos	Tweets Totales
Macri	8253	26366	34619
Fernandez	3684	10537	14221
TOTAL	8678	35465	48840

Tabla.4: Resultados obtenidos aplicando sentiment analysis

Aplicando la fórmula del vote share se obtiene:

$$vot_{share}(AF) = \left(\frac{3684 + 26366}{3684 + 10537 + 8253 + 26366} \right)$$

$$\Rightarrow vot_{share}(AF) = 61,53\% \quad vot_{share}(MM) = \left(\frac{8253 + 10537}{3684 + 10537 + 8253 + 26366} \right)$$

$$\Rightarrow vot_{share}(MM) = 38,47\%$$

Con estos resultados se observa que, en comparación con las elecciones primarias del 11 de agosto, el candidato Mauricio Macri reduce la diferencia de votos con respecto a Alberto Fernández, pero no cuenta con la cantidad de votos necesaria para llegar a un ballottage.

Este resultado fue calculado el domingo 27/10/2019 a horas 19:23, antes de conocerse los resultados oficiales de las elecciones.

Luego de publicarse los escrutinios finales, se pudo llevar a cabo la comparación. En la tabla.5 se adjuntan los resultados reales ocurridos en las elecciones del 27/10/2019:

Candidato	%total de votos
Macri	40.38%
Fernandez	48.10%
Espert	1.48%
Del caño	2.16%
Lavagna	6.01%
Centurión	1.71%
TOTAL	100%

Tabla.5: Resultados reales de las elecciones del 27/10/2019

Normalizando los votos de los dos candidatos más votados se obtiene:

$$\%normalizado_{Alberto.F} = 54,37\%$$

$$\%normalizado_{Mauricio.M} = 45,63\%$$

Esto significa que con la técnica de sentiment analysis se pudo dar una predicción de los resultados finales con un error absoluto de aproximadamente 7%, al mismo tiempo que fue posible predecir que el candidato Mauricio Macri reduciría la diferencia de votos con respecto al candidato del Frente de Todos, Alberto Fernández.

CONCLUSIONES

Con este trabajo queda demostrado el potencial que pueden tener las redes sociales para realizar inferencias sobre elecciones presidenciales, pero todavía falta mucho para mejorar como para considerarlo un método válido de predicción de elecciones.

Sería interesante llevar a cabo este mismo estudio con otras redes sociales como facebook, Instagram o diarios Online y compararlo con los resultados que arrojó la plataforma de Twitter.

Por otro lado, si se cuenta con las herramientas correctas de Hardware, para elecciones futuras podrían tomarse datos en streaming con la API de twitter durante varios meses previos a las elecciones, lo cual permitiría tener una muestra mucho mayor de datos y realizar un estudio más preciso.

Otro punto destacable de este trabajo fue el éxito obtenido en la aplicación de técnicas de NLP y Sentiment analysis gracias a la creación de un corpus político adaptado para nuestro país y el algoritmo de procesamiento de texto diseñado.

Incluso la metodología de clasificación mediante NLP y procesamiento de texto, es aplicable a muchas otras ramas fuera del ámbito político como ser: polaridad de reviews (actitud de la gente con respecto a un producto o marca), categorización de problemas en una empresa a través de comentarios que dejan los clientes, motores de recomendación de productos, chatbots, etc.

Algunas mejoras futuro para este proyecto podría ser:

- La inclusión de politólogos, sociólogos y personas estudiosas de la lengua española para mejorar el corpus de entrenamiento de modo que resulte útil para estudios futuros, como así también buscar alternativas al método del vote-share, de modo que puedan ser incluidos todos los candidatos.

- Mejorar el algoritmo de modo que sea capaz de detectar sarcasmo, cambios en el lenguaje, hacia qué candidato o partido está dirigido un mensaje.

- Hallar algún tipo significado político a los mensajes etiquetados como negativos para candidatos considerados con menor captación de voto, como así también los tweets etiquetados como ambiguos (¿hacia quien podrían ir dirigidos esos votos? ¿Por qué?).

No queda duda que cada vez son más los usuarios que interactúan en redes sociales, por lo que en el mediano plazo el Big Data podría competir de cerca con las encuestadoras tradicionales.

BIBLIOGRAFÍA

[1] Metaxas et al. (2011), *How (not) to predict elections. IEE International Conference on Social Computing.*

[2] Tumasjan et al. (2010), *Election Forecasts with Twitter: How 140 characters reflect the political landscape. Social Science Computer Review.*

[3] Blank (2016), *The Digital divide among Twitter users and its implications for social research. Social Science Computer Review.*

[4] Huberty, M. (2015), *Can we vote with our tweet? On the perennial diculty of election forecasting with social media. International Journal of Forecasting.*

[5] Choy, Murphy & Cheong, Michelle & Laik, Ma & Shung, Koo. (2011). *A sentiment analysis of Singapore Presidential Election 2011 using Twitter data with census correction.*

[6] Patodkar et al. (2016), *Twitter as a Corpus for Sentiment Analysis and Opinion Mining. International Journal of Advanced Research in Computer and Communication Engineering.*

[7] Sobrino Sande (2018), *Análisis de Sentimientos en Twitter. Tesis Maestría en Ingeniería Informática Universitat Oberta de Catalunya.*