



# **Minería de textos y análisis de sentimientos de los tweets de el proceso de impeachment en el Brasil.**

## Proyecto final

Autor:

Caio Fernandes Moreno / Caio Moreno de Souza

Tutor: César de Pablo

Fecha: 6/02/2016

---

## Contenido

1.	Resumen.....	3
2.	Introducción.....	3
3.	Objetivos .....	4
4.	Justificación.....	4
5.	Metodología.....	4
6.	Material y Métodos .....	6
7.	Cronograma.....	24
8.	Periodo y Lugar.....	25
9.	Bibliografía.....	25
10.	Anexo .....	25

## 1. Resumen

El objetivo de este trabajo ha sido estudiar el sentimiento de los mensajes de los *tweets* (positivos y negativos) en tiempo real. También se ha estudiado una muestra recogida entre los meses de diciembre de 2015 y enero de 2016. Se trata de un total de aproximadamente 900 mil *twittes*.

Los mensajes recogidos contenían la palabra "Dilma" en el mensaje. Todos los *twittes* fueron recogidos con la *API de Streaming del Twitter*.

En 03 de diciembre de 2015 se aceptó el pedido de apertura del proceso del *impeachment* de el presidente de Brasil, Dilma Rousseff. El pedido ha sido aceptado por el presidente de la Cámara de los Diputados, el diputado Sr. Eduardo Cunha (PMDB-RJ), y de este modo se creó una expectativa sobre el sentimiento de la población y el futuro de Brasil.

Es muy importante entender que lo que se publica en la red social Twitter, no se puede manipular y representa la opinión de la persona que publica lo mensaje. Por esto se puede decir que hacer el proceso de minería de datos con los datos del Twitter puede ser muy eficiente y verdadero.

La idea es conseguir, a través de la muestra recogida y utilizando técnicas de minería de textos, identificar informaciones como: persona más influyente en el contexto, mensaje que ha sido más compartida en la red social, nube de palabras asociadas a la palabra Dilma, y de esta forma entender mejor el sentimiento o la opinión de la muestra recogida en aquel momento.

## 2. Introducción

En una sociedad democrática el sentimiento de la población es extremadamente importante y refleja la continuación o no de un gobierno o forma de gobierno. Para ayudar a asegurar su éxito un gobierno necesita estar en constante vigilancia del sentimiento de la población para identificar los elementos en que se está cumpliendo con las personas y qué cuestiones deben ser mejoradas.

Los sitios web de microblogging se han desarrollado hasta el punto de convertirse en una valiosa fuente de variada clase de información, debido a su capacidad de recoger lo que la gente publica en los mensajes en tiempo real acerca de sus opiniones sobre una variedad de temas, cuestiones debatidas en la actualidad, se queja, y expresa el sentimiento positivo, negativo o neutral para los productos que se utilizan en la vida diaria.

Un gran desafío es la construcción de la tecnología para detectar y resumir un sentimiento general. Utilizando el microblog popular llamado Twitter, que es la más importante plataforma existente actualmente en condiciones de recoger los sentimientos de la gente, sus opiniones y proporcionarlas de forma **gratuita y abierta**, será desarrollada una herramienta sobre la base de sus *tweets* donde se construirán

modelos de clasificación de sentimiento en positivo o negativo.

Esta herramienta puede ser utilizada por el Gobierno, o por los ciudadanos, para entender mejor la población, a entender lo que la gente de la muestra colectada sea en tiempo real o no, lo que se piensa y también sus líderes en las redes sociales, identificar las personas más activas en el uso de herramientas sociales para expresar su opinión, y también identificar a los más influyentes.

### 3. Objetivos

Es posible detallar el objetivo de este trabajo respondiendo la pregunta ¿que?

- ¿Que?

Crear una herramienta para monitorear los comentarios en Twitter y predecir el sentimiento (positivo o negativo) de tweets.

### 4. Justificación

- ¿Por qué?

Debido a que los datos extraídos del Twitter son información pública, porque la persona que lo publicó expresó su opinión de su propia y legítima voluntad en una red social, esta información es muy valiosa y puede ser extraída y utilizada para fines positivos. Esta información no debe utilizarse para fines malvados, de ninguna manera este control puede ser un instrumento de coacción o intimidación de la población.

Así, uno de los usos positivos, propuesto en este trabajo, es el empleo de la herramienta en cuestión con el propósito de conseguir a través de la muestra recogida entender el sentimiento o opinión de la muestra recogida en el momento (que fue un momento histórico del año 2015 e 2016, el proceso de la solicitud de *impeachment* del Presidente de Brasil).

- ¿Para qué?

El objetivo de la investigación es aprender y aplicar las técnicas de minería de textos y procesamiento de lenguaje natural con la finalidad de entender el sentimiento de las personas en relación a la actual presidente de Brasil.

### 5. Metodología

- ¿Cómo?

Mediante la realización de cuatro acciones principales: extracción y filtro de los datos colectados, preparación de los datos, predicción del sentimiento utilizando un algoritmo de clasificación o la técnica de contar palabras (frecuencia) que sean

---

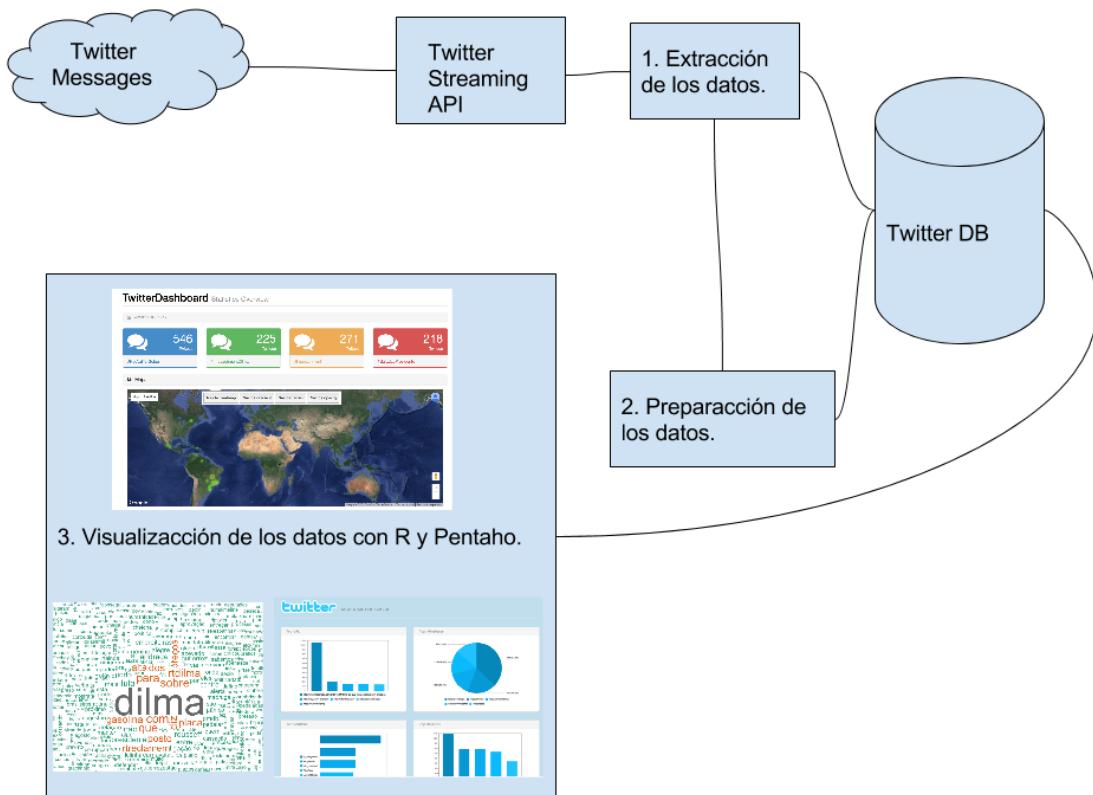
negativas y positivas y entonces categorización los datos en positivo y negativo, y la visualización de las siguientes informaciones:

- ¿Quiénes son las personas con más seguidores con la mayor cantidad de mensajes negativos?
- ¿Quiénes son las personas con más seguidores con la mayor cantidad de mensajes positivos?
- ¿Quiénes son las 10 personas con el mayor número de seguidores con una mayor cantidad de mensajes negativos?
- ¿Quiénes son las 10 personas con el mayor número de seguidores con una mayor cantidad de mensajes positivos?
- ¿Cuáles son los mensajes más relevantes? (La medida de relevancia se hará a través de la cantidad de *retweets* hechos. Es decir, cuanto más menciones, o *retweets*, más relevante el mensaje).

## 6. Material y Métodos

El trabajo se consiste en crear una herramienta capaz de analizar los mensajes de twitter.

Como se puede ver en la imagen abajo se ha construido un sistema compuesto por 3 fases: extracción, preparación y visualización de los datos.



Todas las informaciones han sido extraídas del twitter a través de un programa desarrollado en Python, R y Java.

La plataforma de twitter permite realizar consultas a sus datos como se puede ver en la figura abajo.

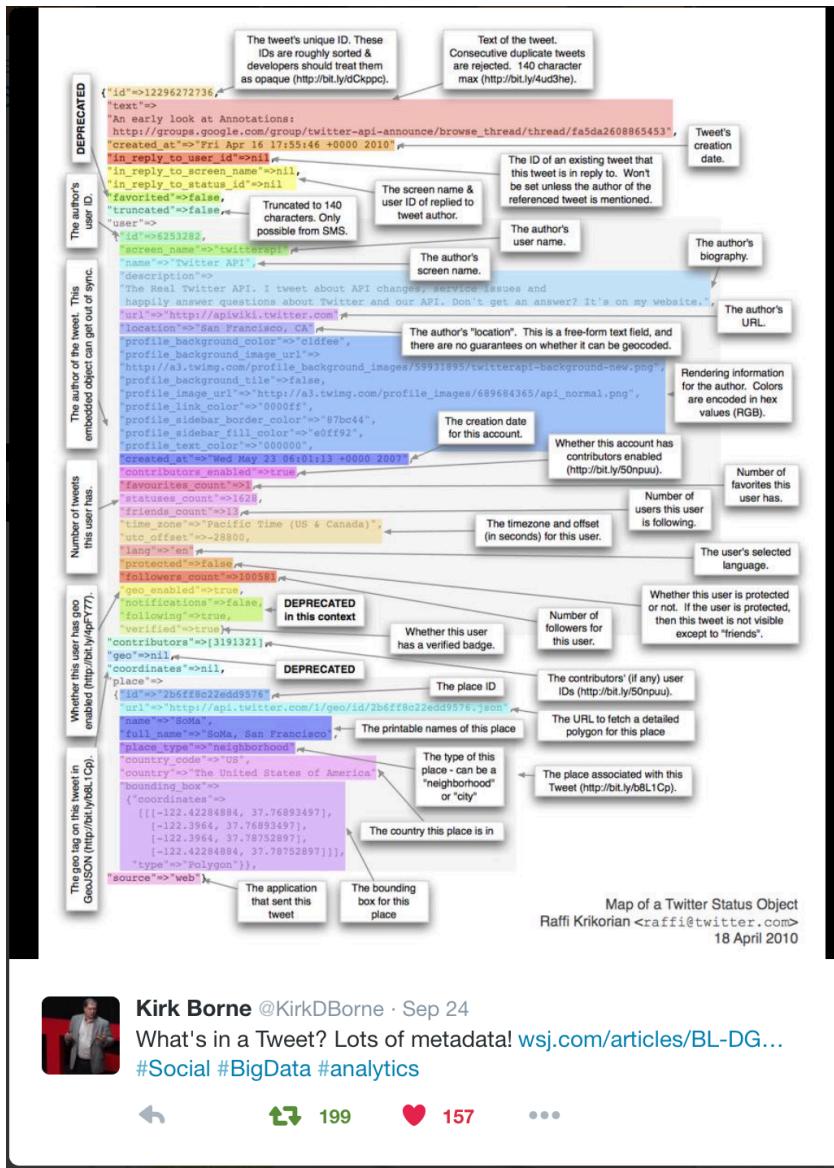
Criterio de búsqueda: Tweets – From Everyone – Near you.

Con el enlace abajo se puede hacer una búsqueda en Twitter.  
<https://twitter.com/search?f=tweets&vertical=news&q=%23bigdata&near=me&src=typd>

Un mensaje de twitter contiene 140 caracteres, pero para cada mensaje de twitter generada se almacena muchos metadatos que puede ser utilizados para minería de datos.

Abajo la imagen llamada “Map of a Twitter Status Object” creada por Raffi Krikorian en 18 de Abril de 2010 donde se puede ver la cantidad de información rica para minería de datos.

Como se puede mirar, no solo el texto es generado pero informaciones como: identificador único del usuario, sus informaciones de perfil (nombre, nombre de tela, descripción, localización, página web), su país, muchas veces su localización exacta de donde ha publicado esto solo cuando la persona permite enviar su latitud y longitud de forma consciente, cantidad de seguidores, idioma de lo mensaje, hora de su publicación, etc).



Kirk Borne @KirkDBorne · Sep 24

What's in a Tweet? Lots of metadata! [wsj.com/articles/BL-DG...](http://wsj.com/articles/BL-DG...)  
#Social #BigData #analytics



199

157

...

Con esta cantidad de información creo que es posible decidir que si es importante hacer minería de datos de twitter.

Este trabajo tiene objetivos muy modestos, pero si existen otros trabajos hechos con la base de datos de twitter donde se puede extraer muchísimas otras conclusiones de las personas basados en estos mismos datos colectados de twitter.

## 1. Extracción de los datos de Twitter.

El primero paso del trabajo ha sido crear una forma de extraer los datos de Twitter.

Todos los datos extraídos han sido publicados en <https://github.com/caiomsouza/TwitterRawData/releases/tag/DITRD-v1.0.0> y pueden ser utilizados de forma libre.

A principio esta etapa puede parecer una tarea simple, pero ha sido una tarea donde he invertido mucho tiempo y incluso no he encontrado la solución que aceptó como la ideal para mis necesidades, pero por razones de tiempo he tenido que darme como satisfecho con una de las herramientas y he avanzado en el proyecto.

Actualmente existe desarrolladas y disponibles de forma gratuita y abierta algunas APIs en *Python*, *Java*, *R* y otras lenguajes donde el primer paso es elegir cual API se desea utilizar, después aprender el funcionamiento de la API y entonces extraer los datos.

Como conozco Java, Python y R he probado distintas formas de extraer datos de Twitter con Java, Python, R, incluso utilizando Apache Spark con Java y lo que me ha resuelto en diversas herramientas de extracción de datos creadas, pero ambas haciendo lo mismo.

Las distintas APIs de Twitter suelen tener la misma capacidad de extraer datos y no suele tener muchas diferencias, entonces no quiero recomendar ninguna en especial porque no he visto muchas diferencias en las que he probado.

Me ha gustado las pruebas que he hecho con R, con Python y con Java utilizando también Apache Spark.

Algo muy importante sobre la API de Twitter es la obligatoriedad de crear una cuenta en Twitter y después un aplicativo capaz de utilizar la API de Twitter.

Con este aplicativo puedes extraer datos de Twitter pero hay un límite de mensajes que se puede extraer y todo eso se debe llevar en consideración en la hora de crear su solución de extracción de datos.

En este trabajo he utilizado la API de Streaming de datos de Twitter donde es permitido captar los mensajes que se están generando en este momento.

Se puede captar los mensajes en tiempo real o hacer una consulta en la base de datos de Twitter para una determinada palabra clave, pero esta segunda opción tiene muchas limitaciones.

Para más informaciones de cómo crear una cuenta en Twitter, una aplicación que permite utilizar la API de Twitter y para conocer mejor la API de Stream y la de consultas se recomienda entrar en [www.twitter.com](http://www.twitter.com) y leer la documentación de Twitter.

Abajo el código comentado para extraer los datos de Twitter utilizando R:

```
r_sentiment_analysis_sentilex-pt01.R  
setwd("~/git/Bitbucket/u-tad/final-project/src/r-script")  
# http://thinktostart.com/sentiment-analysis-on-twitter/
```

```
# https://jeffreybreen.wordpress.com/2011/07/04/twitter-text-mining-r-slides/
#http://www.r-bloggers.com/mining-twitter-for-consumer-attitudes-towards-hotels/

#library(devtools)
#install_github("twitteR", username="geoffjentry")
#install.packages("ROAuth")
#install.packages("RCurl")
#install.packages("bitops")
#install.packages("digest")
#install.packages("rjson")
library(twitteR)
library(plyr)
library(ROAuth)
library(bitops)
library(digest)
library(rjson)

#twitter user @caiomsouza
api_key <- " Cambiar_Para_Sus_Datos "
api_secret <- " Cambiar_Para_Sus_Datos "
access_token <- " Cambiar_Para_Sus_Datos "
access_token_secret <- "Cambiar_Para_Sus_Datos"

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
```

En esta parte se pide para escoger entre dos opciones 1 o 2. Ambas las opciones me ha funcionado sin ningún problema.

```
> setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
[1] "Using direct authentication"
Use a local file to cache OAuth access credentials between R sessions?
1: Yes
2: No
```

Selection:

En próximo paso es llamar la función **searchTwitter** con los dos parámetros: palabra clave y cantidad de mensajes a extraer.

```
tweets = searchTwitter("dilma", n=2000)
```

El código R abajo es para contar la cantidad de twitter colectado.

```
length(tweets)
```

Resultado:

```
> length(tweets)
[1] 2000
```

Se deseado se puede mirar los mensajes que se ha sido colectado y se encuentran en tweets.

Tweets

Resultado:

[[1669]]

[1] "rralves: E o que a governANTA fez? Retirou o menino de circulação para ser esquecido pela imprensa e pela justiça! Não... https://t.co/yoLMNwBymL"

[[1670]]

[1] "zaanganeles: RT @o\_antenista: Dilma em alerta: \"Ninguém sabe o que Otávio Azevedo pode falar\" https://t.co/galJu7xDyP https://t.co/qZlJSWnP4L"

[[1671]]

[1] "diegorr\_: oq a Dilma tá fazendo com o Twitter? Alguém da um jeito nessa mulher urgente!!"

[[1672]]

[1] "joalmagalhaes: Viva Juiz Moro\nDELAÇÃO DA ANDRADE GUTIERREZ – Estão em pânico Dilma, Lula, Lulinha e o PMDB do Rio https://t.co/8SPVpeAUhl vía @veja"

[[1673]]

[1] "heauxn0ire: Presidenta deve tá fritando por lá.\nDilma aproveita folga de Carnaval para pedalar em Porto Alegre https://t.co/6U3zaAvbC5"

Existe mucho material disponible en internet sobre otras formas de extraer twitters pero para mis objetivos he quedado con la opción enseñada anteriormente.

En mi caso, no voy trabajar con una gran cantidad de mensajes, la idea es de forma manual utilizando la herramienta R Studio ejecutar el código de la extracción y recoger 2000 twitters para hacer un pequeño estudio del sentimiento general de las personas que están hablando de determinada palabra clave.

He probado con otras palabras claves de mi interés personal y estas palabras no han llegado a tener ni 2000 mensajes. Para pequeñas y medianas empresas esta solución es suficiente porque ellas no llegan a tener mucha gente hablando sobre sus productos, marca, etc.

En el caso de la palabra clave “dilma” se puede encontrar muchísimos twitters que son creados a cada día.

Durante algunos días yo he recogido mas de 1 millón de mensajes con la palabra clave “dilma”, esto no lo he hecho con R y si con una herramienta hecha en Python y después con una herramienta de extraer datos hecha en Java y con Apache Spark.

En el caso de la palabra “dilma” es recomendable pensar en una infraestructura de Big Data utilizando Apache Kafka, Apache Spark y Apache Hive para la tarea de ingestión y extracción de datos y almacenaje los datos.

Para un mejor entendimiento de esta solución recomendable es necesario explicar sobre los componentes mencionados arriba.

**Apache Spark** es el motor (engine) más rápido actualmente para el procesamiento de datos en grandes volúmenes.

<http://spark.apache.org/>

**Apache Kafka** es un sistema de mensajería abierto y muy utilizado. Ha sido originalmente desarrollado por LinkedIn.

<http://kafka.apache.org/>

**Apache Hive** es una infraestructura para almacén de datos (data warehouse) construida sobre *Hadoop* para proporcionar la summarización de datos, consultas y análisis de bases de datos muy grandes en almacenamiento distribuido.

Ha sido inicialmente desarrollada por *Facebook*, pero actualmente es utilizada y desarrollada por otras empresas como *Netflix*. Amazon mantiene un fork del proyecto *Apache Hive* que ha sido incluida en su producto *Amazon Elastic MapReduce* en *Amazon Web Services*.

<https://hive.apache.org/>

<http://infolab.stanford.edu/~ragho/hive-icde2010.pdf>

## 2. Preparación de los datos extraídos de los mensajes de Twitter.

El código abajo en R es necesario para separar solo el texto de cada mensaje.

```
Tweets.text = laply(tweets,function(t)t$getText())
 Tweets.text
```

Se ha utilizado la función abajo para limpiar los datos.

```
# Función para limpiar los datos
clean.text <- function(some_txt)
{
  some_txt = gsub("&", "", some_txt)
  some_txt = gsub("(RT|via)((?:\\b\\W*@[\\w+]+)|)", "", some_txt)
  some_txt = gsub("@\\w+", "", some_txt)
  some_txt = gsub("[[:punct:]]", "", some_txt)
  some_txt = gsub("[[:digit:]]", "", some_txt)
  some_txt = gsub("http\\w+", "", some_txt)
  some_txt = gsub("[ t]{2,}", "", some_txt)
  some_txt = gsub("^\\s+|\\s+$", "", some_txt)

  # define "tolower error handling" function
  try.tolower = function(x)
  {
    y = NA
    try_error = tryCatch(tolower(x), error=function(e) e)
    if (!inherits(try_error, "error"))
      y = tolower(x)
    return(y)
  }
  some_txt = sapply(some_txt, try.tolower)
  some_txt = some_txt[some_txt != ""]
  names(some_txt) = NULL
  return(some_txt)
}

clean_text = clean.text(Tweets.text)
```

### 3. Cargar en memoria el Lexicón (Diccionario de Palabras positivas y Negativas).

Una parte muy importante en la analice de sentimiento o opinión es encontrar o crear un lexicón con palabras que sean positivas y negativas.

En este trabajo es posible utilizar 3 lexicón, siendo un en inglés y dos en Portugués.

No ha sido posible encontrar un lexicón para el Portugués de Brasil y por esta razón se ha decidido el lexicón llamado SentiLex versión 1 y 2 del Portugués de Portugal. Las tres opciones de lexicón utilizadas en este trabajo son muy conocidas por la comunidad científica y muy utilizados para trabajos de analice de sentimiento.

Para este trabajo ha sido necesario crear un otro programa en python capaz de leer el SentiLex y preparar los diccionarios en el formato solicitado por R. Los códigos están disponibles en [github.com/caiomsouza](https://github.com/caiomsouza).

Después de preparados los diccionarios la próxima etapa en R es:

```
#3. Cargar en memoria el Lexicon (Diccionario de Palabras positivas y Negativas).

# en = Wordbank en inglés | pt01 = Wordbank en portugués con SentiLex-PT01 | pt02 = Wordbank en portugués con SentiLex-PT02
version <- "pt02";

if (version == "en") {

  # Wordbanks from https://github.com/mjheath/twitter-sentiment-analysis/tree/master/wordbanks
  pos = scan('wordbanks/positive-words.txt', what='character', comment.char=';')
  neg = scan('wordbanks/negative-words.txt', what='character', comment.char=';')
  head(pos)
  head(neg)

} else if(version == "pt01"){

  # SentiLex-PT01
  pos = scan('/Users/caiomsouza/git/Bitbucket/u-tad/final-project/src/r-script/SentiLex-PT01/pos-pt01.txt', what='character', comment.char=';')
  neg = scan('/Users/caiomsouza/git/Bitbucket/u-tad/final-project/src/r-script/SentiLex-PT01/neg-pt01.txt', what='character', comment.char=';')
  head(pos,20)
  head(neg,20)

} else if (version == "pt02") {

  # SentiLex-PT02
  pos = scan('/Users/caiomsouza/git/Bitbucket/u-tad/final-project/src/r-script/SentiLex-PT02/pos.txt', what='character', comment.char=';')
  neg = scan('/Users/caiomsouza/git/Bitbucket/u-tad/final-project/src/r-script/SentiLex-PT02/neg.txt', what='character', comment.char=';')
  head(pos,20)
  head(neg,20)

}
```

### 4. La analice de sentimiento.

Para determinar el sentimiento de un tweet es posible utilizar dos técnicas distintas: i) método basado en lexicón ii) método basado en aprendizaje de maquina (*machine learning*).

El método basado en aprendizaje supervisado utiliza un clasificador y un corpus y el método basado en lexicón es un aprendizaje no supervisado o semántico donde se utiliza un diccionario de palabras positivas y negativas.

Se conoce por medio de estudios anteriores de otros investigadores de la comunidad académica mundial que utilizar un clasificador basado en SVM y Naive Bayes sobresale la performance del método basado en lexicón, pero la ventaja del método basado en lexicón es no tener que crear corpus para cada dominio, ahorrando tiempo y creando un método más genérico para distintos dominios.

Es posible mejorar muchísimo la performance con un método ensamblado donde se utiliza una puntuación de sentimiento basado en el método de lexicón como una variable para el método de aprendizaje de maquina con SVM o Naive Bayes.

Actualmente se puede tener resultados positivos con los dos métodos. La precision obtenida con clasificadores multidominio (método basado en lexicon) son de 70 - 75% y la precisión en clasificadores específicos a partir del 80% (método baseado en aprendizaje de maquina y un corpus).

El principal recto actualmente en la investigación de la análisis de sentimiento es tratar la subjetividad? Esto es un gran recto mismo para las personas y para las maquinas aun más grande.

Este trabajo busca encontrar un método automático de clasificar el sentimiento de los tweets con la máxima precisión posible, pues con la gran cantidad de tweets para determinados asuntos es imposible hacer este trabajo manualmente. También es importante aclarar que el enfoque es hacer para el idioma portugués.

La solución encontrada ha sido aplicar la función score.sentiment para extraer el sentimiento del texto.

Esta función es genérica y se puede aplicar para cualquier texto en cualquier idioma lo que hace con que sea muy interesante debido a la flexibilidad encontrada.

Pero, es muy importante dejar aclarado que no es la mejor solución existente en el mundo y tampoco he podido comparar los resultados de este algoritmo con otros existentes en el mundo.

En este trabajo no ha sido posible hacer una investigación profunda sobre el estado de la arte de analice de sentimiento en el mundo y tampoco el estado de la arte de analice con mensajes en el idioma Portugués Brasileño.

Muchas veces, las empresas o instituciones publicas no hacen ningún tipo de analice de lo que se habla en el Twitter, o muchas veces lo hace de forma muy manual.

Este trabajo lo que busca es encontrar una forma de automatizar este proceso y dejarlo lo mas genérico posible mismo que esto tenga que sacrificar la calidad de la predicción del sentimiento.

Para proyectos donde le objetivo es tener una mejor predicción lo recomendable es crear un Corpus específico para el dominio de estudio y aplicar un algoritmo de

aprendizaje supervisado donde es posible conseguir resultados mejores, pero lo que deja la solución muy personalizada y poco genérica.

La función abajo lo que hace es separar en palabras, coger cada palabra y mirar en el Lexicón si es una palabra positiva o negativa.

Se cuenta la cantidad de palabras positivas y la cantidad de palabras negativas y se hace una cuenta simple para intentar predecir el sentimiento de el texto.

La formula final en R ha quedado:

```
score = sum(pos.matches) - sum(neg.matches)
```

Donde la puntuación (score) es disminuir la cantidad de la suma de palabras positivas encontradas en el texto con la suma de palabras negativas encontradas en el texto.

Los valores son negativos, positivos o cero.

Se puede visualizar los datos con los valores numéricos o clasificar los resultados en positivos, negativos o neutro aplicando una regla que puede ser definida por el usuario de la herramienta.

Se puede decir que valores más grandes que 1 o 2 son positivos, de -1 para bajo negativo y neutro lo que no es positivo o negativo, pero la predicción puede no ser de las mejores.

#### #4. Función para hacer la analice de sentimiento.

```
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
  require(plyr)
  require(stringr)

  # we got a vector of sentences. plyr will handle a list
  # or a vector as an "l" for us
  # we want a simple array ("a") of scores back, so we use
  # "l" + "a" + "ply" = "laply":
  scores = laply(sentences, function(sentence, pos.words, neg.words) {

    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    sentence = gsub('\\d+', '', sentence)
    # and convert to lower case:
    sentence = tolower(sentence)

    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, '\\s+')
    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words, .progress=.progress )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

El código abajo ejecuta la función y hace el calculo de el sentimiento.

```
analysis = score.sentiment(clean_text, pos, neg)
```

## 5. La visualización de los datos

La visualización de los datos es la parte mas interesante para el usuario de la herramienta y también la parte donde se puede invertir muchísimo tiempo para generar distintos tipos de visualización de los datos.

Por una limitación de tiempo no se ha podido llegar en todas las visualizaciones idealizadas en el momento del proyecto, pero si se ha podido llegar a algunas visualizaciones interesantes y capaces de permitir al usuario sacar algunas conclusiones de los datos.

Las visualizaciones han sido divididas en dos:

- 1) Visualizaciones utilizando la herramienta R;
- 2) Visualizaciones con la herramienta de BI (Business Intelligence) llamada Pentaho;

Visualizaciones utilizando la herramienta R;

La herramienta R posibilita maneras de visualizar los datos y sacar conclusiones muy interesantes.

Abajo ejecutaremos algunos comandos en R capaces de analizar los datos.

```
# Visualización
table(analysis$score)
mean(analysis$score)
hist(analysis$score)
colnames(analysis)
View(analysis)
```

```
table(analysis$score)

> table(analysis$score)

-6 -5 -4 -3 -2 -1  0  1  2
2 15 117 108 273 756 572 128 17
```

```
mean(analysis$score)
```

```
> mean(analysis$score)
[1] -1.015594
```

Se puede ver que la media es negativa, en general se “puede” decir que hay mas palabras negativas que positivas.

Para tener claro que estos valores son la realidad de la muestra de 2000 twitters, se recomienda el interesado que se debería mirar cada uno de los mensajes y hacer una clasificación manual de positivo, negativo y neutro

Por razones de tiempo y deseo, no se ha planteado en este trabajo hacer esta validación, lo que deja los resultados conseguidos mas difíciles de ser interpretados.

```
hist(analysis$score)
```

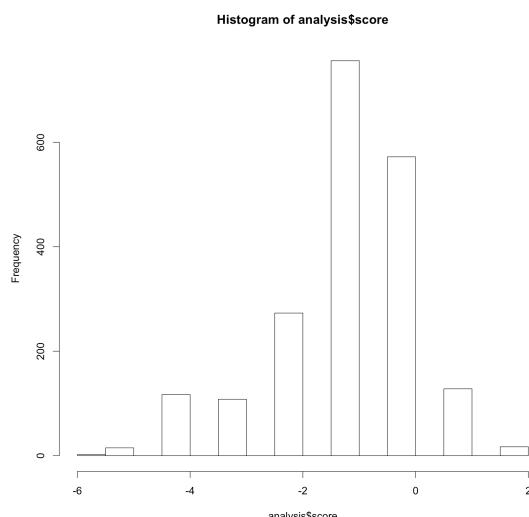


Figura: Histograma de la Puntuación (Score)

En histograma arriba se puede ver que en los 2000 mensajes la puntuación (score) tiene un rango de -6 a 2, que hay mas valores negativos o sea hay mas sentimiento negativo que positivo.

```
colnames(analysis)
```

```
> colnames(analysis)
[1] "score" "text"
```

```
View(analysis)
```

Se puede ver el texto y la puntuación (score).

	score	text
1	-2	rtmo gallo da madrugada não faltaram brincadeiras co...
2	0	rtdilma rousseff renuncia em agosto de
3	-1	rtgilmar mendes sobre janoadvogado de dilma
4	-1	rtate o reiespanha se chocou c gastos d dilma
5	0	rtdilma é a pior presidente da história da humanidade...
6	-2	no gallo da madrugada não faltaram brincadeiras com...
7	-2	rtempara pagar as pedaladas e evitar o impeachment...
8	-2	rtpuxa vida donaperdeu completamente juízo se for v...
9	0	rtisto é dilmapreso delcídio continua sendo líder de d...
10	1	rtse o twier começar a ordenarweets por popularidad...
11	-4	rtibope mostra rejeição devastadora ao governo dilm...
12	-2	riptwier a dilma que estãoe pagando para fazer isso
13	-1	vídeo impressionante mostra plateia virando de costa...
14	-1	hora de sacar dilma os petralhas e o pmdb do poder ...
15	0	uma ousadia de dilmaeditorialo estado de s paulovia
16	-3	enquanto o brasil batia panela contra o pt durante exi...

Showing 1 to 16 of 1,988 entries

```
Console ~/git/Bitbucket/u-tad/final-project/src/r-script/ ↵
> analysis = score.sentiment(clean_text, pos, neg)
Loading required package: stringr
> table(analysis$score)

-6 -5 -4 -3 -2 -1  0   1   2
 2 15 117 108 273 756 572 128 17
> mean(analysis$score)
[1] -1.015594
> hist(analysis$score)
> colnames(analysis)
[1] "score" "text"
> 2+15+117+108+273+756+572+128+17
[1] 1988
> colnames(analysis)
[1] "score" "text"
>
> View(analysis)
> |
```

## Nube de Palabras

Otra técnica muy interesante en la minería de datos es lo que se llama la nube de palabras.

Utilizando la herramienta R y el código abajo se ha creado la nube de palabras.

```

163 # Nube de Palabras
164 #install.packages(c("wordcloud","tm"),repos="http://cran.r-project.org")
165 library(tm)
166 library(wordcloud)
167 require(plyr)
168
169 tweet_corpus = Corpus(VectorSource(clean_text))
170 tdm = TermDocumentMatrix(tweet_corpus,
171   control = list(removePunctuation = TRUE,stopwords = c("machine", "learning", stopwords("english")),
172   removeNumbers = TRUE, tolower = TRUE))
173 m = as.matrix(tdm) #we define tdm as matrix
174 word_freqs = sort(rowSums(m), decreasing=TRUE) #now we get the word orders in decreasing order
175 dn = data.frame(word=names(word_freqs), freq=word_freqs) #we create our data set
176 wordcloud(dm$word, dm$freq, random.order=FALSE, colors=brewer.pal(8, "Dark2")) #and we visualize our data
177 png("~/git/Bitbucket/u-tad/final-project/src/r-script/CloudImageFeb16.png", width=12, height=8, units="in", res=300)
178 wordcloud(dm$word, dm$freq, random.order=FALSE, colors=brewer.pal(8, "Dark2"))
179 dev.off()
180

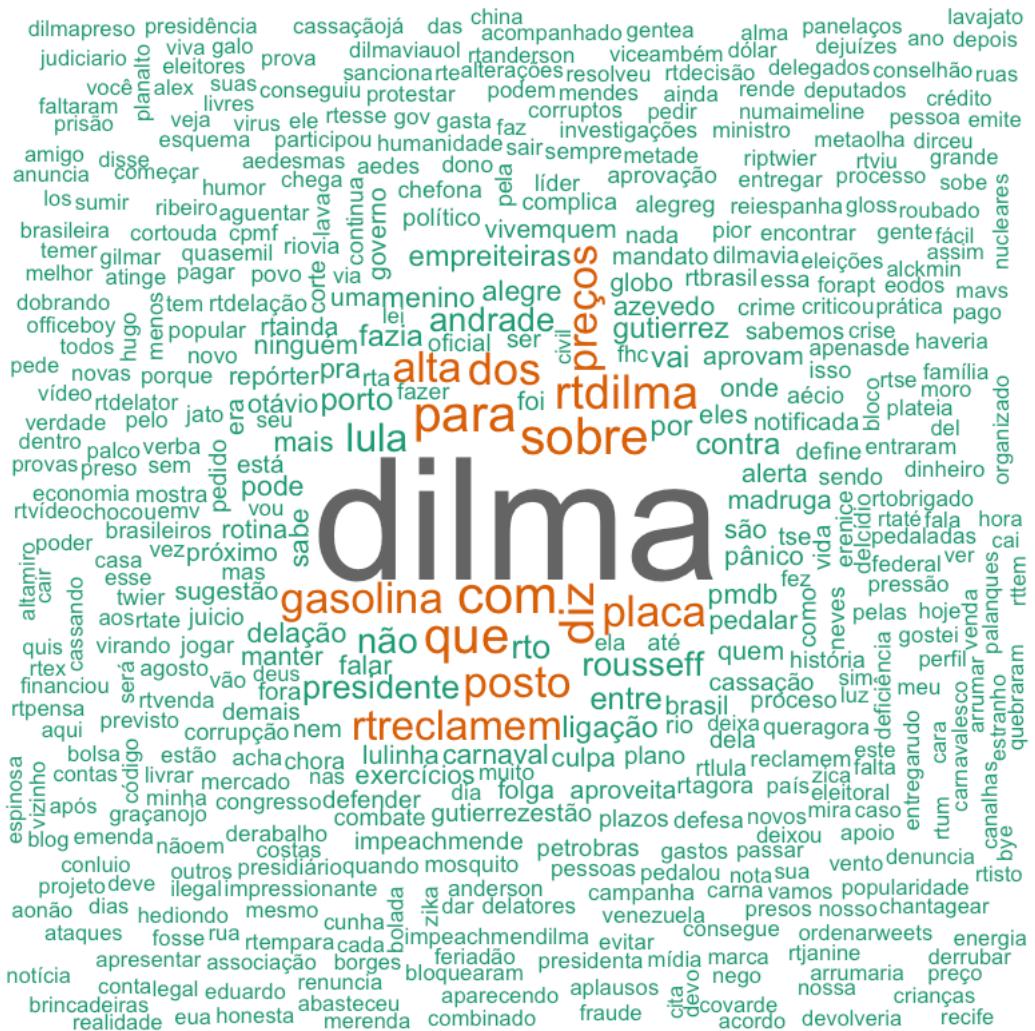
```

Los datos son de una muestra recogida en el día 6 de Febrero de 2016 donde los brasileños están en el periodo de Carnaval.

El Carnaval es una fiesta popular en el Brasil celebrada todos los años en todo el país, es una fiesta conocida internacionalmente y tiene gran impacto en el país.

Durante los días festivos de carnaval las personas suelen dejar de hablar de política, de sus problemas personales, de los problemas del país y intentan celebrar de todas las formas. Hay muchas personas no aficionadas por el carnaval que en estos días se quedan en casa con la familia o aprovechan para descansar de sus labores.

Es muy interesante mirar en la imagen abajo la nube de palabras y ver el sentimiento negativo que existe en las palabras muchas asociados a personas y eventos relacionados a corrupción o insatisfacción del pueblo.



## Visualizaciones con la herramienta de BI (Business Intelligence) llamada Pentaho;

La herramienta Pentaho ha sido utilizada para la creación de un cuadro de mando.

Los datos han sido almacenados en MySQL para posibilitar las consultas SQL ejecutadas por el cuadro de mando.

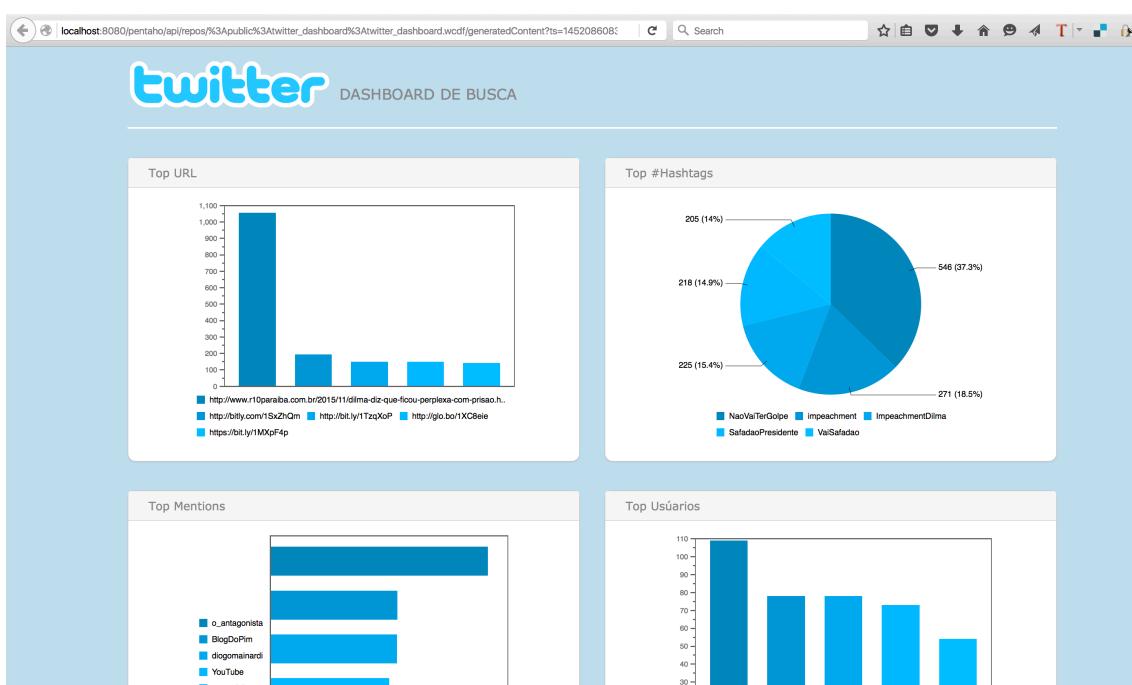
En la imagen abajo se puede ver con detalles las variables disponibles en un twitter que han sido almacenadas en el MySQL.

The screenshot shows the MySQL Workbench interface. On the left, there's a sidebar with sections like MANAGEMENT, INSTANCE, PERFORMANCE, and SCHEMAS. Under SCHEMAS, the 'dwcontrolpao' database is selected, and within it, the 'tweets' table is highlighted. The main area displays a 'Result Grid' with several rows of tweet data. A status bar at the bottom indicates '1 row(s) returned'.

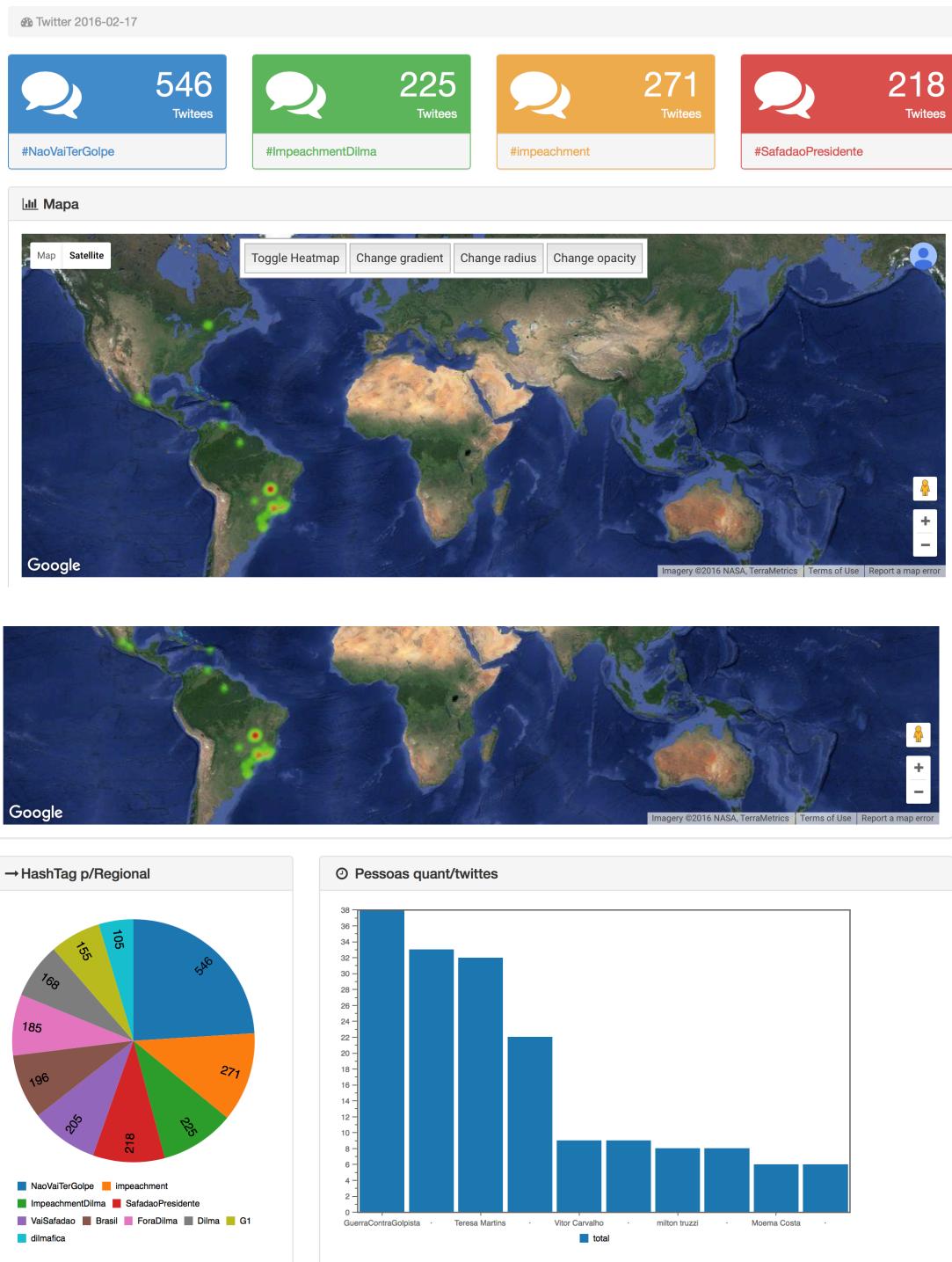
id	created	text	screen_name	name	mention
672441680826469305	Thu Dec 03 15:46:04 +0000 2015	Cunha acusa Dilma de mentir ao dizer que não faz... FacebookOFICIAL	Facebook		NULL
672441682143612928	Thu Dec 03 15:46:05 +0000 2015	RT @therealclubu: Outra delincuente em apuros: Dil... antiku4ever	Daniel - SDV	#NO-K - Vaselina!	NULL
672441682206580736	Thu Dec 03 15:46:05 +0000 2015	Quem está mentindo? ) Dilma ( ) Cunha (X) Os dois danielbdelima	Ercan Ates	RealErcanAtes1	NULL
672441683414532096	Thu Dec 03 15:46:05 +0000 2015	#677EI CIS #777Celi Villalobos #8??Aguirre #9... RealErcanAtes1	Mara Ivanovicht		NULL
672441685482205184	Thu Dec 03 15:46:05 +0000 2015	https://t.co/4ahaWBxtn https://t.co/KghISF2p2y Maralvanovicht	Daisy		NULL
672441686367281152	Thu Dec 03 15:46:06 +0000 2015	RT @toonda: dilma larga o PAC Derme programa... dashhde	Loulair1	#CORRUPTOFOBICO	toonda
67244168753239574	Thu Dec 03 15:46:06 +0000 2015	RT @LouLair1: @braZilnocommu Dilma mentiu... aeciofrajau	Alessandro Silveira	jemonuze	NULL
672441687587864577	Thu Dec 03 15:46:06 +0000 2015	RT @jemunozrivera: Dilma mentiu e tentou bargan... SilverOficial	Paula A Fermiano		Regis Cavalcante
672441688690831361	Thu Dec 03 15:46:06 +0000 2015	Cunha acusa Dilma de mentir ao dizer que não faz... willzinhow	PT Brasil		NULL
672441690079305728	Thu Dec 03 15:46:06 +0000 2015	ptbrasil: RT redebrasiliatua!: Deputados governistas... ptbrasil	FocoBrasil		NULL
672441690423205888	Thu Dec 03 15:46:07 +0000 2015	Tem que ser um ministro muito vagabundo, para a... Focaremudar	FurtadoEduardo		NULL
672441691492732928	Thu Dec 03 15:46:07 +0000 2015	CUNHA pede IMPEACHMENT de DILMA?? https://... duduugcfc	Paula A Fermiano	YouTube	NULL
672441691807223809	Thu Dec 03 15:46:07 +0000 2015	Dilma diz que recebeu com indignação abertura de... paulaapple	Regis Cavalcante	SouCalmo	NULL
672441694659440641	Thu Dec 03 15:46:08 +0000 2015	RT @SouCalmo: Mercado pensando diferente das... regiscavalcante			

Como se puede ver en la imagen abajo la visualización de los datos se puede hacer también por medio de cuadro de mandos hechos con la herramienta de BI Pentaho.

Se ha creado dos cuadros de mando para ayudar en la visualización de los datos.



## TwitterDashboard Statistics Overview



Se busca con la solución en el futuro, contestar de forma muy visual y sencilla las siguientes preguntas de negocio:

- ¿Quiénes son las personas con más seguidores con la mayor cantidad de mensajes negativos?
- ¿Quiénes son las personas con más seguidores con la mayor cantidad de mensajes positivos?
- ¿Quiénes son las 10 personas con el mayor número de seguidores con una mayor cantidad de mensajes negativos?
- ¿Quiénes son las 10 personas con el mayor número de seguidores con una mayor cantidad de mensajes positivos?
- ¿Cuáles son los mensajes más relevantes? (La medida de relevancia se hará a través de la cantidad de *retweets* hechos. Es decir, cuanto más menciones, o *retweets*, más relevante el mensaje).

También es un objetivo en la continuación de este trabajo:

- Crear nuevos cuadros de mandos con Pentaho;
- Mejorar la solución actual para tener mejor performance con grandes cantidades de datos;
- Estudiar el estado de la arte de la analice de sentimiento con mas tiempo y detalle y mejorar el algoritmo actual de predicción del sentimiento.
- Quitar las Stop Words de la Nube de Palabras;
- Probar con un clasificador como Naive Bayes y SVM;
- Utilizar métricas como Precisión, Recall y la Matrix de Confusión para medir la calidad del clasificador;
- Importar y utilizar todos los datos colectados y publicados en <https://github.com/caiomsouza/TwitterRawData/releases/tag/DITRD-v1.0.0>

## Lecciones aprendidas y resultados

El resultado del trabajo ha sido para mi satisfactorio, visto que el tema abordado es muy complejo. No se puede plantear en pocos meses dedicando tiempo parcial a profundarse en un tema tan amplio.

Las lecciones aprendidas han sido, que si es posible sacar muchas conclusiones de los datos de twitter, que si es posible, tener una idea general del sentimiento de las mensajes, que si es posible identificar los principales usuarios que tengan un sentimiento positivo o negativo, se puede identificar la localidad de los mensajes y muchísimas mas informaciones.

Actualmente existen muchas empresas y personas dedicadas al tema de la minería de textos, análisis de sentimiento aplicado a los datos de twitter y el objetivo del trabajo no ha sido estudiar el estado de la arte y si intentar en poco tiempo juntar en un trabajo temas como: minería de textos + análisis de sentimiento + business Intelligence + datos de twitter + geo localización + un tema importante de la actualidad como es el tema del proceso de impeachment en el Brasil.

No se ha medido la calidad del clasificador y tampoco se ha probado todos los métodos que se debería probar para tratar un tema como el tema de la análisis de sentimientos.

Se desea en el futuro mejorar la evaluación formal del clasificador utilizando métricas como precisión, recall, la matrix de confusión, etc.

Por tratarse de una base de datos muy delicados, he decidido no utilizar una técnica muy importante en la área de los científicos de datos llamada Storytelling. Contar una historia con estos datos podría generar reacciones de ambos los grupos: los a favor del impeachment y los que son contra. Dejo entonces las interpretaciones de los datos para mi uso personal y publico mi trabajo con los códigos y datos para que cada persona pueda sacar sus conclusiones personales de los datos.

Si, se puede contar historias con los datos.

## 7. Cronograma

Abajo el cronograma del TFM.

Tareas	% de conclusión	Mes 1			Mes 2				Mes 3			Mes 4				Mes 5					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Estudiar la API de Twitter para entender como extraer datos en tiempo real filtrando por palabras claves deseadas	0%	X	X																		
Extraer los datos de Twitter y grabar en un archivo json	0%	X	X	X	X	X	X	X	X	X	X	X	X	X	X						
Hacer el parser del archivo JSON y grabar en una base de datos MySQL	0%	X	X	X	X	X	X	X	X	X	X	X	X	X	X						
Crear visualizaciones con los datos originales	0%	X	X	X	X	X	X	X	X	X	X	X	X	X	X						
Encontrar un Lexicon en Portugués	0%	X	X	X	X	X	X														
Crear un script en Python para separar el Sentilex 1.0 e 2.0 en dos archivos (pos.txt y neg.txt)	0%																				
Crear una función en R para limpiar los datos sucios de los mensajes de twitter	0%																				
Crear una función en R para contar la frecuencia de palabras negativas y positivas presentes en lo mensaje y definir el sentimiento	0%																				
Crear una función en R para hacer la nube de palabras	0%																				
Crear un cuadro de mando para visualizar los indicadores	0%					X	X	X	X	X	X	X	X	X							
Ejecutar la carga y procesamiento final con todos los datos colectados y generar las análisis del estudio	0%					X	X	X	X	X	X	X	X	X							

## 8. Periodo y Lugar

El trabajo ha sido desarrollado en Madrid, España de Octubre de 2015 a 17 de Febrero de 2016.

## 9. Bibliografía

Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. Computational Linguistics, 2011. 37(2): p. 267-307.

Olga Kolchyna, Tharsis T. P. Souza, Philip Treleaven, Tomaso Aste. Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination. (<http://arxiv.org/abs/1507.00955>)

Mário J. Silva, Paula Carvalho and Luís Sarmento. "Building a Sentiment Lexicon for Social Judgement Mining". In Lecture Notes in Computer Science (LNCS) / Lecture Notes in Artificial Intelligence (LNAI), International Conference on Computational Processing of Portuguese (PROPOR), 17-20 April, 2012, Coimbra.

## 10. Anexo

Se encuentra en mi cuenta de <https://github.com/caiomsouza/u-tad-eds-proyecto-final> algunos de los códigos utilizados para el trabajo, con el tiempo tengo la intención de publicar todos los códigos, en este momento no se publica por una cuestión de no tener tiempo para publicar de una forma organizada.