

1. Resumen Ejecutivo

La elaboración de un modelo de determinación de opinión a partir de un corpus se puede definir como la selección de características de representación de las opiniones (elección de tópicos, recursos estilísticos, análisis morfológicos y semánticos), como input a un modelo de aprendizaje. Los métodos de ensamble demuestran ser una herramienta eficaz para la detección de clase en matrices obtenidas de análisis de textos.

2. Objetivo

Se busca elaborar un modelo de determinación de opinión, mediante la extracción de rasgos de un corpus conteniendo columnas de opinión de dos medios periodísticos distintos. A través de la aplicación de un método de ensamble, se pretende evaluar la performance de la combinación de métodos débiles ante la determinación de la clase de un set de textos sin clase asignada.

3. Arquitectura de la solución

Se cuenta con dos corpus (1253 instancias), que representan una muestra del 80% de un corpus más grande.

- “corpo”, conteniendo 551 archivos planos, cada uno correspondiente a un artículo de opinión de La Nación.
- “k”, conteniendo 702 archivos planos, cada uno correspondiente a un artículo de opinión de Página 12.
- Adicionalmente se recibieron los mismos archivos procesados por “Freeling”¹. Se utilizaron para extraer las etiquetas

Eagle² para obtener información morfológica (archivos “corpo-free” y “k-free”).

Software utilizado para procesamiento de textos y armado de la matriz de features: R

Paquetes utilizados: tm, stringr, RWeka, Rjava, Wordcloud.

Software utilizado para algoritmos de clasificación y ensamble: Weka (Knowledge Flow, Explorer y línea de comandos)

3.1. Selección de features:

Para poder determinar la autoría de un texto de opinión, se deben seleccionar ciertas “características de la representación de las opiniones” o “features” de los artículos, que a través de la aplicación de un algoritmo de clasificación (o varios), permitan determinar si un texto nuevo sin clasificar pertenece a La Nación o a Página 12.

Se decidió aplicar un análisis de elementos propios de los corpus, y de elementos externos (diccionarios o bolsas de palabras), para tener parámetros que no produjeran sobreajuste, es decir, independientes de los textos en sí. Se proceden a detallar los features elegidos³:

3.1.1. Features por tópico:

Para la generación de diccionarios se tuvieron en cuenta los distintos temas sobre los que versan las columnas de opinión (generalmente política, economía, en menor grado deporte). Se pueden dividir las “bolsas de palabras” de la siguiente manera:

² Etiquetas Eagles: etiquetas propuestas por el grupo Eagle para la anotación morfosintáctica de lexicones y corpus para todas las lenguas europeas (<http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html>)

³ Estructura de la solución en Anexo IV

¹ Librería open source de analizadores de procesamiento de lenguaje natural.

3.1.1.1. De sentimiento:

Si bien es difícil asociar una palabra fuera de su contexto con un sentimiento, se tomó como punto de partida varios diccionarios de sentimiento en inglés, se tradujeron los términos, y se seleccionaron aquellos más comunes en el habla española, adaptando algunos que contenían N-gramas. De este análisis surgieron 5 bolsas:

B1: Sentimiento Negativo
B2: Sentimiento Positivo
B3: Sentimiento Litigioso
B4: Modal Fuerte
B5: Modal Débil

Si bien las palabras por sí solas no pueden determinar si la opinión es “positiva” o “negativa”, es una aproximación sobre el tipo de escritor, si aporta sentimiento o tiene una visión “objetiva” del tema que está tratando. Para analizar en profundidad el sentimiento del artículo se debería aplicar un análisis a través de N-Gramas (se seleccionaron algunos y se incluyeron en la bolsa), o teniendo en cuenta las palabras que tengan alta correlación con aquellas listadas como negativas, también se podría estimar la orientación semántica de frases extraídas (para futuros análisis).⁴

3.1.1.2. Política

Se procesaron en R discursos con distintas orientaciones, y se consultaron libros de textos relacionados con la política y el ámbito del poder. De los discursos seleccionados se extrajeron las palabras con mayor frecuencia (dependiendo de la cantidad de palabras totales, se seleccionó un

límite inferior, previamente filtrando las “palabras vacías”⁵).

B6: Perón: procesamiento de discursos entre 1945 y 1974.⁶

B7: Del Poder: elección de palabras claves del libro “Las 48 leyes del poder”⁷.

B8: Néstor Kirchner: procesamiento de discursos entre 2003 y 2007.⁸

B9: Bartolomé Mitre: procesamiento de Arengas de Bartolomé Mitre desde 1848 hasta 1902.⁹

3.1.1.3. Economía

B11: Textos de economía K (fuentes: artículos de página 12; discurso de C.F. de Kirchner; textos de: Panigo Neffa; Santarcangelo; Pinazo; Ortiz y Schorr; Cenda y Kiciloff¹⁰).

B12: Textos de economía no K (fuentes: artículos de Clarín, FortunaWeb, Perfil, textos de Prat Gay, Lindemboim Javier, Kennedy Damián, Graña Juan)¹¹

3.1.1.4. Varios

B10: Deporte: procesamiento de artículos de El Gráfico y Olé digital (selección de notas desde enero 2012 hasta junio 2012¹². A estos términos se agregaron palabras comunes del ámbito del fútbol.

B13: Lanata: Argentinos-Tomo I y II-Ediciones B

⁵ Palabras sin significado como artículos, pronombres, preposiciones, que son filtradas antes del procesamiento de textos.

⁶ <http://www.pjmoreno.org.ar/documentos/discursosperon.aspx>

⁷ Las 48 Leyes del Poder-Robert Greene, Joost Elfers- Editorial ATLANTIDA-Edición Número 4

⁸ <http://www.elhistoriador.com.ar>

⁹ <http://archive.org/details/arengasdebartolo02mitr>

¹⁰ Ver detalle en Anexo III

¹¹ Ver detalle en Anexo III

¹² <http://www.elgrafico.com.ar/>

<http://www.ole.com.ar/>

⁴ <http://www.cs.uic.edu/~liub/teach/cs583-spring-07/opinion-mining.pdf>

B14: Palabras que aparecen en los textos “Corpo” y no en los textos “K” (se generaron las palabras más frecuentes en “Corpo” y en “K” mediante un script de R y se buscaron los términos que aparecían en un corpus y no en el otro.
 B15: Palabras que aparecen en los textos “K” y no en los textos “Corpo”.¹³

3.1.2. Features por características estilométricas:

Se eligieron como características de estilo las siguientes cuentas: cantidad de líneas por artículo, cantidad de párrafos por artículo, cantidad de palabras por artículo, cantidad de caracteres por artículo.

3.1.3. Features por clasificación semántica:

Según la clasificación semántica, se seleccionaron los siguientes tipos de palabras:

Adjetivos: se eligieron adjetivos por entender que los artículos de opinión no suelen ser “neutros” (supuesto), por lo tanto el tipo de adjetivo que acompaña los sustantivos, puede ser un buen indicador en algunos casos.

Determinantes: también indican un sentimiento, su ausencia puede denotar cierta neutralidad.

Verbos: los verbos que se consideraron “indicativos” para este análisis son los del modo condicional.

Nombres propios: se buscaron nombres propios de personas, de ciudades y de organizaciones.

Signos de puntuación: se seleccionaron aquellos signos que denoten algún sentimiento, como admiración, o exclamación. También comillas para analizar quienes citan más que otros. El signo de puntuación se utilizó para contar cantidad de oraciones por artículo.

4. Armado del experimento

De las 63 features resultantes, se arma una matriz, con tantas filas como artículos de opinión, y tantas columnas como features seleccionadas, más la clase (“C”=Corpo, “K”=oficialista).

Debido a la diferencia en escala de los distintos artículos (cantidad de palabras), se relativizan las columnas para que los valores sean comparables, generando una nueva matriz que será el input del experimento. Se tomaron en cuenta varias opciones para proceder a esta relativización, pero si se optaba por dividir las columnas por distintos indicadores, el lenguaje sería más específico, y el experimento estaría más sesgado. Se decidió dividir todas las features por cantidad de palabras. Se realizó una tabla de contingencia para determinar si los textos son independientes de las bolsas, y se detectó una relación de dependencia.

Esta nueva matriz se introduce en el knowledge flow de Weka, utilizando el “Train Split Maker”¹⁴ para separar la matriz en training y test. Se obtienen como resultado dos archivos, training (70%) y test (30%). Para asegurarse de la no influencia de la aleatoriedad en los resultados, se hicieron también las pruebas con validación cruzada con 10 carpetas.

Se aplica por línea de comando un método de ensamble (votación)¹⁵. Para elegir los algoritmos que van a participar del voto, se realizan pruebas en tres pasos:

- 1) Prueba de performance de algoritmos solo variando sus parámetros (sin boosting). Se elige la combinación que genera el mayor porcentaje de instancias correctamente clasificadas. Para encontrar los mejores parámetros se utiliza el

¹³ Ver nube de palabras más frecuentes de cada medio en el Anexo I

¹⁴ Pantalla de Knowledge Flow en Anexo II

¹⁵ Scripts en Anexo II

- algoritmo CV Parameter Selection.
- 2) Se corren los algoritmos con Adaboost, bajo los parámetros elegidos en (1), y se evalúa su performance comparativamente con el desempeño anterior. Se elige boosting como método de ensamble, porque según la bibliografía consultada para el análisis de textos es más eficaz la estrategia de boosting que la de bagging.
 - 3) Se eligen los algoritmos que mejor performance tuvieron, para votar. Se evalúa la performance de Voto mayoritario vs. Voto promedio, eligiendo el de mejor porcentaje de instancias correctamente clasificadas.

Elección de algoritmos:

Las técnicas más comunes para text mining, visto como técnicas de aprendizaje automático aplicadas al análisis de textos, son k-means clustering, Naive Bayes, SVM, vectorización td-if.

Se prueba la combinación de los siguientes algoritmos disponibles en Weka: Ibk (Vecino más cercano), Naive Bayes y J48 (árboles) para generar el modelo de ensamble.

5. Pruebas de eficiencia y análisis de resultados.

Para comprobar la eficiencia de la prueba, se realizan corridas sin algunas columnas (features) que pueden producir sobreajuste al modelo (por ejemplo las columnas correspondientes a las bolsas 14 y 15, que son términos exclusivos utilizados por un medio, que no se repiten en el otro, y que pueden generar una performance engañosa). Se analizan los resultados con validación cruzada.

Resultados obtenidos:

Meta Clasificador	Parámetro	Learner	A	B	C
CV Parameter Selection	"C 0.1 0.5 5.0"	J48 0.3	99.36%	90.50%	84.83%
AdaBoostM1	-P 100 -S 1	J48 0.3	99.60%	93.54%	85.71%
		Naive Bayes	97.61%	64.51%	61.05%
AdaBoostM1	-P 100 -S 1	Naive Bayes	99.20%	75.58%	62.33%
CV Parameter Selection	'-K' ranged from 1.0 to 10.0 with 10.0 steps	IBK k=7	94.41%	82.12%	75.98%
AdaBoostM1	-P 100 -S 1	IBK k=7	92.18%	81.40%	75.98%
Vote	Mayoría	AdaBoostM1 (J48)- AdaBoostM1 (Naive Bayes)- IBK(K=7)	99.68%	90.50%	80.21%
Vote	Promedio	AdaBoostM1 (J48)- AdaBoostM1 (Naive Bayes)- IBK(K=7)	99.60%	92.82 %	83.64 %

Cuadro 1. Performance comparativa de algoritmos de clasificación vs. métodos de ensamble

A: Matriz completa con los 63 atributos

B: Matriz sin las bolsas 14 y 15 (palabras exclusivas Corpo y K).

C: Matriz solo con bolsas 1 a 13

Teniendo en cuenta los resultados generales, se decide elegir la opción de voto promedio, con la combinación AdaBoostM1 (J48)-AdaBoostM1 (Naive Bayes)- IBK(K=7). La peor performance es la de Naive Bayes sin boosting, y la mejor teniendo en cuenta los casos en que clasifica correctamente con las tres estructuras, es el método de votación promedio.

Conclusiones

¿Cuál es la diferencia entre procesar textos y procesar cualquier otro tipo de datos, cuando el objetivo es la detección de opinión, a través de la aplicación de algoritmos de clasificación? Para responder esta pregunta, se analiza el paso a paso del proceso de aprendizaje. En text mining se procede a la limpieza de los textos (remover puntuación, sacar palabras vacías, por ejemplo), al igual que en otro tipo de datos se procede a limpiar los datos antes de su procesamiento.

En los análisis de variables numéricas hay correlación entre variables (y resulta que en text mining también existe un índice de correlación entre palabras). Selección de features: en los análisis “comunes”, hay que elegir entre variables cuáles son indicativas para realizar la “predicción”, y lo mismo sucede para text mining. Selección de modelos y ensambles: se trata de buscar el algoritmo con mejor performance, incluso la combinación de ellos puede llevar a una performance aún superior. Lo mismo se aplica en text mining, y dependiendo del tipo de textos, funcionará mejor un algoritmo que otro. Lo mismo que con las variables “clásicas”. Se concluye que para la detección de opinión son aplicables las mismas técnicas que para cualquier otro tipo de datos, con algunas particularidades, pero en líneas generales, es una aplicación más de aprendizaje automático.

Es interesante los resultados tan altos de performance alcanzados con los

métodos de ensamble, cuando se realice la prueba con los archivos de competencia, habría que comprobar si hubo un sobreajuste por las variables “internas” del texto seleccionadas.

Ideas para futuros desarrollos

Se planeó agregar N-Gramas conteniendo las palabras antes y después de la palabra Cristina, y bolsas externas pertenecientes a economistas clásicos. A su vez, se procesaron discursos de Raúl Ricardo Alfonsín y de la UCR, pero se decidió por simplicidad y performance de la matriz en cuanto a cantidad de columnas, dejarlo fuera de este análisis, así como las bolsas de palabras más frecuentes de La Nación y de Página 12, para no sobreajustar en demasía el modelo.

Otro análisis más profundo a aplicar en opinión mining, es el desarrollo de algún algoritmo para detectar sentimiento, por la concatenación de palabras, más complejo que un simple diccionario positivo y negativo como el que se aplicó en este experimento.

Los algoritmos se aplicaron bajo la restricción de ser algoritmos presentados durante la materia. Sería interesante ver cómo reacciona el experimento ante otro tipo de algoritmos (como SVM, Redes Neuronales). La base se pudo haber dividido en sub-matrices, y aplicar distintos algoritmos a cada matriz, y luego votar, pero por recomendación de los profesores, se dejó todo en una misma matriz.

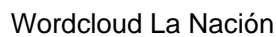
Bibliografía y referencias:

Witten, I.H. and Frank, E. (2005) *Data mining: Practical machine learning tools and techniques*. (second edition). Morgan Kaufmann, San Francisco, CA.

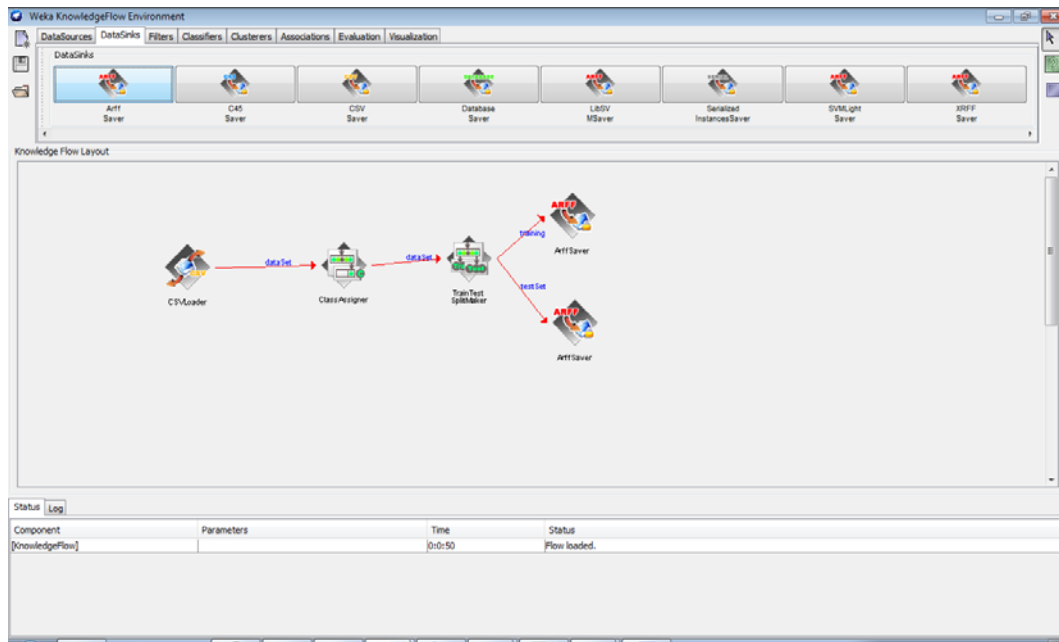
<http://www.public.asu.edu/~jye02/CLASSES/Fall2005/PAPERS/dietterich00ensemble.pdf>

[illegible]

Maestría en Explotación de Datos y Descubrimiento de Conocimiento FCEN/UBA



Anexo II-a. Pantalla de Knowledge Flow para separación en training (70%) y test (30%).



Anexo II-b. Scripts

1. Armado de bolsa

```
Corpus <- Corpus(DirSource("DiscursosNK"), readercontrol
=list(reader=readPlain, language="es", load=TRUE))
# lleva a minúsculas
d <- tm_map(corpus, tolower)

# quita espacios en blanco
d <- tm_map(d, stripWhitespace)

# remueve la puntuación
d <- tm_map(d, removePunctuation)

# remueve palabras vacías genericas
d <- tm_map(d, removeWords, stopwords("spanish"))

# crea matriz de terminos
tdm <- TermDocumentMatrix(d)

m <- as.matrix(tdm)

v <- sort(rowSums(m), decreasing=TRUE)

df <- data.frame(word = names(v), freq=v)

write.table <- (df, file="BOLSA_NK.csv", quote = FALSE, sep = " ")
```

2. Búsqueda de bolsa en corpus

```
B1=readLines("BOLSA_NK.csv")
```



```

corpus <- Corpus(DirSource("corpo-
sinfree"),readercontrol=list(reader=readPlain,language="es",load=TRUE))
# lleva a minúsculas
corpus <- tm_map(corpus, tolower)

# quita espacios en blanco
corpus <- tm_map(d, stripWhitespace)

# remueve la puntuación
corpus <- tm_map(d, removePunctuation)

d <- Dictionary(c(B1))
i= inspect(DocumentTermMatrix(corpus,list(dictionary=d)))
write.table(i,file="C_B7_NK.csv", quote = FALSE, sep = " ")

```

3. Búsqueda de etiquetas Eagle en corpus

Ejemplo Adjetivos:

```

ctofind <- c(" AQ0CP0 ")
files <- as.character(list.files(path="corpo", full.names = TRUE))

t <- c("File","Count")
for (i in 1:n){

  c1 <- files[i]
  c2 <- str_count(readLines(files[i]),ctofind[1])
  t <- rbind(t, t(c(c1,sum(c2))))
}

sink ("C_AQ0CS0.csv")
write.table(t, quote=FALSE, sep=",")
sink()

```

4. Búsqueda de signos de puntuación en corpus

Ejemplo: " (comillas)

```

ctofind <- c("[\""]")

files <- as.character(list.files(path="corpo-sinfree", full.names =
TRUE))

n <- length(files)

t <- c("File","Count")
for (i in 1:n){
  c1 <- files[i]
  c2 <- str_count( readLines(files[i]), ctofind[1])
  t <- rbind(t, t(c(c1,sum(c2))))
}

sink("C_comillas.csv")
write.table(t, quote=FALSE, sep=",")
sink()

```

5. Generación de Matriz para input del modelo

Se adjunta link con script completo, se copia el inicio del mismo a modo de ejemplo:

<https://docs.google.com/file/d/0B2i3yGAL-A7LVWt0Z0hINW9jNGs/edit>

```

#Setear directorio y llamar el path

setwd("G:\\Dropbox\\Maestria\\ML\\Tp2\\Train")

library(stringr)

#Leemos archivos corpo y sacamos nombres de archivo y clase en
vectores

#pth = ".\\corpo"

setwd("G:\\Dropbox\\Maestria\\ML\\Tp2\\Train\\corpo")

files <- as.character(list.files( full.names = TRUE))

n <- length(files)

fileNames_c <- files

class_c <- NULL

for (i in 1:n){
  class_c[i] <- "C"
}
#idem para k
#Limpiamos los nombres de los archivos

library(stringr)

ctofind <- c("./")

n <- length(fileNames_c)

for (i in 1:n){
  new_t <- str_replace(fileNames_c[i],ctofind[1],"")
  fileNames_c[i] = new_t
}

n <- length(fileNames_k)

for (i in 1:n){
  new_t <- str_replace(fileNames_k[i],ctofind[1],"")
  fileNames_k[i] = new_t
}

#Path de las bolsas

setwd("G:\\Dropbox\\Maestria\\ML\\Tp2\\Bolsas\\")

#Bolsa1

temp = as.matrix(read.table("C_B1_NEGATIVA.csv", skip = 1, sep = " "))

```

```

Bolsa1_c <- NULL #Declaramos vector nulo

for (i in 1:length(temp[,1])){ #recorremos el archivo
  if (fileNames_c[i] == temp[i,1]) { #Comparamos los nombres de
    los archivos
      Bolsa1_c[i] <-
sum(as.numeric(as.character(temp[i,2:length(temp[i,])])))) #Cuando
coinciden los nombres de los archivos sumamos toda la fila excluyendo
la columna del nombre y lo guardamos en la posicion que corresponda
segun el nombre del archivo en el vector que generamos en nulo
    }
  }

#Se repite el proceso para el archivo de K,y para todas las bolsas
#Path para las cuentas

setwd("G:\\\\Dropbox\\\\Maestria\\\\ML\\\\Tp2\\\\Bolsas\\\\Cuentas\\\\")

#Exclamation

temp = as.matrix(read.table("C_admiracion.csv", skip = 2, sep = ","))

Exclam_c <- NULL

for (i in 1:length(temp[,1])){
  if (fileNames_c[i] == temp[i,1]) {
    Exclam_c[i] <-
as.numeric(as.character(temp[i,2:length(temp[i,])]))
  }
}

temp = as.matrix(read.table("K_admiracion.csv", skip = 2, sep = ","))

Exclam_k <- NULL

for (i in 1:length(temp[,1])){
  if (fileNames_k[i] == temp[i,1]) {
    Exclam_k[i] <-
as.numeric(as.character(temp[i,2:length(temp[i,])]))
  }
}

#se repite para todo el resto de cuentas (adjetivos, verbos,
determinantes, nombres y signos de puntuación).
#Concatenamos vectores

fileNames <- c(fileNames_c, fileNames_k)

classes <- c(class_c, class_k)

Bolsa1 <- c(Bolsa1_c, Bolsa1_k)

Bolsa2 <- c(Bolsa2_c, Bolsa2_k)

Bolsa3 <- c(Bolsa3_c, Bolsa3_k)

Bolsa4 <- c(Bolsa4_c, Bolsa4_k)

Bolsa5 <- c(Bolsa5_c, Bolsa5_k)

```

```

Bolsa6 <- c(Bolsa6_c, Bolsa6_k)

Bolsa7 <- c(Bolsa7_c, Bolsa7_k)

Bolsa8 <- c(Bolsa8_c, Bolsa8_k)

Bolsa9 <- c(Bolsa9_c, Bolsa9_k)
#hasta tener los 64 vectores.
#Generamos la matriz final

mFinal <- data.frame(fileNames,Bolsa1,Bolsa2,Bolsa3 ...)#se adjuntan
todos los vectores
write.csv(mFinal, file = "MatrixFinal.csv", row.names=FALSE)
write.csv(mWeka, file = "MatrixWeka.csv", row.names=FALSE)se guarda la
tabla input.

```

6. Algoritmo de ensamble para predecir clase (entrenamiento)

```

#cargo modelo de voto promedio con cv
java -classpath "C:\Program Files\Weka-3-6\weka.jar"
weka.classifiers.meta.Vote -S 1 -B "weka.classifiers.meta.AdaBoostM1 -
P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.3 -M 2" -B
"weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W
weka.classifiers.bayes.NaiveBayes --" -B "weka.classifiers.lazy.IBk -K
7 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A
\\\\weka.core.EuclideanDistance -R first-last\\\\\" -R AVG -t
"C:\input.arff" -x 10 -d "C:\model_vote.arff" | find "Correctly
Classified Instances" > "C:\Users\Ana\Dropbox\trainingSetCCI_tp2.txt"

```

7. Test del modelo para el día de la competencia

```

#cargo test ficticio para probar el modelo y ver clases predichas
java -classpath "C:\Program Files\Weka-3-6\weka.jar"
weka.classifiers.meta.Vote -l "C:\model_vote.arff" -T "C:\
MatrixWeka.arff" -p 0 > "C:\Users\Ana\Dropbox\pruebapredict.txt"

```

Anexo III – Referencias textos de economía

Textos K

Página/12: Anses gana el primer round, 26/07/2012

Página/12 :CFK trató de "mezquinos" a los bancos privados, 23/07/2012

Página/12: "El ladrillo como reserva de valor" 27/07/2012

Página/12: "Es un cacareo sin sentido" 21/07/2012

Página/12: "Es una cuestión marginal" 21/07/2012

Página/12: Kicillof: "El Estado está a la cabeza de la reactivación de la economía" 20/07/2012

Página/12: "Eva volvió en millones de puestos de trabajo" 27/07/2012

Página/12: "Hay una cadena nacional del miedo" 24/07/2012

Página/12: Los economistas en su sopa 28/07/2012

Página/12: Marca a presión a los bancos para que presten 24/07/2012

Página/12: El mapa de los subsidios 28/07/2012

Página/12: Un homenaje que estará en mano de todos 26/07/2012

Página/12: YPF y Pdvsa, juntas para llenar el tanque 25/07/2012

Discurso Cristina Fernandez de Kirchner: Proyecto de Ley de Expropiación de YPF 13/04/2012

Panigo, Neffa, El mercado de trabajo argentino en el nuevo modelo de desarrollo

Santarcangelo, Pinazo Los motores del crecimiento económico en la Argentina

Ortiz Y Schorr, Economía de Alfonsín y crisis

CENDA (2010). El trabajo en Argentina Condiciones y perspectivas, informe trimestral nro. 19

Panigo, El impacto en la Asignación Universal por Hijo en Argentina

Kicilloff, Costa y Nahón Consecuencias económicas del Señor Lavagna. Dilemas de un país devaluado.

Textos no K:

Perfil: Prat Gay: "Este Gobierno devalúa hasta a Evita" 26/07/2012

Perfil.com: Axel Kicillof habló de los economistas "mala onda" 20/07/2012

FortunaWeb: España dice que expropiación de YPF le sirve a los políticos más que a la gente 27/06/2012

Perfil.com: El fifty-fifty de Perón, Moyano y CFK 26/06/2012

FortunaWeb: Crecen los consumos con tarjeta de crédito en el exterior 20/07/2012

Perfil.com: Cayó la actividad económica por primera vez en 33 meses 20/07/2012

FortunaWeb: Ex presidente del BCRA: "A Cristina se le fue de las manos el modelo" 27/06/2012

FortunaWeb: AFIP evalúa aumentar controles a las tarjetas de crédito 20/07/2012
Lindenboim, Javier; Kennedy, Damián; Graña, Juan. Distribución, consumo e inversión en la Argentina a comienzos del siglo XXI

Clarín: Repsol pone en marcha juicios en EE.UU. y el CIADI contra Argentina 16/05/2012

Clarín: Plata de los jubilados: el 28% financia gastos de otras áreas 24/07/2012

Perfil.com: Para exdirector del INDEC, la Argentina está en peligro de recesión 23/07/2012

Clarín: Otro paso en la pesificación: ahora el mercado de futuros 28/07/2012

Lindenboim El mercado de trabajo en Argentina en la transición secular Cada vez menos y peores empleos

Perfil: La clase media, blanco de la nueva etapa del modelo económico de CFK 15/07/2012

Clarín: Fuerte intervención oficial en el sector del petróleo 28/07/2012

FortunaWeb: Exportaciones: el efecto búmeran 20/07/2012

FortunaWeb: El 62% de empleados de super de Capital y GBA con irregularidades 25/06/2012

Lindenboim, Grana y Kennedy Distribución funcional del ingreso en Argentina. Ayer y hoy

Graña y Kennedy salario real, costo laboral y productividad : Argentina 1947-2006 : análisis de la información y metodología de estimación

Perfil: Sin viento de cola... ni barco 14/07/2012

FortunaWeb: Nunca se pagó tanto en impuestos: más del 35% del PBI 19/07/2012

Perfil.com: CFK pesificó su plazo fijo de U\$S 3 millones 11/07/2012

Perfil: AFIP agudiza la lupa sobre grandes consumidores de servicios 12/07/2012

Anexo IV-Estructura de la solución

