Selección de Características para Análisis de Sentimientos Basado en Computación Evolutiva: Resultados Preliminares

Neftalí David Watkinson Medina[a], Carlos Alberto Brizuela Rodríguez[a]

[a]*Centro de Investigación Científica y Educación Superior de Ensenada*
Carr. Ensenada-Tijuana 3918, Zona Playitas, Ensenada, B.C., México, 22860.
nwatkins@cicese.edu.mx, cbrizuela@cicese.edu.mx

2013 Published by $\mathcal{DIFU}_{100}ci@$ http://www2.uaz.edu.mx/web/www/publicaciones
Selection and peer-review under responsibility of the Organizing Committee of the CICOMP-2013, www.cicomp.org

Resumen

En soluciones recientes para el análisis de sentimientos automatizado se han utilizado diversas herramientas para extraer características de un texto que van desde la representación de cada palabra hasta características que combinan información semántica y léxica del texto. Sin embargo, es técnicamente imposible para un clasificador automatizado hacer diferencia entre aquellas características que otorgan información sobre la polaridad del texto y aquellas que no, para lo cual se han diseñado diferentes métodos para realizar selección de características utilizando información estadística. Puesto que la mayoría de estos métodos son el tipo voraz, en muchas ocasiones fallan en eliminar características ruidosas y descartan otras que pueden servir para la clasificación, por esto se han diseñado métodos alternativos que optimicen la tarea de selección, En este trabajo se propone la implementación de un algoritmo basado en Evolución Diferencial, experimentos preliminares muestran que los resultados son competitivos con otras propuestas del estado del arte.

Palabras clave: Evolución diferencial, análisis de sentimientos, selección de características.

1. Introducción

El análisis de sentimientos (también conocido como minería de opiniones) es un problema de clasificación de textos, donde dado un conjunto de clases se busca determinar a cual clase pertenece un texto dado. Sin embargo, a diferencia de otros problemas de clasificación de textos, el análisis de sentimientos intenta identificar la información subjetiva que indique la polaridad del documento, que en el caso de clasificación binaria puede ser positiva o negativa.

El análisis de sentimientos dista de ser un problema trivial, en un estudio realizado por [1] se reporta que expertos en clasificación llegan a un 80 % acuerdo entre ellos o precisión (es decir, el 80 % de los documentos fueron clasificados de la misma

manera, esto quiere decir que para un experto el otro clasificó con ese porcentaje de precisión), por lo tanto cualquier sistema automatizado que pueda lograr empatar o sobrepasar este número se considera como bueno.

Existen diversos métodos para la clasificación automatizada, [2] y [3] presentan una lista detallada de los utilizados hoy en día, entre estos se encuentran el uso de clustering, Entropía Máxima, redes bayesianas, SVM (Máquina de Vectores de Soporte), redes neuronales, árboles de decisión, regresión lineal. De estos SVM y redes bayesianas naif (NB) son de los que obtienen mejores resultados de precisión sin sacrificar el costo computacional.

Los clasificadores automatizados no pueden entender el texto directamente como lo hace una persona, por lo cual es necesario representar la información explícita e implícita en un formato entendible para la máquina. A esta tarea se le llama extracción de características, donde se obtiene cada elemento que será utilizado para la clasificación (palabras, patrones de frase, signos de puntuación, etcétera) y dependiendo del clasificador se traducen a un formato con el cual este pueda trabajar (ej. en el caso de SVM los textos se presentan como un vector de índices los cuales apuntan a las características extraídas). En [4] se presenta una lista de las diferentes características que pueden ser extraídas de un texto. En este trabajo nos enfocamos en aquellas que se pueden obtener con herramientas que no dependan de intervención humana como lo son los diccionarios morfo-sintácticos, sino de herramientas de extracción directa de características como lo son n-gramas, signos de puntuación, largo de oraciones y etiquetado morfológico.

La alta dimensionalidad del espacio de características, hace deseable la implementación de una selección previa de las mismas que no sacrifiquen la precisión de clasificación, por lo tanto, una vez obtenidas las características se realiza un proceso de selección previa a la clasificación para eliminar aquellas características de ruido que no solo distan de ser representativas de una clase sino que también afectan negativamente a la clasificación en tiempo y precisión. Las técnicas más comunes de selección de características son aquellas que están basadas en frecuencia de aparición de las características (ej. tf-idf), en información mutua (ej. ganancia de información), y en pruebas de estadística (ej. χ^2 o Chi Cuadrada). La primera se utiliza principalmente para pre-procesar el texto eliminando las características que por frecuencia de aparición son muy comunes en ambas clases como para determinar la pertenencia a una de ellas (ej. el uso de pronombres y artículos generalmente no es indicativo de que un texto sea positivo o negativo), la segunda utiliza información individual de cada característica y su valor de entropía en un árbol de decisión para determinar si es un buen divisor de clases, y la última aplica pruebas estándar de estadística de valor esperado y valor obtenido para determinar la dependencia que hay entre la presencia de una característica y la pertenencia de un texto a una clase. En [5] se establece una correlación entre estos tres métodos, ya que bajo ciertas condiciones arrojan resultados similares.

En la Sección 2 de este documento se presentan los antecedentes que inspiraron el método que aquí se presenta, en la Sección 3 se describe el método basado en Evolución Diferencial y en la Sección 3.3 se exponen los resultados. Finalmente, la Sección 4 presenta las conclusiones obtenidas a partir de los resultados observados y trabajo futuro para el desarrollo del algoritmo.

2. Trabajo relacionado

El análisis de sentimientos comienza a principios de los años 90 como una tarea manual de clasificación. Sin embargo, a partir del año 2000 se considera el nacimiento de la versión moderna del análisis de sentimientos con el surgimiento de los primeros trabajos que utilizan clasificadores y herramientas automatizadas para la clasificación, trabajos como el de [6], [7] y [8] entre otros vienen a formar la base para el trabajo presente y han inspirado en gran parte los estándares de comparación de resultados (ej. En [7] se define por primera vez el corpus de prueba utilizado por un gran número de trabajos recientes). En esta sección se presenta de manera superficial aquellos trabajos que tienen una relación directa con el Algoritmo de Selección basado en Evolución Diferencial (ASED).

2.1. Corpus de texto

Como ya se menciona al inicio de esta sección, en [3] y [7] se presenta un conjunto de documentos (corpus de texto) que consiste en 2000 críticas de cine en inglés extraídas de la página de internet IMDB y etiquetadas 1000 como positivo y 1000 como negativo (versión 2.0 del corpus). Este corpus ha sido utilizado en sus diferentes versiones en varios trabajos de análisis de sentimientos y facilita la comparación de resultados con

sistemas que no se encuentran disponibles pero que han sido probados con este corpus.

2.2. Selección de características basada en Algoritmos Genéticos

En [4] se presenta el algoritmo EWGA (Algoritmo Genético con Peso de Entropía por sus siglas en inglés). Aunque no es la primera vez que se utiliza el cómputo evolutivo en la clasificación de textos (ver [9]), este es de los primeros trabajos publicados acerca del uso de los mismos para la selección de características para clasificación supervisada.

El EWGA utiliza una representación binaria del individuo donde la posición de los valores en el vector indican el índice de la característica en cuestión y el valor 1 dentro del vector indica que la característica es utilizada y el 0 que esta se descarta. La población es construida al azar y uno de los individuos es generado con el método de ganancia de información con un umbral de 0.0025 y procede a realizar los siguientes pasos:

1. Obtener los pesos de las características utilizando la función de ganancia de información (IG)
2. Incluir las características elegidas de acuerdo a un umbral de IG para formar el primer individuo de la población y los n-1 individuos restantes generarlos de manera aleatoria.
3. Evaluar y seleccionar las soluciones basándose en la función de aptitud
4. Cruzar los pares de soluciones de tal forma que se maximice la diferencia total de IG entre las dos soluciones
5. Mutar las soluciones basándose en el peso IG de la característica para definir la probabilidad de mutación dónde el valor de IG es la probabilidad de que mute de 0 a 1 y 1-IG de que mute de 1 a 0
6. Repetir los pasos 3-5 hasta alcanzar el criterio de paro

La aptitud del individuo es la precisión de clasificación obtenida utilizando el método de 10 pliegues (10-fold) con un clasificador SVM con un kernel lineal, en [7] y en [2] se halla más información con respecto al papel que tienen los diferentes tipos de kernel en un SVM. Cabe notar que con este método se realizan $g \times n$ clasificaciones donde g es el número de generaciones y n el número de individuos en la población. Además, el paso 4 requiere del cálculo de la diferencia de IG cuyo costo será en el orden del largo del vector. Sin embargo, la selección de características se considera un problema fuera de línea, por lo que el costo computacional no es tan relevante siempre y cuando sea razonable a consideración de quien lo esté implementando. Otro detalle del EWGA es que en el paso 5 se van a descartar todas las características cuyo valor de IG es 0. Aunque no hay una justificación formal del porque se deban considerar las características que no tienen ganancia de información, tampoco lo hay para no hacerlo.

2.3. Comparación de métodos del estado del arte

En la Tabla 1 se muestran los resultados comparativos de distintos métodos de selección de características, donde Base se refiere a la clasificación sin selección de características, IG es utilizando el método de Ganancia de Información con un umbral de 0.0025, GA son los resultados del Algoritmo Genético de [4] sin utilizar Ganancia de Información, SVMW es utilizando el método de Máquina de Vectores de Soporte con pesos (véase [1]) y EWGA es el algoritmo descrito por [4] con una población de 50 individuos por 200 generaciones. Los resultados de [8], con grupos de apreciación, y de [7] se obtuvieron sobre una versión previa del corpus de 1300 críticas de cine. Los resultados de EWGA parecen ser muy favorables ya que no solo tiene buena precisión sino que también disminuye de manera efectiva el número de características totales sobre el cual se realiza la clasificación. Sin embargo, EWGA aún no ha sido probado con texto nuevo que no haya formado parte del entrenamiento, cabe notar que el uso de SVM de 10 pliegues ayuda a determinar el desempeño de un modelo ante texto que no formó parte del entrenamiento, pero por el diseño del algoritmo en el que la clasificación se realiza un número de veces y esta rige la aptitud del individuo, es imposible conocer si el algoritmo realmente está mejorando la precisión para un caso general o solo para los documentos que están siendo evaluados dentro del mismo. Es decir, es necesario probar la precisión del método con texto que no formó parte de las evaluaciones de aptitud.

Técnica	Precisión	No. Características
Base	87.95 %	26,870
IG	92.50 %	2,316
GA	92.55 %	2,017
SVMW	92.86 %	2,000
EWGA	95.55 %	1,752
Whitelaw [8]	90.20 %	–
Pang [4]	87.20 %	–

Tabla 1. Comparación de métodos

3. Selección de características basada en Evolución Diferencial

En [10] se describe la heurística de Evolución Diferencial, la cual es una versión modificada de Algoritmos Genéticos. La diferencia principal consiste en que la mutación se realiza antes del cruce, y utiliza una mutación definida por:

$$u_i(t) = x_{i_1}(t) + \beta(x_{i_2}(t) - x_{i_3}(t)) \quad (1)$$

Esta consiste en agregar al valor de un individuo tomado al azar ($x_{i_1}(t)$), la diferencia de los valores de otros dos individuos ($x_{i_2}(t) - x_{i_3}(t)$) multiplicado por un factor de escalamiento (β) para generar un vector mutado llamado vector de prueba ($u_i(t)$), que será posteriormente utilizado para cruzarlo y generar el individuo hijo. La variable t indica el índice dentro del vector del individuo.

Originalmente fue diseñado para problemas de optimización continuos, sin embargo, en [9] se presentan distintas técnicas para aplicar Evolución Diferencial a problemas de permutación, una de esas técnicas es el de modificar ligeramente la fórmula de mutación para obtener solo valores enteros y que sean válidos para la permutación al no repetir resultados dentro de un mismo vector. Puesto que en el algoritmo para selección de características, solo importa mantener los valores dentro de un intervalo válido que va de 1 al número total de características, y que los valores sean enteros. La repetición de índices se puede lidiar de manera externa eliminando las repeticiones de un índice previo a la clasificación.

3.1. Representación del individuo

Una de las diferencias entre este algoritmo y el propuesto por [1] es la manera en la que se representa el cromosoma del individuo. Una de las desventajas observadas en la representación binaria para este problema en específico es que mientras más características son descartadas, estas serán sustituidas por un 0 binario, lo cual producirá cromosomas con la mayoría de los genes apuntando a características que no se van a utilizar. Para tratar de solucionar este problema, se ha propuesto una representación de números enteros con longitud variable, cada vector o cromosoma incluirá los índices directos de cada característica (Figura 1). No es la primera vez que se utilizan vectores de longitud variable en algoritmos genéticos o evolución diferencial (ver [12] y [13]), además esto no solo produce vectores de menor tamaño sino que puesto a que la posición del gen no influye en el índice de la característica, permite realizar ordenamiento aleatorio (esto es cambiar el orden en que aparecen los índices dentro del vector) y otras operaciones que permiten prevenir la convergencia prematura del algoritmo. Existe la probabilidad de que dos o más individuos tengan los mismos índices pero en diferente orden, se espera compensar esto con la capacidad de exploración del cruce, donde dos individuos iguales puedan moverse a puntos distintos, además se incluyeron mecanismos que procuran mantener la diversidad en la población agregando características aleatorias a individuos similares.

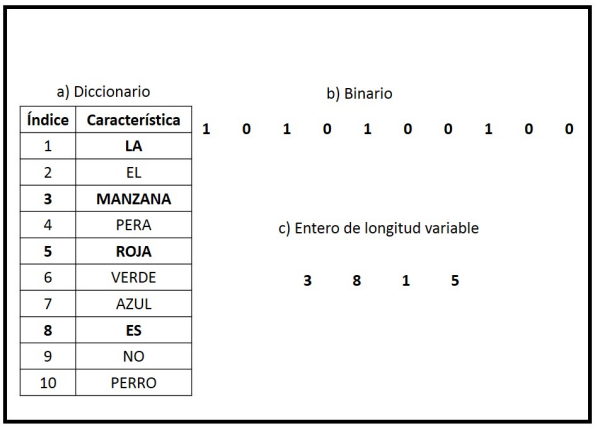

Figura 1. Representación del individuo

3.2. Algoritmo basado en Evolución Diferencial

El algoritmo ASED realiza los siguientes pasos:

1. Inicializar la población de manera aleatoria y añadir un individuo generado con ganancia de información.
2. Evaluar la aptitud de cada individuo
3. Realizar la mutación de los individuos
4. Realizar el cruzamiento entre los individuos y su respectivo vector de prueba
5. Evaluar la aptitud de los individuos hijo
6. Elegir entre el hijo y el padre aquél que tiene una aptitud más alta para la siguiente generación
7. Repetir hasta alcanzar el criterio de parada o número de generaciones

En el Algoritmo 1 se observa el pseudocódigo del método propuesto. En este algoritmo los parámetros de entrada son el número de individuos en la población (n), número de generaciones (Gen), probabilidad de cruzamiento (p_c), el valor del factor de escalamiento (β), y el umbral de Ganancia de Información ($delta$).

1 1 Algoritmo: DifferentialEvolution($n, Gen, p_c, \beta, \delta$)

Entrada: Número de individuos (n), número de generaciones (Gen), probabilidad de cruce (p_c), valor de escalamiento (β) y umbral de Ganancia de Información (δ)

Salida: Individuo con mejor aptitud de clasificación

```
t ← 0;
InitializePopulation(n, C(0));
f(C(0)) ← EvaluatePopulationFitness(C(0))
while t<Gen do
 for x_i ∈ C(t) do
  u_i ← Mutate(C(t), x_i, β);
  x´_i ← Crossover(u_i, x_i, p_c);
  f(x´_i) ← EvaluateFitness(x´_i);
  if f(x´_i)>f(x_i) then
   C(t+1) ← C(t+1) ⋃ {x´_i}
  else
   C(t+1) ← C(t+1) ⋃ {x_i}
  end if
 end for
end while
return BestFitness(f(C(t)))
```

En la línea 2, la población es inicializada tomando índices al azar del conjunto de características. Esto puede tener como consecuencia la generación de individuos de distintas longitudes. En la línea 3 se añade un individuo extra que es construido utilizando un filtro basado en ganancia de información con un umbral *delta*. El método EvaluatePopulationFitness() (línea 4) obtiene la aptitud de cada individuo de la generación 0 haciendo pruebas de clasificación. En las líneas 5 a 16 se realiza la mutación (línea 7) y cruzamiento (línea 8). Además, el método EvaluateFitness() (línea 9) obtiene la aptitud de un solo individuo, mientras que el método BestFitness() (línea 17) obtiene el individuo con la mejor aptitud obtenida en una generación y todo se repite hasta terminar el número determinado de generaciones.

3.3. Mutación

Para la mutación se utiliza el procedimiento descrito en esta misma sección. Para enfrentar el problema de individuos de diferentes longitudes, el factor de escalamiento (β) es multiplicado o dividido por 10 según sea el caso para lidiar con los índices que rebasan al individuo de menor longitud. En la Figura 2, se muestra un ejemplo gráfico de lo que sucede con el vector de prueba, donde **a** incluye los valores que fueron obtenidos sin modificación a la ecuación, **b** se obtuvieron multiplicando por diez el factor de escalamiento para compensar la falta del primer valor y **c** son los valores que fueron descartados como excedente.

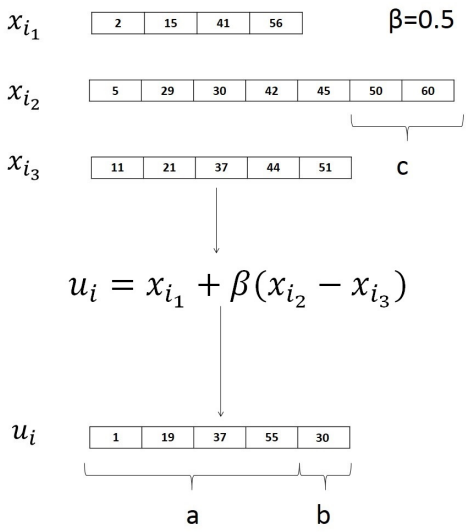

Figura 2. Mutación

3.4. Cruzamiento

El cruzamiento se realiza en base al valor de ganancia de información. En la Figura 3, se observa un ejemplo de cruzamiento con probabilidad de 0.7 de cruce. El elemento con mayor ganancia de información tiene una probabilidad pc de ser añadido al individuo contra 1-p_c del que tiene menor ganancia. Si ambos valores son iguales tienen una probabilidad de 0.5, y si hay diferencia de longitud entre el vector de prueba y el vector padre, la probabilidad de agregar los elementos sobrantes es de 0.5.

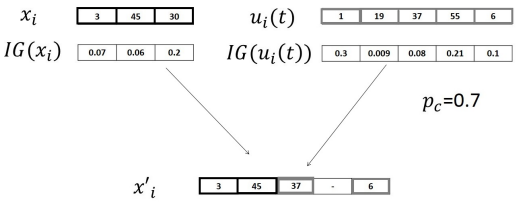

Figura 3. Cruzamiento

4. Evaluación

Los detalles para la experimentación son los siguientes:

- Lenguaje de programación: Java JDK (versión 7.21)
- Plataforma de desarrollo: Eclipse Juno (4.2.2)
- Sistema operativo: Windows 7 46 bits SP1
- Procesador: Intel Core i5 2.67GHz
- Memoria RAM: 6 GB
- Modelo: Asus U43F

4.1. Experimentos

Debido a que el algoritmo requiere de un gran número de clasificaciones para obtener el valor de aptitud, y el tiempo de clasificación aumenta exponencialmente con el tamaño del corpus se extrajeron distintos conjuntos del corpus de críticas de cine para realizar las pruebas. En todas las pruebas se utilizaron n-gramas hasta grado 3 con umbral de aparición de 5 para las características. Primero se realizó una prueba con 300 críticas de cine (150 positivas, 150 negativas) para encontrar el valor óptimo de β. Para mantener los parámetros lo más cercano posible a los empleados en [1], las pruebas se realizaron con un umbral de ganancia de información δ=0.0025, con una población de 20 individuos con un límite de 200 generaciones. En la primer prueba (Figura 4) se hicieron 20 corridas variando el valor de β de 0.1 a 2 y el valor de precisión más alto obtenido fue de 98 % cuando la precisión base fue de 67 % y ganancia de información obtenía una precisión de 95.6 %. La mejoría es consistente cuando 1<β<2. El número base de características es de 20726 entre unigramas, bigramas y trigramas. Ganancia de información obtiene 14000 y el algoritmo basado en evolución diferencial obtiene 426 características.

La tercera prueba se realizó con 500 críticas de cine (250 positivas, 250 negativas), precisión base de 82 %, el resultado obtenido fue de 99.8 % de precisión con 3566 (de 43,521 de base) características con β=1.5.

Método	Precisión
Base	73.33 %
IG	73.33 %
ASED	80 %

Tabla 2. Comparación preliminar

En la Tabla 2 se presentan los resultados de una cuarta prueba con el corpus de 300 documentos donde se extrajeron 30 de estos para que no formaran parte de la selección, al finalizar el algoritmo se clasificaron estos

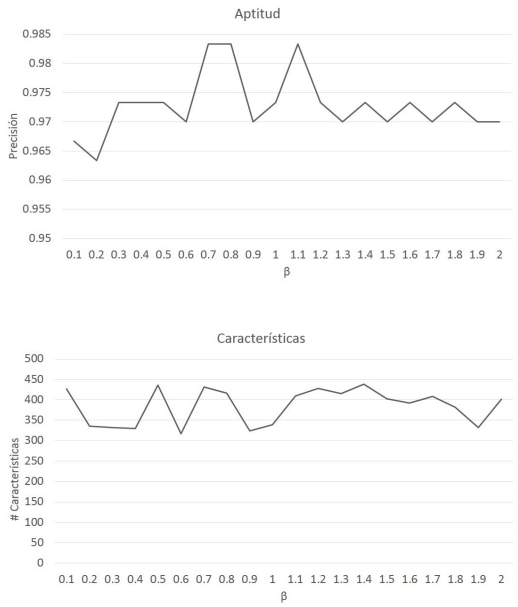

Figura 4. Resultados de primera prueba

30 documentos utilizando el modelo construido por los 270 restantes. Cuando no hay selección ninguna (Base) se obtiene una precisión de 73.33 % utilizando el modelo de 270 documentos para clasificar a 30 restantes. El filtro basado en ganancia de información obtiene una solución similar, mientras que la lista de características obtenida por el algoritmo logra el 80 % de precisión.

4.2. Comparaciones

Con el fin de comparar este método con otros, la quinta y última prueba se realizó con todo el corpus (2000 críticas de cine), 53.425 características de base con precisión de 81.3 % y al final se obtuvieron 2447 características con un 93.25 % de precisión. Aunque aún es necesario realizar más pruebas, si comparamos este resultado con el de la Tabla 1, el método aquí descrito no dista mucho del propuesto por [1] y utiliza una población de menor tamaño lo que se traduce en menor número de clasificaciones. Sin embargo, este resultado no es directamente comparable puesto que no se ha obtenido el mismo número de características base ni de porcentaje de precisión reportado en la Tabla 1.

5. Conclusiones y trabajo futuro

Los resultados obtenidos muestran que el sistema logra mejorar la precisión obtenida por el método es-

tadístico que en este caso es el basado en ganancia de información. Además, requiere de un menor número de iteraciones y de individuos en la población que el algoritmo EWGA propuesto por [1]. Aún es necesario realizar más pruebas con el mismo corpus de texto y el mismo número de características para poder concluir si el método basado en evolución diferencial es mejor que el basado en algoritmos genéticos. Puesto que los resultados dependen fuertemente de la precisión inicial de ganancia de información, hace falta probar este método integrando otros del tipo estadístico como lo es χ^2. La representación del individuo como un vector de números enteros de longitud variable combinado con las operaciones de mutación y cruzamiento y el uso de ordenamiento aleatorio permiten explorar índices que no están presentes en la población, aun cuando los individuos son muy similares, lo cual previene la convergencia prematura. Aún es necesario probar este método y otros del estado del arte con documentos desconocidos por el clasificador.

Referencias

[1] T. Wilson, J. Wiebe, & P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis." In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT ´05. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 347-354

[2] P. Manning, C. D. Raghavan, & H. SchÃijtze An Introduction to Information Retrieval 1st Edition. Cambridge University Press, New York, NY, 2008.

[3] A. Westerski, Sentiment Analysis: Introduction and the State of the Art overview. Universidad Politecnica de Madrid, España, 2007, pp 211-218

[4] A. Abbasi, H. Chen, & A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums." In ACM Transactions on Information Systems, 2008, pp. 1-34.

[5] A. Ginsca, E. Boros, A. Iftene, D. Trandabat, M. Toader,M. Corici,D. Cristea, "Sentimatrix: multilingual sentiment analysis service." In Proceedings of the second Workshop ACL - WASSA, 2011, pp. 189-195.

[6] B. Pang, & L. Lee, "Opinion Mining and Sentiment Analysis". Foundations and Trends in Information Retrieval, 2006, pp 91-231.

[7] A. Go, R. Bhayani, & L. Huang, "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford. Stanford, California, 2009

[8] C. Whitelaw, N. Garg, & S. Argamon, "Using Appraisal Groups for Sentiment Analysis." In Proceedings of the 14th ACM Conference on Information and Knowledge Management, 2005, pp. 625-631.

[9] A. Abraham, S. Das, & A. Konar, "Document Clustering Using Differential Evolution." 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 1784-1791.

[10] R. Storn, & K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." Journal of global Optimization, 1997, pp 341-359.

[11] G. C. Onwubolu, & D. Davendra, Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009

[12] B. Pang, & L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." Proceedings of the Association for Computational Linguistics, 2004, pp 271-278.

[13] Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In International Conference on Machine Learning. Retrieved from http://faculty.cs.byu.edu/~ringger/Winter2007-CS601R-2/papers/yang97comparative.pdf