Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Proyecto tecnológico

Sistema para la clasificación de artículos científicos mediante el algoritmo K-means utilizando características semánticas

Juan Antonio Lopez Ornelas
209305611
al209305611@alumnos.azc.uam.mx

Trimestre 2015 Invierno
22 de Abril de 2015

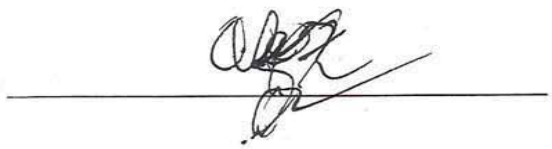
Dra. Maricela Claudia Bravo Contreras
Profesor Asociado "D"
Departamento de Sistemas
mcbc@correo.azc.uam.mx

Dr. José Alejandro Reyes Ortiz
Profesor Titular “A”
Departamento de sistemas
jaro@correo.azc.uam.mx

Yo, Maricela Claudia Bravo Contreras, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, Juan Antonio Lopez Ornelas, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Contenido

I. Resumen 1

II. Objetivos 2

III. Antecedentes 3

IV. Introducción 3

V. Justificación 4

VI. Marco Teórico

6.1. Clustering(agrupamiento) 5

6.2. Algoritmo K-Means 7

6.2.1. Métricas 9

6.3.Tf-Idf 10

6.4.Algoritmo Porterstemming 11

VII. Desarrollo

7.1. Extracción y transformación de características 12

7.2. Algoritmo de clasificación 13

7.3. Representación de la clasificación 14

VIII. Pruebas 18

IX. Desarrollo

9.1. Extracción y transformación de características 12

9.2. Algoritmo de clasificación 13

9.3. Representación de la clasificación 14

X. Manual de uso e instalación

10.1. Instalación de la aplicación 15

10.2. Uso de la aplicación 16

XI. Pruebas 19

XII. Resultados 21

XIII. Conclusión 23

XIV. Referencias bibliográficas 24

Índice de figuras

Figura 1 Cluster con cohesión natural. 6
Figura 2 Cluster con solución externa 6
Figura 2 Despliegue en app manager 15
Figura 3 Inicio de la aplicación 16
Figura 4 Selección de K 17
Figura 5 Pantalla de espera 17
Figura 6 Presentación de resultados 18

Índice de figuras

Tabla 1 n=1 k =3. 21
Tabla 1 n=3 k =3. 21
Tabla 1 n=20 k =3. 21
Tabla 1 n=60 k =3 22
Tabla 1 n=60 k =2 22
Tabla 1 n=60k =2 22
Tabla 1 n=60 k =2 22

Resumen

Se diseñó un sistema para la clasificación de artículos científicos en idioma inglés utilizando el algoritmo de agrupamientoK-means. El sistema implementa la obtención, de un conjunto de documentos en formato PDF, y presentación de resultados por medio de la tecnologíaweb Java Servlets.

La aplicación está desarrollada en el lenguaje de programación Java y para su despliegue se utilizó el servidor web Apache Tomcat. En la extracción de texto de los documentos se utilizóla librería ApachePDFBox. En la homogenización y tratamiento del texto fue utilizado el algoritmo Porterstemming.

Laaplicación desarrollada en este proyecto genera la clasificación de cada documento por su tópico principal muestra los resultados en una página web. También genera en un archivo XMLdescargable, donde se indica a que clase pertenece cada documento ingresado.

El resultado final de este proyecto es una aplicación web java con formato de archivo war, que fue desarrollada usando el entorno de desarrollo integrado Netbeans 8.0, debe ser desplegada en un servidor Apache Tomcat 7.0.

Objetivos

- Diseñar e implementar un sistema Web para la clasificación de documentos científicos en inglésmediante el algoritmo K-means basado en características sintácticas, semánticas y contextuales.

- Diseñar un módulo para la extracción y trasformación de características sintácticas, semánticas y contextuales de documentos científicos utilizando el modelo de bolsa de palabras.

- Implementar el algoritmo k-means para la clasificación de documentos científicos basada en las características extraídas con un enfoque semántico.

- Diseñar e implementar un sistema web que integre el proceso completo de clasificación de un conjunto de documentos y muestre los resultados de manera textual.

- Implementar un módulo para enriquecer la base de datos semántica con el tópico descubierto en la clasificación.

Antecedentes

Existen una gran cantidad de plataformas en las cuales se puede tener acceso a artículos de investigación pero cada una de ellas utiliza sus propios buscadores y clasificadores y esto dificulta la búsqueda de artículos relacionados al tópico de interés, ya que para encontrarlos se necesita buscar en cada una de estas plataformas por separado y después revisar cada uno de los documentos obtenidos.

Aunque existen motores de búsqueda de artículos científicos existe la limitante de que estos solo buscan en bases de datos de artículos científicos libres condicionando así el número de artículos a los que se puede tener acceso.

Introducción

La aparición de nuevas tecnologías, como las redes académicas o de investigación ha marcado un constante crecimiento en la cantidad de artículos científicos publicados. Gracias a esto existe la posibilidad de acceder a publicaciones desde cualquier lugar y en cualquier momento.

El extraer y analizar información de un artículo permite su clasificación de acuerdo al principal tópico del que este trata. La tarea de clasificar documentos en base a la temática que abordan, ayuda a que encontrar artículos relacionados al tema de interés sea más sencillo.

Existen distintos algoritmos que resuelven el problema de *clustering*o agrupamiento, en el cual los objetos contenidos en el mismo *cluster*(grupo) comparten características similares definidas por el mismo criterio.

Uno de los algoritmos más sencillos es el algoritmo *k-means*en el cual se especifica el número de grupos a identificar y los objetos están representados como un conjunto de características numéricas.

Justificación

Buscar información relevante, puede llegar a complicarse si se hace de forma manual ya que se debe navegar entre una gran cantidad de artículos y revisar cada uno de ellos.

La migración de publicaciones de papel hacia las publicaciones electrónicas es cada vez mayor, ya que esto hace que compartir la información sea más sencillo.

Aunque las bases de datos académicas clasifican la información para facilitar las búsquedas, solo permiten hacerlo con los artículos que las conforman, al obtener publicaciones de varios sitios se regresa al problema de revisar cada una de ellas.

Diseñar un sistema de información que clasifique artículos científicos facilitara la consulta y reducirá la tarea de buscar información relacionada al tema de interés.

Marco teórico

Clustering(agrupamiento)

Los métodos numéricos para clasificar fueron originados en su gran mayoría en las ciencias naturales como la zoología y biología para librar a la taxonomía de su naturaleza subjetiva. Con la finalidadde proveer un sistema de clasificación estable y objetivo. Objetivo en el sentido de que el análisis del mismo conjunto de organismos por la misma secuencia de operaciones produce la misma clasificación y estable en que la clasificación permanece igual al agregar una amplia cantidad de organismos o de características que los describan.

Estos métodos numéricos reciben un nombre diferente dependiendo del área de estudio donde se utilicen pero en general se conocen con el nombre de clustering o agrupamiento a los procedimientos que buscan descubrir grupos en un conjunto de datos.

En muchas de las aplicaciones del agrupamiento un objeto o individuo pertenece a un solo cluster (grupo). En algunas ocasiones el traslape y la inexistencia justificada de grupos es válida y puede ser la solución más acertada.

El conjunto de datos básico para la mayoría de las aplicaciones de clustering es la matriz de datos multivariados X que es,

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x1_p \\ x_{21} & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & \dots & x_{np} \end{pmatrix}$$

La entrada X_{ij}en X indica el valor de la j variable en el objeto i. Las variables en X a menudo pueden ofrecer una mezcla de valores ordinales, categóricos o inexistentes como entrada. Los valores mixtos o inexistentes pueden complicar el agrupamiento. Los datos puedes ser estructurados o no estructurados y estos deben ser tratados de distinta forma.

Un cluster puede ser definido como un grupo homogéneo y aislado de elementos que comparten características similares y en algunos casos este se forman de manera natural y en otros no es posible no existe un patrón visible en los elementos.

Figura 1 Cluster con cohesión natural.

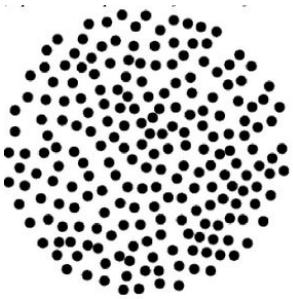


Figura 2 Cluster con solución externa

K-means

El objetivo del algoritmo K-means es dividir M puntos en N dimensiones en K clustersde manera que la suma de cuadrados dentro del clusterse minimize.

El procedimiento sigue una manera simple y fácil de clasificar un conjunto de datos a través de un cierto número de clusters.

La idea principal es definir K centroides, uno para cada cluster. La posición inicial de los clusters iniciales se puede determinar de distintas formas ya que diferentes posiciones iniciales causan diferente resultados.
Existen dos métodos comúnmente utilizados para determinar los K centroides iniciales

a) Seleccionarlos aleatoriamente.
   Para que este método proporcione resultados apropiados se necesita iterar varias veces el algoritmo y hacer un análisis de los resultados de cada iteracion.

b) Obtenerlos de un pre procesado de los datos
   La precisión de resultados de esta iteración depende de la calidad del pre procesado.

El siguiente paso es tomar cada punto que pertenece a un conjunto de datos dado y asociarlo al centroide más cercano.

Cuando no quede ningún punto pendiente, se completó el primer paso y se ha hecho una primera agrupación. En este punto tenemos que volver a calcular k nuevos centroides como baricentros de las clustersobtenidos anteriormente.

Después de que tenemos estos nuevos K centroides se realiza una nueva asignación entre el mismo conjunto de datos y el nuevo centroide más cercano generándose así un ciclo.

Cuando los K centroides convergen, dejan de moverse, se ha obtenido una solución

Este algoritmo pretende minimizar la función objetivo:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} D(X_i, C_j)$$

Donde $D(X_i , C_j)$ es la distancia entre el punto X y el centroide del clusterC_j.

El algoritmo está compuesto por los siguientes pasos:

1. ColocarK puntos en el espacio representado por los objetos que están siendo agrupados. Estos puntos representan centroides de los grupos iniciales.
2. Asignar a cada objeto al grupo que tenga el centroidemás cercano.
3. Cuando se hayan asignado todos los objetos, re calcular las posiciones de los K centroides.
4. Repetir los pasos 2 y 3 hasta los centroides ya no se muevan. Esto produce una agrupacón de los objetos en grupos donde la métrica a minimizar puede ser calculada.

Métricas

Distancia euclidiana

La distancia euclidiana es definida como la distancia entre dos puntos:

$$D = \sqrt{\sum_{k=1}^{m} \left(X_{ik} - X_{jk}\right)^2}$$

Distancia Manhattan

La distancia Manhattan distancia describe un movimiento bastante restrictivo en bloques rectangulares, como en la ciudad de Manhattan. Calcula las diferencias absolutas entre coordenadas de dos puntos:

$$D = \sum_{k=1}^{m} \left|X_{ik} - X_{jk}\right|$$

Distancia Chebyshev

Se define como la mayor diferencia absoluta individual de cualquier par de coordenadas entre de dos puntos:

$$D = max_k \left|X_{ik} - X_{jk}\right|$$

Similaridad del Coseno

Tomando el coseno del ángulo entre dos vectores, se obtiene un valor entre 0 y 1 que es indicativo de la similitud entre estos. Cuanto menor sea el ángulo, mayor es el valor del coseno, y por lo tanto mayor es la similitud.

$$cosine\ (v_1, v_2) = \frac{(v_1 \bullet v_2)}{\|v_1\| \|v_{12}\|}$$

Donde • indica el producto punto entre 2 vectores.

Tf-Idf

Tf-idfsignificafrecuencia de término – frecuencia inversa de documento, el peso-tfidfes una medidaestadística que se utilizapara evaluar que tan importante esa una palabra para un documento en una colección. La importanciaaumentaproporcionalmente al númerode veces que unapalabra aparece enel documento, perose compensapor la frecuencia dela palabra en elcorpus.Para calcular el peso Tf-Idf es necesario calcular la frecuencia de término normalizada y la frecuencia inversa.

$$tf = tf * idf$$

Tf

Es una métrica que mide que tanfrecuente aparece un término en un documento. La frecuencia de los términos se divide por el número total de términos en el documento, como una forma de normalización.

$$tf = \frac{Número\ de\ apariciones}{Cantidad\ de\ términos}$$

Idf

Es una métrica que cuantifica que tan común es en un término en la colección de documentos.

$$tf = \log_e \frac{Número\ total\ de\ documentos}{Numero\ de\ documentos\ que\ contienen\ el\ témino}$$

Algoritmo Porterstemming

El algoritmo Porterstemming es proceso para remover el común morfológico y las terminaciones inflexivas de palabras en inglés conservando solo su lema. Aplicando este algoritmo se disminuye la cantidad de términos y por tanto la complejidad de la información con la que trabaja es reducida. Reducir las palabras a su lema o raíz es útil en áreas de la computación como la recuperación de la información. Existen diversas implementaciones de este algoritmo y adaptaciones a distintos idiomas.

Desarrollo

Para el desarrollo de este proyecto se utilizó el lenguaje de programación java y su tecnología java servlets. La aplicación web desarrollada es desplegada en el servidor web Apache Tomcat. A continuación se explicaran detalles de cómose desarrollaron cada una de las partes de este proyecto.

Extracción y transformación de características

En la extracción de características se utilizaron artículos científicos en idioma ingles en formato PDF. Se utilizó la librería Apache PDFBox para la extracción de texto a una cadena de texto plano. Posteriormente se removieron las palabras vacías de esta cadena y se redujo cada palabrano vacía a su raíz.

e.g.

Teniendo el archivo "*IntentionIsChoicewith Commitment.pdf"* se muestra un fragmento del texto extraído por medio del método *Pdf2Text* de la clase GetText.

This paper explores principles governing the rational balance among an agent's beliefs, goals, actions, and intentions. Such principles provide specifications for artificial agents, and approximate a theory of human action (as philosophers use the term). By making explicit the conditions under which an agent can drop his goals, i.e., by specifying how the agent is ommitted to his goals, the formalism captures a number of important properties of intention. Specifically, the formalism provides analyses for Bratman's three characteristic functional roles played by intentions [7, 9], and shows how agents can avoid intending all the foreseen side-effects of what they actually intend. Finally, the analysis shows how intentions can be adopted relative to a background of relevant beliefs and other intentions or goals. By relativizing one agent's intentions in terms of beliefs about another agent's intentions (or beliefs'), we derive a preliminary account of interpersonal commitments.

Al aplicarle el método *RemoveStopWords(Stringtext)*, el cual elimina las palabras vacías números y signos de puntuación, de la clase *StopWords*a la cadena obtenida anteriormente se obtiene la cadena:

paper explores principles governing rational balance agent beliefs goals actions intentions principles provide specifications artificial agents approximate theory human action philosophers term making explicit conditions agent drop goals agent ommitted goals formalism captures number important properties intention specifically formalism analyses bratman characteristic functional roles played intentions shows agents avoid intending foreseen side effects intend finally analysis shows intentions adopted relative background relevant beliefs intentions goals relativizing agent intentions terms beliefs agent intentions beliefs derive preliminary account interpersonal commitments

Posteriormente se reducen cada una de las palabras a su raíz o lema con el método *Stemtext(Stringtext)* de la clase Stemm con lo cual se obtiene la cadena:

paper explorprincipl govern ration balanc agent belief goal action intent principlprovidspecifartifici agent approximtheori human action philosoph term make explicit condit agent drop goal agent ommit goal formal captur number import properti intent specif formal analysbratmancharacterist function role plai intent show agent avoid intend foreseen side effect intend final analysi show intent adopt rel background relev belief intent goal relativ agent intent term belief agent intent belief derive preliminari account interperson commit

Algoritmo de clasificación

Después de obtener una cadena "limpia" de cada archivo se procede a contar las palabras que la componen por medio del método *GetPaperWordsSet*de la clase *WordsSet* y con esto se obtiene unTreeMap<String, Integer> con la lista de cada palabra y su frecuencia de aparición en el documento.

A continuación se al mapa obtenido anteriormente se le calcula el peso de cada palabra con el método *Tfidifweight* de la clase *Tfidifweigh, del* cual se obtiene un mapa con cada uno de los atributos que son utilizados para clasificar.

Una vez obtenido este mapa de atributos se procede a la clasificación utilizando el método *clustering*de la clase *Kmeans*.

Para el método *clustering*el mapa de palabras se transforma a unArrayList<double[]> que será el espacio vectorial con el que el algoritmo trabajara.

El algoritmo se implementa de la siguiente forma.

Inicialmente se seleccionan k "vectores" que serán los centroides iníciales.

Posteriormente este método itera entre 2 pasos que son realizados hasta que los centroides C no cambian:

1. Se calcula la similitud del coseno de cada vector V con cada centroide C, se asigna cada vector V al centroide C en donde exista la mayor similitud.

2. Los nuevos centroides se obtienen calculando media de los vectores que están asignados a estos.

Para obtener el óptimo se itera la implementación del algoritmo k*(número de documentos) y de los resultados obtenidos se selecciona el que tenga la mayor suma de similitudes de cada vector V y centroide C.

El resultado de este método es un mapa que contiene un entero indicando el cluster y una lista de cadenas con el nombre de cada archivo que pertenece a este cluster.

Representación de la clasificación

Se utiliza el mapa obtenido anteriormente y se le da formato con la finalidad de mostrar una tabla de cada clase y que documentos pertenecen a ella. Además se almacena un archivo XML, con la clasificación, el cual se puede descargar de cada archivo y la clase a ala a que este pertenece.

Manual de uso e instalación de la aplicación

Instalación de la aplicación

A continuación se describe como desplegar la aplicación *ClasificadorKmeans.war* en un servidor Apache Tomcat 7 desde el web manager app.

Se ingresa al manager app por mediohttp://<hostname><ip>:<puerto>/manager

Una vez en el Gestor de Aplicaciones Web de Tomcat se desplaza hasta

Desplegar >> Archivo WAR a desplegar

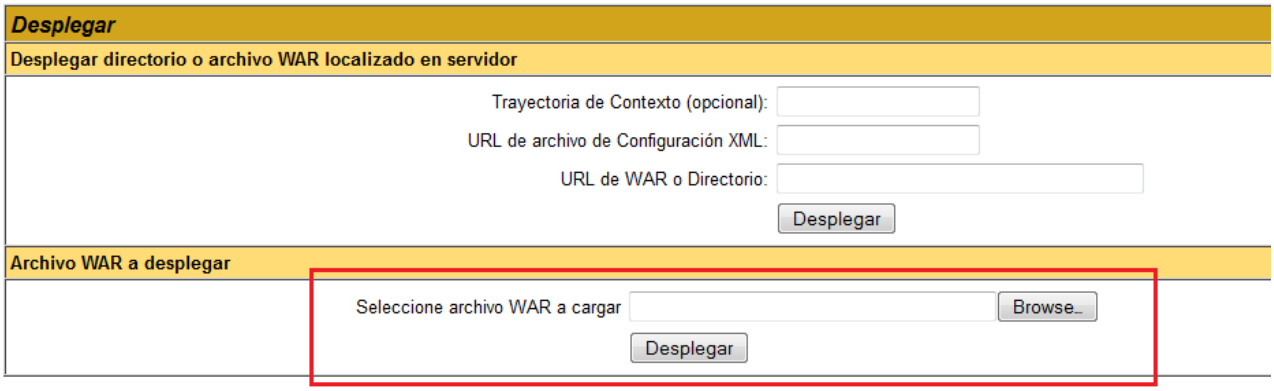


Figura 2 Despliegue en app manager.

Se selecciona el archivo*ClasificadorKmeans.war y se presiona desplegar.*

Uso de la aplicación

En la primera pantalla mostrada se seleccionan todos los archivos .pdf que serán clasificados presionando el botón Browse.., estos deben encontrarse en la misma carpeta.

Posteriormente se presiona el botón cargar.

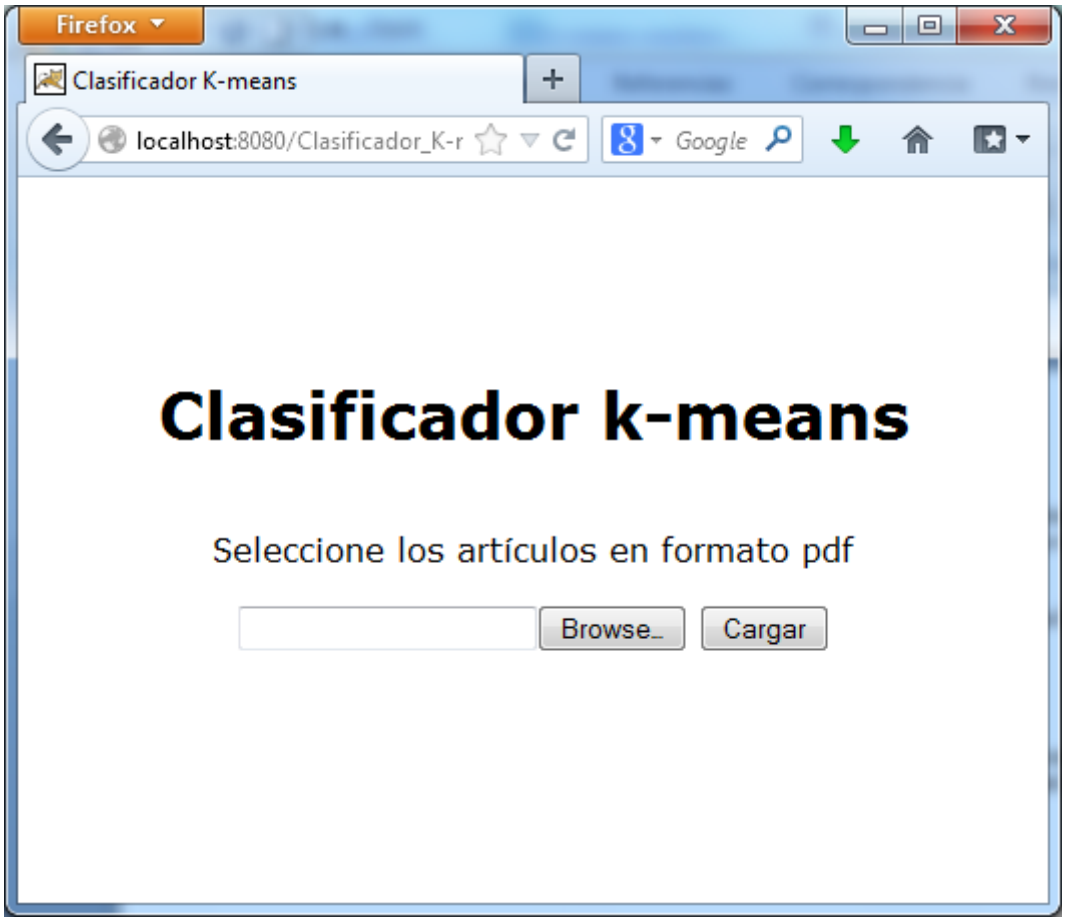


Figura 3 Inicio de la aplicación

En la pantalla 2 se ingresa un valor de k menor al número de artículos cargados. Se pulsa el botón clasificar y la aplicación empieza con la clasificación.

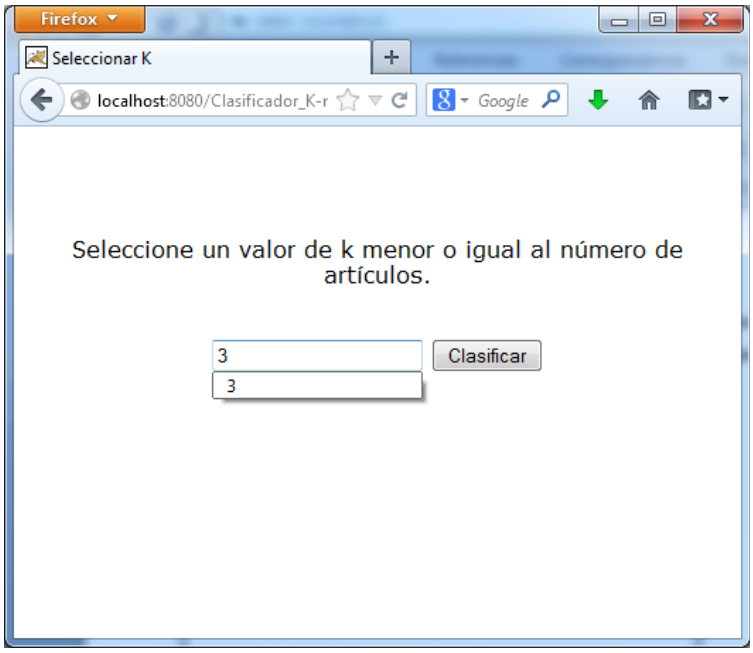


Figura 4 Selección de K

A continuación se muestra la pantalla de espera.

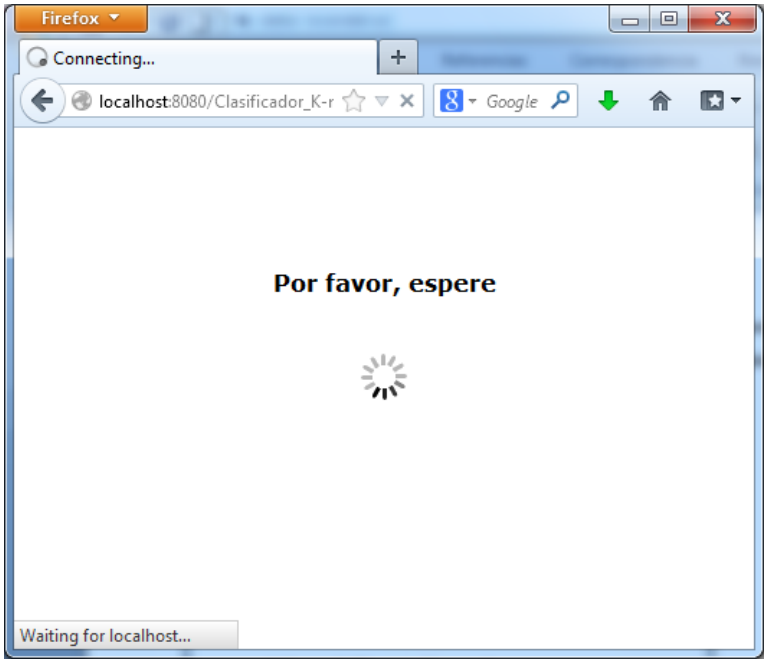


Figura 5 Pantalla de espera

Al terminar se muestran los resultados en forma de tabla y un enlace a un archivo XML con la clasificación obtenida. Para descargar el archivo se presiona el botón derecho > guardar enlace como…

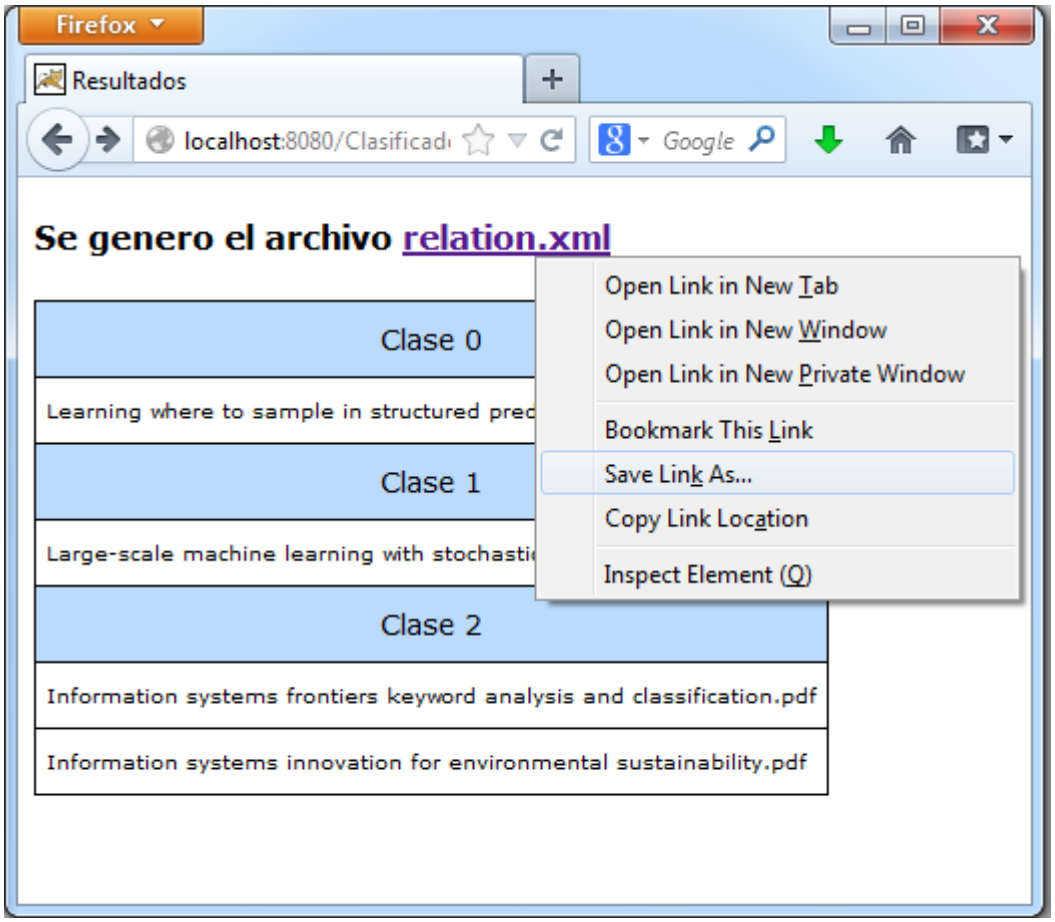


Figura 6 Presentación de resultados

Pruebas

Para evaluar el funcionamiento del sistema se utilizaron 20 documentos los cuales pertenecían a 3 categorías previamente conocidas.

1. Informationsystems

    a) Information systems innovation for environmental sustainability
    b) Information systems frontiers keyword analysis and classification
    c) A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research
    d) Virtualization security, strategy and management
    e) User adaptation and infusion of information systems
    f) A software development capstone course and project for CIS majors
    g) The role of continuous trust in information systems continuance

2. Artificial intelligence

    a) A computational foundation for the study of cognition
    b) Modeling the lifespan of discourse entities with application to coreference resolution
    c) Ai methods in algorithmic composition a comprehensive survey
    d) Learning where to sample in structured prediction
    e) Experimental demonstration of associative memory with memristive neural networks
    f) Large-scale machine learning with stochastic gradient descent
    g) Building high-level features Using Large Scale unsupervised Learning

3. Computer Networks

    a) Scheduling in networks with time-varying channels and reconfiguration delay
    b) Dynamic server allocation over time varying channels with switchover delay
    c) Cooperative routing in static wireless networks
    d) Robustness of interdependent networks the case of communication networks and the power grid
    e) Adaptive backpressure efficient buffer management for on-chip networks
    f) Ethernet distributed packet switching for local computer networks

Para probar la precisión se cambió la cantidad de iteraciones del algoritmo k-means que se realizan y cuyo valor final es era k*(número de documentos).

También se probaron distintos valores de k y combinaciones de las 3 clases.

Resultados

Aquí se encuentra una muestra de los resultados más significativos de las pruebas realizadas.

Se muestra la precisión la cual es calculada de la siguiente forma.

$$\boldsymbol{precisión = \frac{correctos}{total} * 100}$$

Con el número de iteraciones n=1 y k=3

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precision	50%	75%	55%	65%	50%	60%	60%	85%	60%	60%

Tabla 1 n=1 k =3

Con el número de iteraciones n=3 k=3

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precision	60%	75%	80%	70%	75%	60%	70%	70%	60%	75%

Tabla 2 n=1 k =3

Con el número de iteraciones n=20 y k=3

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precision	90%	100%	95%	80%	100%	95%	90%	95%	1000%	95%

Tabla 3 n=1 k =3

Con el número de iteraciones n=60 k=3

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precision	95%	100%	90%	95%	95%	95%	100%	95%	100%	95%

Tabla 4 n=1 k =3

Tomando los combinaciones de clases e iteraciones n=60

Clase 1 y Clase 3 con n=60

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precisión	100 %	100%	100%	100%	100%	100%	100%	100%	100%	100%

Tabla 5 Clase 1 y 2

Clase 2y Clase 3 con n=60

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precisión	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Tabla 5 Clase 2 y 3

Clase 1 y Clase 2 con n=60

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Precisión	92.85 %	100%	92.85 %	100%	92.85 %	100%	100%	92.85 %	100%	92.85 %

Tabla 5 Clase 1 y 2

Como se puede ver entre mayor sea el número de iteraciones mejor es la precisión. También se nota que si los tópicos de los documentos son de la misma área de conocimiento, eg Computación, la clasificación de estos pude ser afectada.

Conclusión

Utilizar clustering para clasificación de documentos es una forma útil para clasificar pero cuenta con la desventaja de que pude ser lento y solo proporciona información acerca de cuáles documentos son similares entre sí pero no proporciona información sobre el tópico principal de cada clase.

Otra desventaja de este método es que para encontrar una mejor categorización de inicio existen pocas opciones, como es la búsqueda de similitud entre todos los documentos o hacerlo aleatoriamente, la primera es lento ya que al aumentar la cantidad de documentos analizados aumenta el tiempo para encontrar una mejor categorización inicial y el segundo es lento porque se debe aplicar varias veces para que arroje resultados confiables.

Al utilizar las características semánticas de un documento pueden existir errores si los tópicos utilizan un léxico similar como es el caso de redes neuronales y redes de computadoras. También si el artículo científico es acerca del estudio de alguna lengua el eliminar las palabras vacías podría afectar los resultados.

En general se cumplieron los objetivos y este trabajo puede mejorarse agregando aprendizaje supervisado, etiquetado al algoritmo de clasificación y así mejorando la clasificación de inicio.

Referenciasbibliográficas

Steinbach, M., Karypis, G., & Kumar, V. (2000, August).A comparison of document clustering techniques.In *KDD workshop on text mining* (Vol. 400, No. 1, pp. 525-526).

Everitt, B. (2011). *Cluster analysis* (5th ed.). Chichester, West Sussex, U.K.: Wiley.

Kahkashan K, Sunit(2013 November), A comparative study of K Means Algorithm by Different Distance Measures, International Journal of Innovative Research in Computer and Communication Engineering(Vol 1, Issue 9).

Zong, J. (2014, November 17). K Means Clustering with Tf-idf Weights. Recuperado Marzo 16, 2015, fromhttp://jonathanzong.com/blog/2013/02/02/k-means-clustering-with-tfidf-weights

Apache PDFBox (2014, Febrero) Apache PDFBox - A Java PDF Library. [En línea] Disponible en: https://pdfbox.apache.org/index.html

Apache Tomcat (2014, Marzo) Apache TomcatDocumentation - . [En línea] Disponible en: http://tomcat.apache.org/tomcat-7.0-doc/index.html