Effects on Time and Quality of Short Text Clustering during Real-Time Presentations

Diego Fuentealba, Mario López and Héctor Ponce

Abstract— **Technologies for live presentations should consider users' capabilities to manage large amounts of data in real-time, particularly, exchanges of short texts (e.g., phrases). This study examines the effects on time and quality of text clustering algorithms applied to short, medium, and long size texts, and examines whether short text clustering shows a reasonable performance for live presentations. We run several simulations in which we varied the number of phrases (from 5 to 200) contained in each text type (long, medium, and short) and the number of generated clusters (from 2 to 10). The algorithms used were snowball steamers, TF-IDF, and K-means for clustering; and the text types were Reuters, 20 NewsGroup and an experimental data set, for the long, medium, and short size texts, respectively. The first result showed that text size had a large effect on the algorithm's execution time, with the shortest average time for the short texts and longer average time for the longest texts. The second result showed that the number of phrases in each text type significantly predicts execution time but not the number of clusters generated by K-means. Inertia and purity measures were used to test the quality of the clusters generated. Text size, number of phrases and number of clusters predict inertia; showing the lowest inertia for the short texts. Purity measures were like previously reported results for all text types. Thus, clustering algorithms for short texts can confidently be used in real-time presentations.**

Index Terms— **Text Mining, TF-IDF, K-Means, Short Phrases, Short Text, Sentences, Interactivity, Clustering.**

I. Introduction

Piense en una conferencia con un gran número de participantes donde el presentador invita a la audiencia a hacer preguntas u ofrecer comentarios. Si muchas personas tienen preguntas y hay limitaciones de tiempo, el presentador puede seleccionar a las personas que levanten la mano para hacer sus preguntas abiertamente. Otro método consiste en pedir a los participantes que escriban sus preguntas en una hoja de papel y las entreguen a un asistente que procederá a organizar las preguntas o comentarios y a seleccionar algunas de ellas para que sean leídas y respondidas por el presentador.

Paper Submitted 14/07/2020.

The authors are grateful for the financial support of the Project FONDEF/CONICYT IT18I0011 and FONDECYT 1200803.

D. Fuentealba is researcher in VirtuaLab in the Department of Industrial Engineering, Universidad de Santiago de Chile, Santiago, Chile. E-mail: diego.fuentealbac@usach.cl. https://orcid.org/0000-0001-5284-0448.

M. Lopez is professor in the Department of Industrial Engineering, Universidad de Santiago de Chile, Santiago, Chile. E-mail: mario.lopez@usach.cl. https://orcid.org/0000-0003-0909-7702

H. Ponce is professor in the Department of Accounting and Auditing, Faculty of Administration and Economics Universidad de Santiago de Chile, Santiago, Chile. E-mail: hector.ponce@usach.cl. https://orcid.org/0000-0002-7984-3945

En ambas circunstancias, muchas preguntas no serán respondidas, el proceso de selección suele ser aleatorio, y los participantes pueden sentirse decepcionados al observar que sus preguntas o comentarios no fueron considerados. ¿Es posible optimizar estos métodos de participación durante las presentaciones en vivo? Una posible solución es pedir a los miembros del público que utilicen sus dispositivos móviles para enviar sus preguntas o comentarios a un servidor, y utilizar técnicas de minería de textos para examinar y agrupar los textos que comparten similitudes, seleccionando una pregunta o comentario representativo.

Las técnicas de minería de textos se han utilizado para facilitar el proceso de descubrimiento de conocimientos a partir de una gran cantidad de textos identificando sesgos en los medios de comunicación [1], problemas a partir de las opiniones de los clientes [2], estado de patentes [3], autoría de texto [4], y para analizar las tendencias en los medios de comunicación sociales [5]. Sin embargo, estos métodos suelen presentar problemas cuando el conjunto de datos es más pequeño [6]. Además, tampoco se reportan los tiempos de ejecución de los algoritmos, existiendo un potencial desafío al utilizarlos en presentaciones en vivo donde se requiere un tiempo de procesamiento rápido y selección de textos más representativos tal como las preguntas más repetidas.

Existen herramientas que se han utilizado para resumir el contenido de eventos interactivos, como la API de Twitter [7] o aplicaciones para votaciones en vivo (ej.: PollEverywhere) que facilitan la participación en aulas y conferencias [8]–[10]. Sin embargo, estas herramientas no han propuesto formas de agrupación de frases y tampoco se integran a la interactividad en presentaciones en vivo [11]. En el presente trabajo se abordarán los desafíos relacionados al procesamiento de texto corto mediante las siguientes preguntas: ¿Pueden estas técnicas ofrecer resultados rápidos que se puedan mostrar a la audiencia en tiempo real durante las presentaciones en vivo? ¿Pueden las técnicas de minería de textos proporcionar un texto corto representativo de la población agrupada?

Este trabajo está organizado en diferentes secciones: en la segunda sección se muestra el planteamiento del problema. En la tercera sección se propone un algoritmo para agrupar textos cortos con la elección de la frase representativa. En la cuarta sección se valida el algoritmo en relación con textos cortos, medianos y largos para observar si cumple con los requisitos como la limitación de tiempo y la fiabilidad. Finalmente, se concluye el trabajo.

II. Planteamiento del Problema

Una frase no tiene un criterio estándar en cuanto a la longitud de palabras. Según [12], las personas pueden leer sin problemas un texto cuando su longitud no supera las 15 a 17 palabras, afectando su comprensión cuando la frase es más larga. Por esta razón, se ha definido como texto corto a un texto cuyo largo de palabras sea cercano a las 14 palabras. Se han identificado varios casos donde se necesita la síntesis de texto corto tal como preguntas en conferencias, preguntas y respuestas en aulas de clases o reuniones de negocio.

Fomentar la participación de la audiencia durante una presentación es una tarea difícil para el presentador porque implica detenerse a escuchar o recibir preguntas o comentarios de la audiencia [7], [13], [14], siendo necesario procesar y clasificar esta información en tiempos breves para cubrir la mayor cantidad de preguntas o respuestas posibles durante la presentación, lo que podría ser factible sólo con asistentes [15].

Situaciones similares se han reportado en educación y negocios. En educación, por ejemplo, se hace difícil analizar los comentarios realizados en encuestas de los estudiantes y así entender su experiencia durante los cursos [16], resumir contenidos de las clases [17] o las interacciones en las plataformas de aprendizaje [18]. En los negocios, el análisis de textos cortos puede revelar los problemas más frecuentes de transporte en las zonas urbanas [19], puede clasificar las opiniones de los productos [20], lo que puede mostrar una mejora potencial del apoyo y la garantía de los negocios [2].

En estos estudios, el análisis textual de frases cortas y su visualización son componentes claves para interpretar los textos y dar sentido al posible contenido oculto. No obstante, dichos análisis dependen del observador o investigador y de la capacidad de la herramienta para personalizar las consultas y representar los datos.

III. Trabajos Relacionados

Esta sección examina los trabajos actuales sobre la agrupación de textos cortos que pueden ser aplicados a un contexto de agrupación de texto corto en vivo, técnicas de minería de textos y sus formas de validación.

A. Agrupamiento de Texto Corto

La tendencia actual de la síntesis de texto es la aplicación de técnicas de extracción de datos para clasificar, agrupar y asociar textos [21]. Los algoritmos utilizados implican la indexación, codificación y agrupación de textos. Éstos pueden clasificarse en dos categorías: agrupación por frecuencia de términos y agrupación por corpus.

La agrupación por frecuencia de términos utiliza únicamente los datos del texto para crear un espacio vectorial por la frecuencia de los términos en los documentos, excluyendo los términos de detención y transformando cada palabra en raíces de palabras. La agrupación de textos cortos mediante frecuencia de términos ha dado resultados razonables con los algoritmos *k-means* y *tf-idf*, pero los textos cortos tienen en promedio más de 14 palabras y no se reporta cuanto tiempo demoran. En [22], trabajaron con agrupación jerárquica aglomerada (AHC) y agrupación espectral (SPEC), una variante de k-means con una transformación de los vectores, para agrupar resúmenes y noticias. Los autores reportaron un mejor rendimiento con AHC y el SPEC en comparación a otras formas de agrupaciones jerárquicas. En [23] y [24], los autores mejoraron el rendimiento de k-means con un algoritmo de partículas de enjambre y su mutación para encontrar mejores centroides que la solución inicial. En [25] propusieron una agrupación difusa después de la vectorización por el método de bolsa de palabras (BOW, por sus siglas en inglés, *Bag-of-Words*) para las noticias y sus resultados fueron mejores que los de otros métodos como el modelo de temas bitérmino (BTM, por sus siglas en inglés, *Biterm Topic Model*) y la asignación de Dirichlets latentes (LDA, por sus siglas en inglés, *Latent Dirichlets Allocation*). El conjunto de datos utilizado por los autores fue de un mínimo de 12340 noticias con una longitud de 15 palabras, lo que se considera datos suficientes para vectorizar al menos 20 grupos. En [26], los autores agruparon entre 2280 a 20000 noticias con AHC y un k-means iterativo. Los autores reportaron una mejora en el agrupamiento cuando eliminaron los documentos atípicos antes de la segunda aplicación de k-means. En [27] utilizaron las etiquetas de microblogs como Twitter para identificar el título y los datos del cuerpo, procesando la posición de cada elemento del post. La propuesta es interesante porque fusiona dos modelos, BTM y GloVe, para generar vectores que pueden ser agrupados en una versión modificada de k-means. Su fuente de datos son miles de microblogs con más de 15 palabras. En resumen, los trabajos de agrupación por frecuencia de términos aplicado a textos cortos utilizan modificaciones del método k-means y de agrupamiento jerárquico para procesar el texto corto. Estos métodos son una buena aproximación para esta investigación. Sin embargo, los trabajos revisados utilizaron grandes conjuntos de datos (miles), lo cual afecta al agrupamiento resultante [28], más de 15 palabras y no informan el tiempo de procesamiento para agrupar los datos.

La agrupación por corpus considera el contexto para agrupar el texto. Este método analiza las conexiones entre términos, frases, discurso y la parte de la oración. Ejemplos de conexión entre términos se puede encontrar en [26],[28], donde utilizan redes de palabras (WordNets) para detectar términos similares. Estas redes son extraídas desde Wikipedia o tesauros para cambiar el proceso de vectorización al identificar términos similares, aplicando algunos de los métodos mencionados como k-means para agrupar el texto. La agrupación utilizando partes de la oración implica analizar la estructura del texto para identificar sustantivos, verbos y su conexión. Según [32], la identificación de estos elementos tal como sustantivos puede sopesar la falta de repetición de ciertas palabras para agrupar elementos o crear resúmenes. En [33] proponen la identificación de las palabras de transición y las relaciones anafóricas para crear una matriz de peso que permita agrupar mediante el algoritmo de Markov, mostrando un mejor rendimiento frente a los métodos de k-means. En [34], los autores propusieron el análisis de conectores como comas e "y" para agrupar y extraer frases esenciales que resuman el documento. Estas formas de agrupación y resumen implican otros elementos para descubrir el conocimiento, y la agrupación reportada parece ser mejor que los métodos basados en k-

means. Sin embargo, el uso de redes de palabras puede sacrificar el rendimiento [35]. Del mismo modo que los trabajos que agrupan por frecuencia de términos, los trabajos revisados no informan el tiempo de procesamiento y sus fuentes de datos son miles de palabras. Esto afecta al proceso de caracterización cuando el texto procesado se compone de cientos de textos en contraste a los miles utilizados. De forma concreta, al existir menos texto y al ser de longitud corta, hay una menor probabilidad de encontrar palabras que caractericen los datos en contraste con una fuente de datos más grande.

B. *Técnicas de Minería de Texto*

La minería de textos combina técnicas para transformar los datos en una forma manejable para clasificar y agrupar textos. La minería de textos incluye los siguientes pasos:

Indización: El proceso de indización o indexación elimina las palabras que no añaden un valor semántico, como los artículos. Posteriormente, se lleva a cabo el proceso de lematización o enraizamiento, que convierte las palabras restantes a una forma más simplificada de posibles conjugaciones [36].

Codificación: Este método transforma el texto en una forma matemática como puntos o vectores. El método de codificación más común es el método TF-IDF, que utiliza el corpus completo para obtener la frecuencia de repetición de cada término [37].

Agrupamiento: Este paso agrupa los datos transformados. El método k-means es uno de los más utilizados, ya que proporciona agrupaciones en tiempos razonables, y funciona como un método de comparación para evaluar el rendimiento con otros métodos [33], [35]. Este algoritmo es efectivo, pero la elección de su solución inicial puede afectar a la solución final [38]. Sin embargo, según los trabajos de [23] y [22], el rendimiento del método k-means no difiere de otras formas de agrupación cuando se procesa una gran cantidad de corpus.

C. *Eficiencia en Minería de Texto*

La eficiencia de los algoritmos de agrupamiento se ha evaluado comparando el resultado con una agrupación realizada por personas [26]. Hay dos maneras de evaluar los grupos, una evaluación interna y otra externa.

Índice de clasificación interna: La clasificación interna mide los elementos del grupo, como puntos, vectores y centroides, para ver cuán eficiente es la partición. La métrica más utilizada para k-means es la distancia de cada vector o punto a su centroide, pero también se utiliza la distancia entre los centros. La distancia entre los elementos a su centroide también se suele utilizar para encontrar el número ideal de particiones (K) de un conjunto de datos con el método del codo o la silueta [39]. Henning [28] propone la distancia de cada elemento a su centroide del grupo como un índice.

Índice de clasificación externa: El índice de clasificación externa, también conocido como "el estándar de oro", comprueba si los grupos tienen sentido para un observador externo [40]. Aunque existen varias métricas para evaluar los grupos tal como la puntuación F, la entropía o la pureza, aún no existe un consenso sobre sobre cuál de ellas es más eficaz para evaluar grupos de textos cortos [22]. En este trabajo se utiliza el índice de pureza, dado que se realizó más de un test con personas, el cual incluía el estándar de oro y la validación de agrupamientos generados por el algoritmo de forma independiente [33], [35].

IV. Preguntas de Investigación e Hipótesis

Tal como se indica en las secciones anteriores, el agrupamiento de textos cortos tiene avances, pero no se consideran tres elementos importantes cuando se desea realizar una aplicación práctica:

Tiempos mínimos: Esta limitación depende del contexto debido a la cantidad de datos y el tiempo de espera. Por ejemplo, estos factores son diferentes para una presentación en vivo comparado con la visualización de datos en un contexto comercial debido a que el primero apela a la paciencia de la audiencia. Según [41], tres segundos son suficientes para que el 38% de los usuarios abandonan las páginas webs cuando existe una retroalimentación, llegando al 70% cuando esta no existe.

Interactivo: La visualización del texto debe apoyar el juicio del intérprete. Siendo necesario personalizar la cantidad de agrupamientos y su resumen para su potencial visualización [5].

Confiable: La agrupación debe ser coherente al usuario siendo necesaria la evaluación de un método con una data que cumpla las características de texto corto. Esta evaluación debe ser realizada por personas y se debe validar que la elección de un representativo sea coherente con el agrupamiento.

Las siguientes preguntas de investigación resumen los requisitos anteriores:

P1: ¿Hasta cuantas frases de texto corto en español pueden procesar los métodos de minería de texto en menos de tres segundos?

P2: ¿En cuánto difiere el rendimiento del tiempo de la minería de textos al agrupar texto corto en comparación con texto más largo? ¿Existirá una diferencia tanto en tiempo como inercia al variar una cantidad de texto y/o agrupamientos?

P3: ¿Podrá la minería de texto lograr un rendimiento similar con texto corto al ser evaluado por un humano?

P4: ¿Es el principio de agrupación plana una forma efectiva de seleccionar una frase representativa?

Por lo tanto, es factible proponer las siguientes hipótesis para cada pregunta de investigación:

H1: Las técnicas de minería de textos pueden agrupar grupos personalizados y diferentes cantidades de texto corto en menos de tres segundos.

H2: El tiempo o rendimiento del algoritmo depende del número de textos o frases, pero no del número de agrupamientos.

H3: La inercia del algoritmo depende del número de frases y número de agrupamientos.

H4: La evaluación de la agrupación de texto corto realizada por una persona mostrará un rendimiento similar a una evaluación de texto más largo.

H5: Un texto corto elegido posterior al agrupamiento puede representar a su grupo.

V. Solución Propuesta

Este problema fue analizado con la definición de

automatización propuesta por [42]. Esta taxonomía propone diez niveles de automatización donde el nivel superior no requiere ninguna intervención humana. La agrupación de textos cortos para la presentación en vivo necesita mantener las interacciones entre el presentador y el público, de modo que los dos niveles superiores se descartan. Por esta razón, el problema se centra en el nivel tres, porque este nivel reduce el conjunto de opciones para el usuario. En este caso, el conjunto de opciones son los datos del texto, que se analizan para proporcionar algunos elementos, donde el usuario puede definir y repetir el proceso. El proceso comienza cuando el usuario quiere cargar los datos. Luego el usuario pide la agrupación del texto que puede incluir el número de agrupaciones para ver si la agrupación puede tener sentido para él. El proceso de agrupación de frases cortas incluye la indexación, codificación y agrupación de datos. El algoritmo propuesto considera la misma base con una variante en la elección del más representativo. El algoritmo sigue los siguientes pasos:

1. Indizado: Se divide un texto corto en varias palabras. Se eliminan palabras de detención como artículos y las reduce a una forma de raíz (Ej: "clases" a "clas"). El algoritmo utilizado es el de bola de nieve para español y Steem Porter en el caso de inglés [43].
2. Codificación a través de TF-IDF: Se aplica el algoritmo TF-IDF explicado en [44]. Este algoritmo cuenta las repeticiones de cada palabra en una frase, y la cantidad de veces que aparece en el resto de las frases para compensar palabras que se repitan mucho. En este algoritmo, se remueven artículos y puntuaciones para no afectar la matriz resultante. El resultado es una matriz dispersa que tiene sus vectores normalizados.
3. Agrupación con k-means: Se aplica el algoritmo k-means explicado en [45]. La medida de agrupamiento es la distancia euclidiana entre los vectores normalizados a un centro elegido aleatoriamente. La elección del centro va mejorando a medida que se itera el algoritmo. Según [46], utilizar la distancia euclidiana con vectores normalizados genera resultados similares a utilizar la distancia del coseno como medida de similitud.
4. Selección de frase representativa por grupo: Se propone la distancia entre cada frase a su centroide como criterio de selección de la frase representativa. Sin embargo, al realizar pruebas iniciales se pudo observar que, al existir poco texto, muchos vectores, potencialmente representativos, tendían a empatar, ubicándose al centro del grupo. Se propone como variante, utilizar la suma de distancias a cada centroide de los grupos. El primer algoritmo suma la distancia a cada centroide y selecciona la frase cuya distancia total es la mínima por grupo. El segundo algoritmo, utiliza la distancia total máxima para evitar un posible resultado proveniente de la aleatoriedad del algoritmo.

El algoritmo 1 es el algoritmo base donde tanto el indizado, TFIDF y k-means pueden ser vistos en detalle en los trabajos [43]–[45]. El algoritmo 2 muestra el cálculo del representativo, donde la única diferencia es la elección de la frase con la distancia total mínima versus la máxima. Esto cambiaría la línea $R_{[i]} = Max(D_i)$ a $R_{[i]} = Min(D_i)$.

Algoritmos 1: Agrupamiento de sentencias.

Entrada:
T_i ← Texto en archive con i textos cortos (uno por línea).
K ← Número de grupos pedidos por el usuario.
Salida:
K grupos indicando los textos por grupo junto con el texto que representa al grupo.

S ← Indizado (T)
M ← $TFIDF$(S)
G, C ← KMean(M, K)
R ← Representativo(G, C)
Devolver (G, R)

Algoritmo 2: Selección de representativo con distancia máxima.

Entrada:
G_i ← Grupo i de frases
C_i ← Centroide i por cada grupo
Salida:
R_i Frases representativas

$R_i = [\,]$
Para Cada G_i Entonces:
  $D_i = [\,]$
  Para Cada $Frase_i$ en G_i
    $Dist = 0$
    Para Cada C_i Entonces:
      $Dist = Dist + Distancia(Frase_i, C_i)$
    Fin Para
  $D_{[i]} = Dist$
  Fin Para
  $R_{[i]} = Max(D_i)$
Fin Para
Devolver (R_i)

VI. Resultados

En esta sección se muestra la configuración del experimento y los resultados de la aplicación del algoritmo.

A. Configuración del Experimento

El algoritmo se probó en español con una encuesta solicitada a estudiantes al finalizar un módulo de anatomía. Los datos fueron limpiados usando los siguientes criterios: Se arreglaron los errores de ortografía, los datos fueron anonimizados; Cada frase larga se dividía en una idea por cada frase corta. Se eliminaron los caracteres especiales como los emoticones.

Después de limpiar los datos de la encuesta, el archivo quedó con 368 líneas con un texto corto (frase) por línea con un largo promedio de 11 palabras. Utilizando este archivo como base se generaron aleatoriamente 10 archivos con 5, 10, 20, 30, 45, 60, 100, 150 y 200 textos cortos (líneas) respectivamente. Este subconjunto de texto representa la recepción de una pregunta o comentario por participante en condiciones de reunión (5 y 10 textos), aulas (20, 30 y 45 textos) y conferencias (60, 100, 150 y 200 textos). Además, se descargaron los datos de la colección Reuters-21578 [47] y los datos del 20 NewsGroup [48] para comparar los resultados con las fuentes de datos utilizados en otras investigaciones [6], [23], [29]. El conjunto de datos de Reuters corresponde al archivo llamado reut2-001.sgm, que fue convertido a una versión csv con dos campos, la clasificación y el documento en una línea. Por otro lado, el conjunto de datos de NewsGroup es una selección de 100 noticias en cinco categorías, eliminando encabezados, pies de página y citas. Esto fue guardado en un formato csv con dos campos al igual que el documento de Reuters. La Tabla 1 muestra un resumen de los

datos del experimento, donde se muestran el número de archivos (N), los largos medios en número de palabras (M), la desviación estándar (SD), el texto más corto (Min) y el más largo (Max) de los datos utilizados medidos en palabras.

TABLA I
NÚMERO DE TEXTOS (N) Y MEDIA (M) DE PALABRAS POR TEXTO

Texto	N	M	SD	Min	Max
Cortos	362	10.85	5.00	1	28
Medios	497	114.22	118.23	11	787
Largos	490	164.46	352.25	3	4950

Según la Tabla 1, se ha catalogado a los datos de la investigación como textos cortos, los datos de Reuters como textos medios y los textos de 20 News como textos largos. Un análisis de varianza (ANOVA) sobre el largo de los textos dio como resultado diferencias de medias significativas entre los textos, $F(2, 1346) = 49.597$, $p < .001$. La prueba Post-Hoc Bonferroni dio como resultado diferencias significativas entre textos cortos y medios ($p < .001$), cortos y largos ($p = .001$), y entre medios y largos ($p = .001$).

El algoritmo fue construido en Python 3.8 con los siguientes paquetes: Nltk: Esta librería contiene funciones para procesar lenguajes naturales como stemmer Porter y Snowball stemmer; Sklearn: Esta biblioteca tiene funciones científicas como Kmeans y TfidfVectorizer; Numpy: Esta biblioteca está optimizada para trabajar con vectores y matrices.

El programa se ejecutó en un computador con procesador Ryzen 5 Modelo 2500 con Radeon Vega Mobile Gfx 2.00 GHz, y 16 Gb Ram. Las métricas para evaluar el algoritmo son las siguientes.

- Tiempo: La agrupación de la información debe hacerse en tiempos humanamente aceptables, así que se medirá el tiempo (en segundos) de procesamiento de los datos.
- Inercia: Corresponde a la distancia entre cada elemento y su centroide, dividida por el número de elementos.
- Inercia Centroide: Inercia promedio de cada agrupamiento.
- Calidad: Corresponde a la pureza de los grupos comparada con la agrupación hecha por personas. Se seleccionaron examinadores externos de diferentes antecedentes y posiciones en la organización (universidad) para evitar un posible sesgo sistémico en la evaluación [49].

B. Evaluación Interna (P1)

Los resultados internos pueden comparar el rendimiento del proceso de agrupación evaluando los tiempos de ejecución e inercia. El algoritmo fue ejecutado para demostrar si había una diferencia significativa cuando cambia el número de palabras de los textos (cortos, medios y largos). Además, se quiso ver si existía diferencias entre los textos cortos y sus posibles variaciones, por esta razón se probaron desde 2 hasta 10 agrupamientos para archivos con más de 10 textos o frases. Para archivos con más de 10 textos se crearon hasta 9 agrupaciones (de tamaño 2 al 10), para archivos de 5 textos se crearon hasta 3 agrupaciones (de tamaño 2, 3 y 4), y para 10 textos se crearon 8 agrupaciones (de tamaño 2 al 9). Por ejemplo, para el caso de 20 textos cortos se crearon 10 archivos aleatorios utilizando como archivo base el archivo de 368 líneas de texto. Por cada uno de estos archivos, se crearon desde 2 a 10 agrupamientos. En el caso de los textos medios y largos, sólo se creó un archivo por cada condición y el mismo número agrupaciones. Además, estos ya poseían una clasificación realizadas por personas [6], [23], [29]. La Tabla 2 muestra el tiempo de simulación para los casos propuestos.

TABLA II
TIEMPO PARA TEXTOS CORTOS, MEDIOS Y LARGOS

Frases o textos	Agrupa-mientos	Cortos (N=74) M (SD)	Medios (N=74) M (SD)	Largos (N=74) M (SD)
5	3	0.22 (0.01)	0.37 (0.02)	0.66 (0.03)
10	8	0.26 (0.04)	0.60 (0.04)	0.54 (0.06)
20	9	0.30 (0.02)	0.86 (0.08)	0.64 (0.08)
30	9	0.35 (0.02)	1.12 (0.06)	4.65 (0.12)
45	9	0.47 (0.03)	2.42 (0.06)	1.95 (0.12)
60	9	0.58 (0.02)	2.62 (0.06)	3.94 (0.11)
100	9	0.78 (0.03)	4.22 (0.14)	8.47 (0.34)
150	9	1.07 (0.04)	6.73 (0.12)	9.90 (0.32)
200	9	1.35 (0.09)	8.48 (0.20)	17.07 (0.35)

Los tiempos promedios (M) y sus desviaciones estándares (SD) para la variable tiempo de ejecución del algoritmo fueron los siguientes: textos cortos (M = 0.63, SD = 0.37), medios (M = 3.30, SD = 2.73), y largos (M = 5.75, SD = 5.33), con tamaños del efecto Cohen's d = 1.37 (textos cortos versus medios), d = 1.35 (textos cortos versus largos). Un análisis de varianza (ANOVA) sobre el logaritmo de la variable tiempo dio como resultado diferencias de medias significativas entre los textos, $F(2, 219) = 73.889$, $p < .001$. La prueba Post-Hoc Bonferroni dio como resultado diferencias significativas entre textos cortos y medios ($p < .001$), cortos y largos ($p < .001$), y entre medios y largos ($p = .041$).

A continuación, se realizó una inspección gráfica (ver Fig. 1) para el número de frases y para la variable número de agrupamientos en relación con la variable tiempo de ejecución.

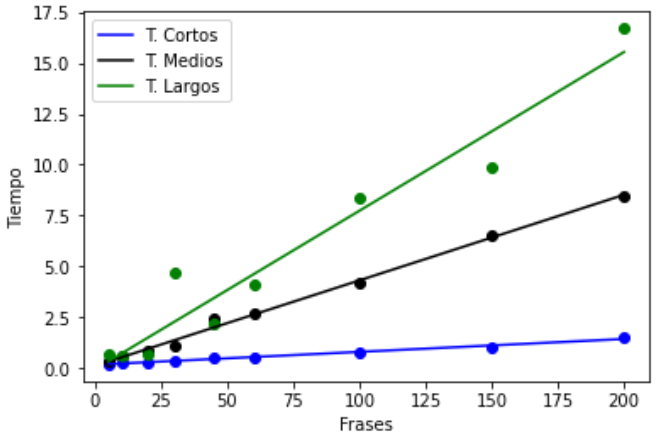


Fig. 1. Tiempo del Algoritmo por Frases.

Los números de los agrupamientos del gráfico fueron elegidos mediante el algoritmo del codo, que permite identificar el mejor número de grupos al analizar visualmente las diferencias en las inercias cuando los grupos aumentan. Se encontró una relación lineal positiva para los tres conjuntos de

textos (corto, medio y largo), número de frases y tiempo de ejecución, pero no para número agrupamientos, donde el tiempo se mantuvo relativamente constante para cada número de grupos. Por lo tanto, se ejecutó un modelo de regresiones múltiples para predecir el tiempo de ejecución (transformado con logaritmo) a partir de las variables conjunto de datos, número de frases y número de agrupamientos.

El modelo de regresiones múltiples predice significativamente el tiempo de ejecución del algoritmo, $F(3, 218) = 334.078$, $p < .001$, R2 ajustado = .819. Las variables que participan en la predicción son largo del texto (corto = 0, medio = 1, largo = 2), B1 = .392, 95% IC [0.355 – 0.429], $p < .001$, y número de frases, B2 = .006, 95% IC [0.005 – 0.006], $p < .001$), pero no la variable agrupamiento (B3 = .010, 95% IC [-0.002 – 0.022], $p = .089$). Se acepta hipótesis H2.

Por lo tanto, el largo de los textos, junto con la cantidad de frases procesadas impactan fuertemente los tiempos de procesamientos del algoritmo, pero no el número de agrupamientos involucrados.

TABLA III
INERCIA PARA TEXTOS CORTOS, MEDIOS Y LARGOS

Frases	Agrupa-mientos	Cortos (N=74) M (SD)	Medios (N=74) M (SD)	Largos (N=74) M (SD)
5	3	1.75 (0.95)	1.87 (0.96)	1.89 (0.99)
10	8	4.01 (2.37)	3.96 (2.36)	4.27 (2.40)
20	9	7.05 (3.29)	8.78 (3.04)	10.34 (3.19)
30	9	8.77 (4.27)	13.95 (3.47)	20.78 (3.33)
45	9	8.62 (6.11)	24.28 (4.50)	25.01 (4.35)
60	9	13.99 (8.15)	33.53 (5.49)	42.88 (4.34)
100	9	18.62 (12.47)	52.37 (8.24)	75.06 (5.26)
150	9	24.39 (18.56)	79.57 (12.05)	107.17 (9.68)
200	9	32.81 (24.58)	107.11 (14.15)	136.99 (11.8)

Por otro lado, la validación tanto de H3, H4, y H5, implican revisar la dispersión del agrupamiento mediante la observación del índice de inercia. Este índice es la suma de las distancias de cada punto al centro de su grupo, y es utilizado como una medida de calidad porque puede comparar los métodos de agrupamiento [28]. La inercia (ver Tabla 3) es un índice que puede comparar la selección del centroide entre grupos. Cuanto menor sea la inercia, mejor será el algoritmo de agrupamiento. Para el caso de los textos cortos la inercia reportada corresponde a los promedios para cada condición de agrupamiento en los 10 archivos. En este estudio, los promedios (M) y desviaciones estándares (SD) para la variable inercia fueron los siguientes: textos cortos (M = 14.40, SD = 15.01), medios (M = 39.37, SD = 35.52), largos (M = 51.40, SD = 46.40). Un análisis de varianza (ANOVA), después de aplicar la función logaritmo sobre la variable inercia, las diferencias de medias resultaron significativas entre los textos, $F(2, 219) = 17.999$, $p < .001$. La prueba Post-Hoc Bonferroni dio como resultado diferencias significativas entre textos cortos y medios ($p < .001$), cortos y largos ($p < .001$), pero no entre medios y largos ($p = .706$).

Al transformar la variable inercia con la función logaritmo, se obtiene una relación lineal negativa para la variable agrupamientos con los tres largos de textos; y una relación positiva para la variable número de frases. Por lo tanto, se ejecutó un modelo de regresiones múltiples para predecir la inercia a partir de las variables largo del texto (corto = 0, medio = 1, largo = 2), número de frases y número de agrupamientos. El modelo de regresiones múltiples predice significativamente la inercia del algoritmo, $F(3, 218) = 177.178$, $p < .001$, R2 ajustado = .705. Las variables que participan en la predicción son largo del texto, B1 = .238, 95% IC [0.190 – 0.287], $p < .001$, número de frases, B2 = .006, 95% IC [0.006 – 0.007], $p < .001$), y la variable agrupamiento (B3 = -.043, 95% IC [-0.058 – -0.027], $p < .001$). Así, se acepta H3.

Se observa que es más baja la inercia con mayor número de textos cortos que textos más largos. Esto significa que los grupos son más densos (contienen un mayor número de textos), lo que se ve influido por el número de grupos por muestra. Este índice es útil para determinar qué muestra contiene un mejor agrupamiento para compararlo entre los archivos utilizados.

C. Resultados Externos (P2 y P3)

La comprobación de H4 y H5 necesita una evaluación externa utilizando examinadores humanos. Los criterios para seleccionar la muestra se basan en el índice de inercia presentado en la sección IV.B. Por ejemplo, el siguiente conjunto de datos representa la inercia por grupo para 10 archivos con 30 frases cortas con 7 agrupamientos. El primer número representa el archivo y el segundo su inercia: (1, 5.46), (2, 5.26), (3, 8.71), (4, 8.39), (5, 2.78), (6, 9.07), (7, 7.01), (8, 6.12), (9, 5.36) y (10, 8.07). Se puede observar que la inercia del quinto archivo es la más baja con 2.78, con una diferencia de 6.28 en comparación con el archivo con la inercia más alta. Utilizando este criterio, se eligieron dos archivos con la inercia más baja. Uno con 7 agrupamientos para 30 frases cortas y otros con 6 agrupamientos con 100 frases cortas, como el extracto mostrado en Fig. 2, del archivo de 30 frases con 7 grupos.

```
cluster 0: Mayor cantidad de actividades prácticas
        sentence 0: organizar mejor las yincanas, práctica y de imagen
        sentence 1: Mayor cantidad de actividades prácticas
        sentence 2: Que se pudieran mantener las solemnes teórica y practica en instancias separadas
        sentence 3: prácticos más ordenados
        sentence 4: Que se mantenga la participación de todos en los prácticos
        sentence 5: las pruebas teóricas y prácticas hacerlas por separado creo que generaría una mejora en ambas pruebas
cluster 1: que hagan mas preguntas dirigidas a cada uno en cada una de las estaciones
        sentence 0: Mas exigencia con la asistencia de clases.
        sentence 1: Los trabajos permiten que se entienda mejor las relaciones y de una manera mas didactica.
        sentence 2: que hagan mas preguntas dirigidas a cada uno en cada una de las estaciones
        sentence 3: Creo que se podria mejorar y progundizar mas las clases de imagenología
cluster 2: Agradezco la preocupación por parte de los profesores respecto a la integridad total del alumno
        sentence 0: Más clases con el profesor R
        sentence 1: Agradezco la preocupación por parte de los profesores respecto a la integridad total del alumno
        sentence 2: Le asignaría más clases al profesor v
cluster 3: Me gustaría destacar la buena disposición que los profesores tenían cada día
        sentence 0: los profesores son muy buenos, me gustaría que se mantuvieran.
        sentence 1: Me gustaría destacar la buena disposición que los profesores tenían cada día
        sentence 2: Es bueno que se mantenga un buen ambiente, como el uso de bromas y de respeto entre alumnos y docentes.
```

Fig. 2. Extracto Archivo de 30 frases y 7 grupos.

A continuación, se solicitó a tres examinadores externos que realicen 7 grupos para 30 frases y 6 grupos para 100 frases, con al menos 2 frases por grupo. Luego, se pidió a otros tres examinadores que comprueben si cada frase fue clasificada correctamente en cada grupo encontrado por el algoritmo y que comprueben si la frase representativa resume el contenido del grupo. El puntaje de Cohen Kappa fue calculado para determinar el nivel de acuerdo de los examinadores que crearon los grupos, resultando en k = .62 para 30 frases y k = .59 para 100 frases. Estos puntajes son considerados moderados [50].

La Tabla 4 muestran los resultados para un agrupamiento con 30 y 100 textos cortos. La pureza para los textos cortos fue contrastada con los agrupamientos realizados por los examinadores. En el caso de los textos medios y largos, fueron contrastados por los grupos definidos en sus respectivas fuentes

de datos. La columna Chk muestra la validación realizada a los agrupamientos creados por el algoritmo de forma independiente a una agrupación pre-definida. En este caso, se solicitó a los examinadores validar que existiera algún patrón en cada agrupamiento, junto con los textos que pertenecieran a ese grupo. Esto permitió contar una cantidad de aciertos por grupo, los cuales fueron divididos por la cantidad de frases. La columna representativos corresponde al porcentaje de aciertos, medido como la cantidad de textos cortos que los examinadores validaron como representativo del grupo, dividido por la cantidad de grupos. La columna representativos está dividida en la validación del algoritmo utilizando el criterio min (AL1) y el criterio max (AL2).

TABLA IV
PUREZA Y REPRESENTATIVO PARA 30 Y 100 FRASES

	Pureza				Representativos	
Frases	Cortos	Medios	Largos	Chk	AL1	AL2
30	0.63	0.52	0.66	0.76	0.62	0.57
100	0.39	0.43	0.63	0.81	0.78	0.50

La hipótesis H4 se acepta ya que la pureza se encuentra entre el rango reportado por [33] y [35], cuyos valores están entre 0.324 y 0.875 de pureza. Además, la pureza para textos largos y medios es similar a la pureza para las frases cortas.

Por otro lado, la frase representativa para 30 frases y 100 frases es más del 50%. El rendimiento en el algoritmo de la distancia mínima total (AL1) alcanzó un mejor rendimiento que el de máximos. Se acepta el H5 con el algoritmo AL1 porque su rendimiento es superior al 61%.

VII. CONCLUSIÓN

Este trabajo ha mostrado el resultado de métodos conocidos de minería de textos como la agrupación por k-means y la vectorización con TFI-DF, cuyo rendimiento en tiempo es superior con textos de tamaño corto (ej.: frases de 14 palabras) comparado con textos de tamaño medio o largo (más de 160 palabras) independiente de número de agrupamientos. Esto es relevante en situaciones de interacción en tiempo real como conferencias, reuniones o clases donde se requiere un tiempo de respuesta razonable para el presentador y su audiencia. En nuestro estudio se demuestra que para 200 frases cortas y 9 agrupamientos el algoritmo tarda 1.35 segundos comparado con los 17.07 segundos para textos largos. Además, se demostraron niveles aceptables de agrupamiento de frases tanto desde el punto de vista de métricas algorítmicas como son la pureza como la agrupación realizada por observadores externos.

Futuras investigaciones estarán orientadas a la elección del número de grupos en tiempos razonables y mejorar la vectorización. Se observaron grupos que no tenían un patrón claro cuando su número aumentaba, y varios empates en la elección del texto representativo. Posiblemente, al agregar información contextual a la vectorización se podría lograr una agrupación que incluya niveles semántico, pragmático y/o social [51]. Además, es factible que los conocimientos del examinador puedan inducir diferencias para evaluar agrupaciones.

Referencias

[1] A. N. Srivastava and M. Sahami, *Text Mining*, 1st ed. New York: Chapman and Hall/CRC, 2009.

[2] W. F. Cody, J. T. Kreulen, V. Krishna, and W. S. Spangler, "The integration of business intelligence and knowledge management," *IBM Syst. J.*, vol. 41, no. 4, pp. 697–713, 2002, doi: 10.1147/sj.414.0697.

[3] A. I. De Lima, A. B. Argenta, I. C. Zattar, and M. Kleina, "Applying Text Mining to Identify Photovoltaic Technologies," *IEEE Lat. Am. Trans.*, vol. 17, no. 5, pp. 727–733, 2019, doi: 10.1109/TLA.2019.8891940.

[4] R. Neto, R. Ribeiro, and A. Emília, "Investigating the influence of groups of variables on the task of predicting the age of an author in blog posts," *IEEE Lat. Am. Trans.*, vol. 18, no. 5, pp. 838–844, 2020, doi: 10.1109/TLA.2020.9082911.

[5] S. Schreibman, R. R. Siemens, and J. Unsworth, *A New Companion to Digital Humanities*. Chichester, UK: John Wiley & Sons, Ltd, 2015.

[6] L. Kotlerman, I. Dagan, and O. Kurland, "Clustering small-sized collections of short texts," *Inf. Retr. J.*, vol. 21, no. 4, pp. 273–306, 2018, doi: 10.1007/s10791-017-9324-8.

[7] C. Greenhow, J. Li, and M. Mai, "From tweeting to meeting: Expansive professional learning and the academic conference backchannel," *Br. J. Educ. Technol.*, vol. 50, no. 4, pp. 1656–1672, Jul. 2019, doi: 10.1111/bjet.12817.

[8] W. M. Kappers and S. Cutler, "Poll Everywhere! Even in the classroom: An investigation into the impact of using PollEverywhere in a large-lecture classroom," *Comput. Educ. J.*, vol. 6, no. 20, pp. 140–145, 2015.

[9] H. R. Ponce, R. E. Mayer, V. A. Figueroa, and M. J. López, "Interactive highlighting for just-in-time formative assessment during whole-class instruction: effects on vocabulary learning and reading comprehension," *Interact. Learn. Environ.*, vol. 26, no. 1, pp. 42–60, 2018, doi: 10.1080/10494820.2017.1282878.

[10] H. R. Ponce, R. E. Mayer, M. J. López, and M. S. Loyola, "Adding interactive graphic organizers to a whole-class slideshow lesson," *Instr. Sci.*, vol. 46, no. 6, pp. 973–988, Dec. 2018, doi: 10.1007/s11251-018-9465-1.

[11] M. Compton and J. Allen, "Student Response Systems: a rationale for their use and a comparison of some cloud based tools," *Compass J. Learn. Teach.*, vol. 11, no. 1, pp. 267–271, Apr. 2018, doi: 10.21100/compass.v11i1.696.

[12] J. Mikk, "Sentence length for revealing the cognitive load reversal effect in text comprehension," *Educ. Stud.*, vol. 34, no. 2, pp. 119–127, May 2008, doi: 10.1080/03055690701811164.

[13] L. M. Goldstein and S. M. Conrad, "Student Input and Negotiation of Meaning in ESL Writing Conferences," *TESOL Q.*, vol. 24, no. 3, p. 443, 1990, doi: 10.2307/3587229.

[14] J. R. Cox, "Enhancing student interactions with the instructor and content using pen-based technology, youtube videos, and virtual conferencing," *Biochem. Mol. Biol. Educ.*, vol. 39, no. 1, pp. 4–9, Jan. 2011, doi: 10.1002/bmb.20443.

[15] L. M. Shah, E. P. Quigley, and R. H. Wiggins, "The Next Wave: Confexting," *J. Digit. Imaging*, vol. 25, no. 1, pp. 25–29, 2012, doi: 10.1007/s10278-011-9398-6.

[16] X. Chen, M. Vorvoreanu, and K. P. C. Madhavan, "Mining Social Media Data for Understanding Students' Learning Experiences," *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 246–259, Jul. 2014, doi: 10.1109/TLT.2013.2296520.

[17] A. Shimada, F. Okubo, C. Yin, and H. Ogata, "Automatic Summarization of Lecture Slides for Enhanced Student Preview - Technical Report and User Study-," *IEEE Trans. Learn. Technol.*, vol. 11, no. 2, pp. 165–178, Apr. 2018, doi: 10.1109/TLT.2017.2682086.

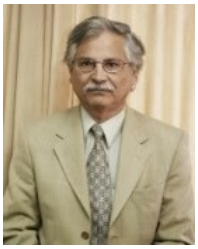
[18] P. Vrablecova and M. Simko, "Supporting Semantic Annotation of Educational Content by Automatic Extraction of Hierarchical Domain Relationships," *IEEE Trans. Learn. Technol.*, vol. 9, no. 3, pp. 285–298, Jul. 2016, doi: 10.1109/TLT.2016.2546255.

[19] A. Szmelter-Jarosz and J. Rześny-Cieplińska, "Priorities of Urban Transport System Stakeholders According to Crowd Logistics Solutions in City Areas. A Sustainability Perspective," *Sustainability*, vol. 12, no. 1, p. 317, Dec. 2019, doi: 10.3390/su12010317.

[20] G. Somprasertsri and P. Lalitrojwong, "Mining feature-opinion in online customer reviews for opinion summarization," *J. Univers. Comput. Sci.*,

vol. 16, no. 6, pp. 938–955, 2010, doi: 10.3217/jucs-016-06-0938.
[21] T. Jo, *Text Mining*, 2nd ed., vol. 45. Cham: Springer International Publishing, 2019.
[22] P. Shrestha, C. Jacquin, and B. Daille, "Clustering Short Text and Its Evaluation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7182 LNCS, no. PART 2, 2012, pp. 169–180.
[23] L. Cagnina, M. Errecalde, D. Ingaramo, and P. Rosso, "An efficient Particle Swarm Optimization approach to cluster short texts," *Inf. Sci. (Ny).*, vol. 265, pp. 36–49, May 2014, doi: 10.1016/j.ins.2013.12.010.
[24] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "An Indicator-based Multi-Objective Optimization Approach Applied to Extractive Multi-Document Text Summarization," *IEEE Lat. Am. Trans.*, vol. 17, no. 8, pp. 1291–1299, 2019, doi: 10.1109/TLA.2019.8932338.
[25] J. Rashid, S. M. A. Shah, and A. Irtaza, "Fuzzy topic modeling approach for text mining over short text," *Inf. Process. Manag.*, vol. 56, no. 6, p. 102060, Nov. 2019, doi: 10.1016/j.ipm.2019.102060.
[26] M. R. H. Rakib, N. Zeh, M. Jankowska, and E. Milios, "Enhancement of short text clustering by iterative classification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12089 LNCS, Cham: Springer, 2020, pp. 105–117.
[27] D. Wu, M. Zhang, C. Shen, Z. Huang, and M. Gu, "BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery," *IEEE Access*, vol. 8, pp. 32215–32225, 2020, doi: 10.1109/ACCESS.2020.2973430.
[28] C. Hennig, "Cluster Validation by Measurement of Clustering Characteristics Relevant to the User," in *Data Analysis and Applications 1*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2019, pp. 1–24.
[29] B. Altınel and M. C. Ganiz, "Semantic text classification: A survey of past and recent advances," *Inf. Process. Manag.*, vol. 54, no. 6, pp. 1129–1153, Nov. 2018, doi: 10.1016/j.ipm.2018.08.001.
[30] Z. Qiu and H. Shen, "User clustering in a dynamic social network topic model for short text streams," *Inf. Sci. (Ny).*, vol. 414, pp. 102–116, Nov. 2017, doi: 10.1016/j.ins.2017.05.018.
[31] L. Yang, X. Cai, Y. Zhang, and P. Shi, "Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization," *Inf. Sci. (Ny).*, vol. 260, pp. 37–50, Mar. 2014, doi: 10.1016/j.ins.2013.11.026.
[32] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge: Cambridge University Press, 2015.
[33] D. Sahoo and R. Balabantaray, "A novel approach to sentence clustering," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Apr. 2016, pp. 1–6, doi: 10.1109/CCAA.2016.7813697.
[34] A. Khan *et al.*, "Abstractive Text Summarization based on Improved Semantic Graph Approach," *Int. J. Parallel Program.*, vol. 46, no. 5, pp. 992–1016, 2018, doi: 10.1007/s10766-018-0560-3.
[35] H.-T. Zheng, B.-Y. Kang, and H.-G. Kim, "Exploiting noun phrases and semantic relationships for text document clustering," *Inf. Sci. (Ny).*, vol. 179, no. 13, pp. 2249–2262, Jun. 2009, doi: 10.1016/j.ins.2009.02.019.
[36] W. B. A. Karaa and N. Dey, *Mining Multimedia Documents*. New York: Chapman and Hall/CRC, 2017.
[37] C. Zhang and J. Han, *Multidimensional Mining of Massive Text Data*, vol. 11, no. 2. Morgan & Claypool Publishers, 2019.
[38] J. Peña, J. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the K-Means algorithm," *Pattern Recognit. Lett.*, vol. 20, no. 10, pp. 1027–1040, Oct. 1999, doi: 10.1016/S0167-8655(99)00069-0.
[39] P. Bholowalia and A. Kumar, "EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN," *Int. J. Comput. Appl.*, vol. 105, no. 9, pp. 17–24, 2014.
[40] M. E. Ares, J. Parapar, and Á. Barreiro, "An experimental study of constrained clustering effectiveness in presence of erroneous constraints," *Inf. Process. Manag.*, vol. 48, no. 3, pp. 537–551, 2012, doi: 10.1016/j.ipm.2011.08.006.
[41] F. F.-H. H. Nah, "A study on tolerable waiting time: How long are Web users willing to wait?," *Behav. Inf. Technol.*, vol. 23, no. 3, pp. 153–163, May 2004, doi: 10.1080/01449290410001669914.
[42] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A Model for Types and Levels of Human Interaction with Automation," *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 30, no. 3, pp. 286–297, May 2000, doi: 10.1109/3468.844354.
[43] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 40, no. 3, pp. 211–218, Jul. 2006, doi: 10.1108/00330330610681286.
[44] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, 1st ed. Cambridge: Cambridge University Press, 2008.
[45] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Appl. Stat.*, vol. 28, no. 1, p. 100, 1979, doi: 10.2307/2346830.
[46] T. Korenius, J. Laurikkala, and M. Juhola, "On principal component analysis, cosine and Euclidean measures in information retrieval," *Inf. Sci. (Ny).*, vol. 177, no. 22, pp. 4893–4905, 2007, doi: 10.1016/j.ins.2007.05.027.
[47] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, "The UCI KDD archive of large data sets for data mining research and experimentation," *ACM SIGKDD Explor. Newsl.*, vol. 2, no. 2, pp. 81–85, 2000, doi: 10.1145/380995.381030.
[48] P. Fabian *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: http://scikit-learn.sourceforge.net.
[49] D. Fuentealba, K. Liu, and W. Li, "Organisational Responsiveness Through Signs," in *IFIP Advances in Information and Communication Technology*, vol. 477, 2016, pp. 117–126.
[50] M. L. McHugh, "Interrater Reliability: the Kappa Statistic," *Biochem. Medica*, vol. 22, no. 3, pp. 276–282, 2012, doi: 10.11613/BM.2012.031.
[51] K. Liu and W. Li, *Organisational Semiotics for Business Informatics*. London and New York: Routledge, 2014.

Diego A. Fuentealba, es Ing. en Informática e Industrial. Sus estudios de postgrado son un Master en Ing. Industrial de la Universidad de Santiago de Chile, y un Ph.D en Informática de la Universidad de Reading, Inglaterra. Es investigador en VirtuaLab-USACH. Sus intereses de investigación incluyen la semiótica organizacional, ingeniería de requerimientos, desarrollo de software y sistemas inteligentes.



Mario J. López, Ph.D. es Profesor Titular en el área de tecnologías de la información y la comunicación en el Departamento de Ingeniería Industrial de la Universidad de Santiago de Chile. Es el CEO de VirtuaLab-USACH, un laboratorio multidisciplinario para la investigación, desarrollo y transferencia de tecnologías visuales. Ha sido director de varios proyectos de investigación y desarrollo financiados con fondos públicos y privados; tiene numerosas publicaciones en revistas, capítulos de libros y congresos internacionales. Además, dirigió el equipo que obtuvo la patente estadounidense de iSlides. Sus intereses de investigación incluyen bases de datos distribuidas, desarrollo de componentes de software, innovación basada en la tecnología y aprendizaje de ciencias aplicadas.

Hector R. Ponce es licenciado en Ingeniería Informática de la Universidad de Santiago de Chile y tiene un PhD de la Universidad de Lincoln, Inglaterra. Es Profesor Titular en el área de sistemas de información en la Facultad de Administración y Economía de la USACH. Es el CSO de VirtuaLab-USACH, un laboratorio multidisciplinario para la investigación, desarrollo y transferencia de tecnologías visuales. Su área de investigación

se centra en el desarrollo de componentes de software que implementan estrategias visuales y la evaluación de su impacto en el aprendizaje.