

Aplicación de Algoritmos de Clasificación de Minería de Textos para el Reconocimiento de Habilidades de E-tutores Colaborativos

Pablo Santana Mansilla ^{1,2}, Rosanna Costaguta ², Daniela Missio ²

¹ Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

² Instituto de Investigación en Informática y Sistemas de Información (IISI), Facultad de Ciencias Exactas y Tecnologías (FCEyT), Universidad Nacional de Santiago del Estero (UNSE), Santiago del Estero, Argentina

{psantana, rosanna, dmissio}@unse.edu.ar

Abstract. Computer Supported Collaborative Learning (CSCL) systems enable group learning regardless of the time and space where students and teachers are located. However, interactions that conduct to collaborative creation of knowledge do not occur spontaneously and technology can inhibit or hinder them. Students need (among other things) of an e-tutor (teacher) that coordinates group interaction in order to effectively collaborate. Selection of qualified e-tutors is key for the success of CSCL, but manual analysis of interactions recorded in these environments to know the performance of teachers requires a lot of time and effort. This paper describes a research that applies text mining techniques to build classifiers that allow automatic identification of manifested skills by e-tutors. Results obtained by different classification algorithms are presented, analyzed and compared.

Resumen. Los sistemas de Aprendizaje Colaborativo Soportado por Computadora (ACSC) permiten el aprendizaje grupal con independencia del tiempo y espacio donde estén localizados los estudiantes y los docentes. Sin embargo, las interacciones que conducen a la creación colaborativa de conocimiento no surgen de manera espontánea y la tecnología puede inhibirlas u obstaculizarlas. Para colaborar efectivamente los estudiantes necesitan (entre otras cosas) de un e-tutor (docente) que coordine la interacción grupal. La selección de e-tutores capacitados es clave para el éxito del ACSC pero el análisis manual de las interacciones registradas en estos entornos para conocer el desempeño de los docentes requiere mucho tiempo y esfuerzo. En este artículo se describe un trabajo de investigación que aplica técnicas de minería de textos para crear clasificadores que permitan identificar automáticamente las habilidades manifestadas por e-tutores. Los resultados obtenidos mediante diferentes algoritmos de clasificación son presentados, analizados y comparados.

Keywords: Computer Supported Collaborative Learning, Skills of e-tutor, Text Mining, Automatic Identification of Skills.

1 Introduction

El soporte computacional de los sistemas de ACSC permite a los estudiantes trabajar colaborativamente ubicados en puntos geográficos distantes y contribuir en momentos diferentes en el tiempo [15]. Estos sistemas cuentan con herramientas de comunicación síncrona y/o asíncrona que soportan tanto a la comunicación como a la colaboración y coordinación entre los estudiantes. Además, los sistemas de ACSC son entornos propicios para el desarrollo de habilidades consideradas clave en un líder exitoso: solución de problemas, pensamiento crítico, mayor retención, establecimiento de metas, interpretación, y análisis [5]. Sin embargo, no basta con organizar a los estudiantes en grupos porque las interacciones que conducen a la creación colaborativa de conocimiento no surgen de manera espontánea y la tecnología puede inhibirlas u obstaculizarlas [15, 22]. En consecuencia, es

fundamental la intervención de los e-tutores para promover las interacciones entre los estudiantes, coordinar las discusiones, y tratar con conflictos que surgen durante el trabajo grupal [15,19, 22]. No obstante, la mayoría de los e-tutores no tienen experiencia con las técnicas propias del ACSC, y además se cree erróneamente que los docentes solo necesitan aprender a usar la tecnología porque en un entorno mediado por computadora se pueden seguir utilizando las técnicas propias de las aulas presenciales [14, 15].

Dado que los e-tutores de ACSC son los principales responsables del proceso de enseñanza y de brindar asistencia a los estudiantes, la selección de docentes con las habilidades necesarias para sacarle provecho a las herramientas tecnológicas disponibles es esencial para el éxito del ACSC [13, 15]. De aquí que sea necesario conocer las habilidades con las que cuentan los e-tutores para, en caso de ser necesario, capacitarlos en las habilidades que manifiestan con deficiencia. De acuerdo con Barker [1] una manera de propiciar la adquisición de habilidades es desarrollando sistemas de software que entrenen a los e-tutores. Sin embargo, los efectos negativos sobre el proceso de comunicación que tienen las técnicas de análisis de interacciones basadas en interfaces estructuradas o semi estructuras, utilizadas comúnmente en las investigaciones sobre ACSC [3, 14, 21], son uno de los principales obstáculos a vencer si se pretende desarrollar aplicaciones que permitan a los e-tutores adquirir las habilidades que manifiestan con deficiencia. Con el propósito de contribuir a la superación del obstáculo antes mencionado, se está desarrollando una línea de investigación orientada a utilizar técnicas de minería de textos para construir clasificadores que identifiquen automáticamente habilidades manifestadas por e-tutores. Los clasificadores que se han logrado construir se describen en este trabajo.

El presente artículo se estructura como sigue. En la próxima sección se detallan las técnicas utilizadas para analizar interacciones en ACSC. En la sección 3 se define la tarea de clasificación de textos y se especifican los pasos metodológicos ejecutados para construir los clasificadores, la experimentación realizada y los resultados obtenidos aplicando diferentes algoritmos de clasificación. Finalmente, en la sección 4 se enuncian algunas conclusiones.

2 Análisis de interacciones en ACSC

En los entornos de ACSC se dispone de un registro completo de las actividades e interacciones, tanto de estudiantes como de docentes. Un análisis posterior de este conjunto de actividades e interacciones permite comprender lo sucedido en una determinada discusión, actividad, o curso. Pero, cuando se tiene una cantidad considerable de interacciones el análisis manual es prácticamente inviable debido al tiempo y esfuerzo que demanda [3, 16]. Por otro lado, en el análisis automático de la información registrada por un sistema de ACSC, se ha recurrido tanto a interfaces estructuradas o semi estructuradas como a texto libre para modelar las interacciones. Las interfaces estructuradas o semi estructuradas (oraciones de apertura, contribuciones dirigidas por menú, y contribuciones basadas en diagramas) permiten simplificar el proceso de análisis al no requerir el uso de técnicas de procesamiento de lenguaje natural [3, 21, 22]. Sin embargo, existen investigaciones que demuestran que este tipo de interfaces socavan el proceso de colaboración porque restringen las posibilidades de interacción, provocan que la comunicación sea lenta, y crean stress relacional [3, 14, 21]. Por su parte, el texto libre permite que la comunicación se realice sin restricciones, pero dificulta el proceso de análisis ya que no es posible usar la propia interfaz para hacer inferencias sobre el proceso de interacción, y por consiguiente es necesario recurrir a técnicas de procesamiento de lenguaje natural.

Dadas las deficiencias asociadas con el análisis manual y con el análisis automático de interacciones estructuradas o semi estructuradas, es necesario encontrar una técnica capaz de detectar las habilidades que poseen los e-tutores sin afectar negativamente la dinámica de trabajo de los estudiantes o los docentes, ni representar un alto costo o sobrecarga de trabajo. En este contexto la minería de textos se muestra como una alternativa viable para cumplir con las condiciones mencionadas debido a su capacidad para manejar la vaguedad, borrosidad, diversidad de estructuras, y gran cantidad de palabras que caracterizan al lenguaje natural [7]. El empleo de técnicas de procesamiento de lenguaje natural no es algo novedoso en el contexto del ACSC, pero hasta el momento se han utilizado de manera limitada en tareas como clasificación de los tópicos de una conversación o caracterización de patrones de discusión [22]. Esto indica que la originalidad del trabajo que se describe en el presente artículo radica en usar minería de textos para construir clasificadores que reconocen las habilidades manifestadas por docentes de ACSC.

3 Los clasificadores desarrollados

La minería de textos se refiere al proceso de extracción de patrones interesantes y no triviales, o conocimiento desde documentos de texto [7]. Si bien la minería de textos utiliza técnicas de minería de datos, en el proceso de descubrimiento de conocimiento la minería de textos parte de datos textuales no estructurados, mientras que la

minería de datos se aplica sobre bases de datos estructuradas [6]. Las técnicas de clasificación de la minería de textos, que consisten en asignar objetos a categorías predefinidas, se adecuan naturalmente al problema de identificar las habilidades manifestadas por los profesores de ACSC, porque la intención es asignar una o más habilidades a cada contribución de los e-tutores.

La clasificación de textos puede definirse como la tarea de aproximar una función de asignación de categoría desconocida $F: D \times C \rightarrow \{0,1\}$, donde D es el conjunto de documentos y C es el conjunto de categorías pre definidas [6]. El valor de $F(d,c)$ es 1 si el documento d pertenece a la categoría c de otra manera el valor es 0. La función de aproximación $M: D \times C \rightarrow \{0,1\}$ se conoce como clasificador o modelo, y el objetivo es construir un clasificador que produzca resultados tan cercanos como sea posible a la función de asignación de categoría F [6, 20]. Cuando un clasificador se construye mediante aprendizaje de máquina un proceso inductivo examina las características de un conjunto de documentos de entrenamiento clasificados por expertos, y a partir de estas características, el proceso inductivo infiere las condiciones que documentos nuevos o no examinados deberían cumplir para ser clasificados bajo una u otra categoría [20]. En tanto que, la comparación de las decisiones de clasificación de los expertos (sobre un conjunto de prueba) con las categorías reconocidas automáticamente permite evaluar la efectividad de los clasificadores automáticos.

Cuatro cuestiones deben abordarse cuando se usa aprendizaje de máquina para construir un clasificador automático [6]. Primero, se tienen que decidir las categorías usadas para catalogar las instancias. Segundo, es preciso contar con un conjunto de entrenamiento para cada categoría. Tercero, se deben decidir las características que representan a las instancias. En cuarto y último lugar, se debe seleccionar el algoritmo de clasificación a usar. Cada uno de los aspectos mencionados se trata en los apartados venideros.

3.1 Categorías utilizadas para catalogar habilidades

Tabla 1: Sub habilidades sociales con sus atributos de habilidad.

Sub Habilidades Sociales	Atributos de Habilidad
Estimular la interacción entre estudiantes	Usar técnicas para estimular la interacción- Mejorar las relaciones- Promover la interacción sin dominar la discusión- Discutir formas de facilitar la interacción- Estimular respuestas mutuas
Estimular la participación	Solicitar comentarios sobre tópicos tratados- Fomentar explicaciones- Asegurar contribuciones igualitarias- Cambiar el tema de discusión- Plantear tópicos o tareas que mejoren el involucramiento- Regresar preguntas- Indagar causas de la participación reducida
Desarrollar, soportar y mantener una comunidad de aprendizaje	Desarrollar la cohesión grupal- Promover la colaboración- Promover el respeto
Crear un ambiente de aprendizaje agradable	Dar la bienvenida- Usar rompe hielos- Fomentar comentarios sobre presentaciones- Proteger contra el miedo al ridículo – usar un estilo amigable e informal- Promover la amistad
Manejar comportamientos de discusión inapropiados	Tratar con mirones- Tratar con no participantes- Tratar con bloqueadores- Tratar con dominadores- Tratar con buscadores de reconocimiento
Manejar contribuciones inapropiadas	Actuar contra la críticas- Actuar contra la intimidación- Actuar contra la discriminación
Alabar y modelar el comportamiento de discusión bueno	Alabar comportamientos de discusión bueno- Modelar comportamientos de discusión bueno
Manejar conflictos	Fomentar el desarrollo de normas para manejar conflictos- Mediar diferencias- Lograr consenso- Excluir a personas conflictivas
Trabajar con sentimientos y sensaciones	Expresar sentimientos y sensaciones – Estimular la expresión de sentimientos y sensaciones- Manejar conflicto afectivo
Motivar	Proveer palabras de aliento y apoyo- Valorar el esfuerzo- Mostrar confianza – Establecer objetivos desafiantes

La falta de una base teórica sólida sobre la manera en que los docentes de ACSC intervienen en las actividades de aprendizaje colaborativo de los estudiantes [14, 15] llevó a que se planteara un esquema de clasificación de las habilidades que deberían poseer los e-tutores para desempeñarse de manera efectiva en entornos de ACSC. Esta

clasificación de habilidades, que se describe de manera detallada en Santana *et al.* [17], está organizada en 3 niveles consecutivos de detalle (habilidades, sub habilidades, y atributos de habilidad). En la taxonomía propuesta existen 6 categorías de habilidades que se corresponden con los roles desempeñados por los e-tutores (Administrativo, Pedagógico, Social, Técnico, Comunicación y Evaluación), 42 sub habilidades y 190 atributos de habilidad. A fin de evaluar la viabilidad del uso de la minería de textos en el reconocimiento automático de las habilidades de docentes de ACSC, en este trabajo se consideraron solo las sub habilidades sociales. El interés en las sub habilidades sociales, vinculadas con la gestión del proceso de interacción entre estudiantes, se debe a que la adquisición de conocimiento en un ambiente de ACSC está influenciada por las interacciones grupales y relaciones interpersonales [14, 15]. La Tabla 1 muestra las sub habilidades sociales con sus correspondientes atributos de la habilidad.

3.2 Conjunto de entrenamiento e instancias

El conjunto de datos (entrenamiento y prueba) usado para construir los clasificadores se obtuvo a partir de diferentes fuentes: 2 experiencias de ACSC donde participaron docentes y estudiantes de la FCEyT de la UNSE, y 2 cursos dictados en el marco del proyecto de investigación Tactics [8]. Teniendo en cuenta que las interacciones por sí mismas no son suficientes para construir un clasificador, ya que es necesario conocer los atributos de habilidad manifestados en cada caso por los e-tutores, sobre el conjunto de interacciones se aplicó análisis de contenido siguiendo el planteamiento metodológico de Krippendorff [9]. De este modo expertos humanos (un psicopedagogo y un especialista en ACSC) hicieron corresponder a cada una de las oraciones o sentencias que componen los mensajes publicados por los docentes con un atributo de habilidad social de la clasificación propuesta por Santana *et al.* [17]. Puesto que solo se considerarían las habilidades sociales, como paso previo al análisis de las interacciones por parte de los expertos, se aplicó la técnica de muestreo de relevancia [9] para seleccionar las interacciones docente-estudiante en las cuales se podrían manifestar habilidades sociales. De este modo, los expertos no tuvieron que catalogar todos los mensajes de los e-tutores (más de 4000 oraciones provenientes de discusiones de chats y foros) sino solamente los pre seleccionados mediante el muestreo de relevancia.

El nivel de acuerdo entre los expertos humanos (para las 891 sentencias resultantes) fue de $\alpha = 0.938$, y si bien este valor se encuentra por arriba del 0.80 recomendado por Krippendorff, indica que existieron oraciones a las que los expertos asignaron atributos de habilidad diferentes. Para resolver estas discrepancias se realizó un segundo proceso de análisis donde los expertos unificaron criterios y acordaron la asignación de un único atributo de habilidad para las oraciones en las que inicialmente existió desacuerdo.

3.3 Selección de características y algoritmo de clasificación a utilizar

Para construir los clasificadores se siguió la metodología CRISP-DM [2]. Si bien CRISP-DM fue propuesta para guiar el desarrollo de proyectos de minería de datos, también puede usarse para abordar problemas de minería de textos en la medida que los datos textuales se transformen en un formato estructurado o semi estructurado [12]. Durante la aplicación de CRISP-DM se recurrió a la herramienta de minería de textos GATE [4] para automatizar las actividades de preparación de datos, construir los clasificadores, y calcular métricas de efectividad. Se decidió utilizar GATE en lugar de otras herramientas *open source* para minería de textos (concretamente Weka/Kea¹, Pimiento², y R/tm³) puesto que: cuenta con facilidades para adaptar sus recursos de procesamiento a las necesidades de cada proyecto, dispone de documentación de usuario más completa y detallada, y permite trabajar con mayor cantidad de formatos de archivo. A continuación se describen las actividades desarrolladas en las primeras 5 fases de la metodología CRISP-DM ya que algunos resultados de la sexta fase (fase de despliegue) se mencionan en las conclusiones.

3.3.1 Fase 1: Comprensión del Negocio

En parte el uso de minería de textos estuvo motivado por la necesidad de reducir la carga de trabajo de las personas interesadas en analizar las interacciones de ACSC. Por lo tanto, en esta fase se determinó que como parte del reconocimiento automático de habilidades los mensajes de los e-tutores debían ser descompuestos en oraciones sin necesidad de intervención del usuario. Téngase en cuenta que se asumió que para identificar las

¹ <http://www.nzdl.org/Kea/>

² <http://erabaki.ehu.es/jjga/pimiento/>

³ <http://tm.r-forge.r-project.org/>

habilidades manifestadas basta con analizar el contenido textual de los mensajes. Esto implica que no se consideraron datos tales como fecha, hora, identidad del emisor, etc., que suelen ser almacenados por entornos de ACSC.

3.3.2 Fase 2: Comprensión de los Datos

El examen de la frecuencia de ocurrencia de los atributos de habilidad en la colección de datos permitió detectar que no se podrían reconocer automáticamente 10 de los 42 atributos de habilidad sociales ya que no se tenía ejemplos que puedan ser utilizados para la construcción de los clasificadores. La frecuencia de los atributos de habilidad también permitió establecer que existía una distribución desbalanceada de los mensajes entre los atributos de habilidad, ya que, se tenía atributos de habilidad que aparecían más de 100 veces pero también atributos cuya cantidad de apariciones no llegaba a 10.

En cuanto al contenido textual de los mensajes existen 3 aspectos a considerar: errores de ortografía (acentuación, omisión de letras, inserción de letras, y unión de palabras), incompatibilidad de caracteres, y archivos duplicados. Las sesiones de ACSC estaban almacenadas en 4 formatos de archivo (htm, rtf, doc, y pdf) y para evitar problemas de compatibilidad en la representación de los caracteres se decidió convertirlos a texto plano utilizando Unicode. Cada archivo de texto plano corresponde a un mensaje o contribución de un e-tutor que puede estar constituido por una o más oraciones. Por archivos duplicados se entiende archivos que coinciden tanto en su contenido textual como en los atributos de habilidad asignados a las oraciones que los componen.

3.3.3 Fase 3: Preparación de los Datos

Los clasificadores se construyen inductivamente a partir de las características de las sentencias clasificadas por expertos humanos, por lo cual es difícil que con pocos ejemplos se puedan inferir los rasgos que distinguen a los atributos de habilidad. Por ello, se excluyeron los atributos de habilidad con menos de 5 instancias, reduciéndose así a 21 los atributos de habilidad social usados en la investigación. Seguidamente para hacer una limpieza de datos se modificaron y eliminaron archivos como se describe a continuación:

- **Modificación de archivos:** en primer lugar se corrigieron las palabras que presentaban errores de ortografía. Luego en los archivos de texto plano, se procedió a insertar los caracteres Unicode equivalentes a los emoticones, representados mediante una imagen en los archivos originales de las sesiones de ACSC. La última tarea de edición del contenido de los archivos consistió en eliminar las oraciones a las que se les había asociado un atributo de habilidad que tenía menos de 5 instancias en el conjunto de datos.
- **Exclusión de archivos:** se eliminaron archivos duplicados, archivos constituidos únicamente por sentencias con atributos de habilidad cuya frecuencia era inferior a 5, y archivos que contenían solo nombres de lugares o de personas. Se decidió excluir los archivos formados solo por nombres de lugares y de personas porque al procesar nuevas interacciones de ACSC lo más probable es que los nombres fueran totalmente distintos. Esto significa que a menos que los nombres estén acompañados de otras palabras, no serán características de ayuda para construir clasificadores que asignen habilidades a contribuciones docentes provenientes de nuevas experiencias de ACSC.

Los documentos de texto son por naturaleza desestructurados pero los sistemas de minería de textos no aplican sus algoritmos de descubrimiento de conocimiento a colecciones de documentos no estructurados, en consecuencia en la minería de textos se pone énfasis en las operaciones de pre procesamiento [6, 23]. Estas operaciones permiten transformar datos no estructurados en un formato intermedio estructurado más explícitamente [6]. Las operaciones de pre procesamiento se centran en la identificación, extracción, refinamiento y adición de características a los documentos en lenguaje natural de modo tal que, las características más representativas sean usadas para la minería y las restantes descartadas [6]. Si se considera que cada característica es una dimensión en un espacio de características, se pueden representar a los documentos por medio de un vector en este espacio. En el presente trabajo de investigación se recurrió a las operaciones de pre procesamiento indicadas a continuación: identificación de token, identificación de sentencias, normalización (*stemming to a root* y lematización), *Part-of-Speech Tagging* (POST), y supresión de *stop words* [6, 7, 23]. Estas operaciones no se aplicaron de manera aislada sino que se combinaron para determinar si, las características que permitían obtener, tenían influencia en la efectividad de los algoritmos de clasificación. La Tabla 2 muestra todas las combinaciones posibles de las operaciones de pre procesamiento aplicadas.

Para identificar las oraciones que forman parte de las contribuciones de los e-tutores se realizaron adaptaciones al recurso de procesamiento *ANNIE Sentence Splitter* [4] porque las reglas incluidas originalmente en GATE funcionan para textos en inglés. En Santana *et al.* [18] se brindan detalles sobre las reglas que se crearon de modo

tal que mediante *ANNIE Sentence Splitter* se puedan identificar oraciones en textos en español. GATE tampoco cuenta con un recurso dedicado exclusivamente a la supresión de *stop words* por lo cual, se tuvo que implementar esta operación de pre procesamiento mediante *JAPE Transducer* y *ANNIE Gazetteer* [4].

Tabla 2: Combinaciones de operaciones de pre procesamiento.

Pre Procesamiento Combinaciones	Identificación de Token	Identificación de Sentencias	Eliminación de Stop Words	Lematización	Stemming to a root	POST
Procesamiento 1	x	x				
Procesamiento 2	x	x	x			
Procesamiento 3	x	x		x		
Procesamiento 4	x	x			x	
Procesamiento 5	x	x				x
Procesamiento 6	x	x	x			x
Procesamiento 7	x	x	x	x		
Procesamiento 8	x	x	x		x	
Procesamiento 9	x	x		x		x
Procesamiento 10	x	x			x	x
Procesamiento 11	x	x	x	x		x
Procesamiento 12	x	x	x		x	x

3.3.4 Fases 4 y 5: Modelado y Evaluación

Para la construcción de los clasificadores se usaron los algoritmos K-ésimo Vecino más Cercano (*K-nearest Neighbor Algorithm- KNN*), Máquinas de Vector de Soporte (*Support Vector Machines- SVM*), Perceptrón con Márgenes Desiguales (*Perceptron Algorithm with Uneven Margins-PAUM*), Naïve Bayes y C4.5 [6, 10, 20, 23]. En vistas de que los algoritmos de clasificación de GATE son binarios (se aplican a dos clases o categorías), no se puede construir un clasificador con la capacidad de asignar uno de los 21 atributos de habilidad social a las oraciones que forman los mensajes de los e-tutores sino que, se deben construir 21 clasificadores binarios. Por otro lado, puesto que los clasificadores construidos con GATE generan como salida un número real comprendido en el rango [0,1], para determinar la categoría correspondiente a una sentencia se verifica si la salida del clasificador es mayor a un valor umbral. En este trabajo se estableció como umbral al valor 0.50.

Para evaluar la efectividad en el reconocimiento de habilidades se calcularon las métricas de precisión, recall, y F_1 mediante validación cruzada 10-fold, ya que este método protege contra resultados influenciados por la composición de los conjuntos de prueba y de entrenamiento [20]. La precisión indica el porcentaje de oraciones clasificadas correctamente entre todas las oraciones a las que se les asignó un atributo de habilidad, independientemente de si el clasificador no asignó atributos de habilidad a oraciones que debieron ser clasificadas [4, 6]. Por su parte, el recall señala el porcentaje de oraciones clasificadas correctamente entre todas las oraciones a las que se les debía asignar un atributo de habilidad, independientemente de cuantas clasificaciones erróneas se hayan realizado [4, 6].

Algoritmo KNN. Los clasificadores basados en el algoritmo KNN no construyen una representación declarativa explícita de las categorías sino que computan directamente la similitud entre el documento a ser clasificado y los documentos de entrenamiento [6]. El entrenamiento para estos clasificadores consiste simplemente en almacenar representaciones de los documentos de entrenamiento junto con sus categorías. Para decidir si el documento d pertenece a la categoría c el algoritmo KNN chequea si los K documentos de entrenamiento más similares a d pertenecen a la categoría c [6]. Si la respuesta es positiva para una proporción suficientemente grande se toma una decisión positiva, caso contrario una decisión negativa.

La Tabla 3 contiene los valores de precisión (π), recall (ρ) y F_1 para el algoritmo KNN cuando se consideran 5, 10 o 15 oraciones del conjunto de entrenamiento para clasificar una sentencia del conjunto de prueba (esta cantidad está indicada por K). Se observa que a medida que se aplica mayor cantidad de operaciones de preparación de datos, es decir, se reduce la variabilidad en el espacio de características lingüísticas a considerar, se mejora la efectividad en la identificación de habilidades. Si se considera al pre procesamiento 1 como una línea base, puede verse que los incrementos de precisión y recall que se logran mediante el pre procesamiento se mantienen ante variaciones del parámetros K . De la Tabla 3 también se desprende que un incremento del valor de

K tiene aparejado una disminución de los valores máximos de precisión y el recall de los clasificadores. Es probable que la disminución de la efectividad de los clasificadores conforme crece K se deba al conjunto de datos usado en la investigación, pues como se mencionó existen atributos de habilidad que no superan las 10 instancias mientras otros llegan a las 100. En este contexto, ante un incremento en el número de instancias de entrenamiento consideradas, es de esperar (para clases con pocas instancias) que las oraciones con mayor grado de similitud a la oración a clasificar, pierdan peso en relación a sentencias que quizás no sean tan similares pero son mayoría. Una última cuestión a resaltar sobre KNN es que los incrementos de precisión y de recall que pueden lograrse mediante la preparación de datos no superan en más del 0.10 a los valores de la línea base ya sea para K=5, K=10 o K=15.

Tabla 3: Micro promedio de medidas de efectividad de clasificadores construidos con KNN.

Parámetros Procesamiento	KNN con K=5			KNN con K=10			KNN con K=15		
	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1
Procesamiento 1	0.29	0.27	0.28	0.28	0.25	0.26	0.23	0.24	0.23
Procesamiento 2	0.31	0.29	0.30	0.28	0.25	0.26	0.26	0.26	0.26
Procesamiento 3	0.33	0.31	0.32	0.32	0.29	0.31	0.30	0.30	0.30
Procesamiento 4	0.34	0.33	0.33	0.31	0.29	0.30	0.28	0.29	0.28
Procesamiento 5	0.30	0.27	0.28	0.30	0.25	0.27	0.24	0.24	0.24
Procesamiento 6	0.30	0.27	0.28	0.32	0.28	0.30	0.29	0.29	0.29
Procesamiento 7	0.33	0.31	0.32	0.35	0.31	0.33	0.32	0.33	0.32
Procesamiento 8	0.34	0.32	0.33	0.32	0.29	0.30	0.28	0.29	0.29
Procesamiento 9	0.34	0.32	0.33	0.32	0.29	0.30	0.29	0.29	0.29
Procesamiento 10	0.34	0.32	0.33	0.32	0.28	0.30	0.27	0.26	0.26
Procesamiento 11	0.36	0.33	0.35	0.35	0.31	0.32	0.32	0.32	0.32
Procesamiento 12	0.37	0.33	0.35	0.36	0.31	0.33	0.30	0.31	0.31

Algoritmo SVM. En términos geométricos un clasificador SVM binario (cuando se tienen solo dos categorías o clases), puede ser visto como un hiperplano en el espacio de características que separa los puntos que representan instancias positivas de la categoría de los puntos que representan instancias negativas [6]. El hiperplano de clasificación se selecciona durante el entrenamiento como el único hiperplano que separa las instancias positivas conocidas de las instancias negativas conocidas con el máximo margen. El margen es la distancia desde el hiperplano al punto más cercano de los conjuntos positivo y negativo. Los hiperplanos de SVM son completamente determinados por un conjunto relativamente pequeño de instancias de entrenamiento conocidas como vectores de soporte.

Para los 21 atributos de habilidad social usados en este estudio la cantidad de ejemplos positivos es bastante pequeña en relación a la cantidad de ejemplos negativos, por consiguiente se decidió recurrir a la versión de SVM con márgenes desiguales [11]. En el algoritmo SVM el margen puede ser considerado como una medida de la habilidad del clasificador para tolerar errores puesto que, un clasificador probablemente clasificará correctamente una instancia de prueba si tiene el margen más grande. Si el conjunto de entrenamiento es representativo del conjunto de datos, un clasificador con el margen más grande con respecto al conjunto de entrenamiento tendrá el mejor desempeño en cuanto a generalización. Sin embargo, si el conjunto de entrenamiento no es representativo el máximo margen aprendido puede llegar a tener pobre performance de generalización tal como se puede apreciar en la Figura 1. Esta figura muestra un problema de clasificación binario con dos dimensiones junto con dos tipos de conjuntos de entrenamiento y sus respectivos clasificadores SVM. En los dos gráficos cada “+” representa una instancia positiva y cada “x” representa una instancia negativa, en tanto que los símbolos “+” y “x” con línea sólida son los que forman parte del conjunto de entrenamiento. Los ejemplos de entrenamiento del gráfico de la izquierda en la Figura 1 son representativos del conjunto de datos por lo cual el máximo margen aprendido tiene buena capacidad de generalización. En cambio, en el gráfico de la derecha el número de ejemplos de entrenamiento positivos es muy pequeño y los ejemplos de entrenamiento se acumulan en una pequeña región del espacio de características. En este caso el clasificador SVM con máximo margen no es una buena predicción puesto que clasifica erróneamente algunos ejemplos positivos. No obstante, si el límite de clasificación fuera alejado apropiadamente de los ejemplos de entrenamiento positivos, el clasificador tendría una mejor performance de generalización. Para distinguir entre el margen positivo y el margen negativo Li y Shawe-taylor [11] proponen introducir el parámetro τ en el algoritmo SVM estándar.

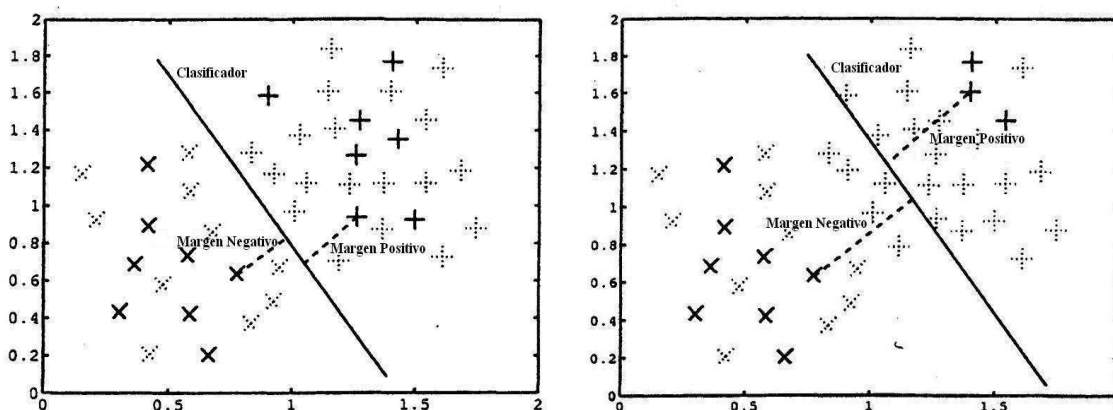


Figura 1. Problema de clasificación bi dimensional con sus clasificadores SVM [11].

Los valores óptimos del parámetro τ para SVM con márgenes desiguales usualmente se encuentran dentro de los siguientes: $\{-0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 2\}$ [11]. No se construyeron modelos con los 15 valores listados ya que al combinarlos con las 12 alternativas de preparación de datos se tendría un total de 180 modelos. Se decidió comenzar con $\tau = 0.5$ para luego incrementar o reducir el valor en función del efecto que la variación producía sobre la efectividad en el reconocimiento. Siguiendo esta estrategia solo se probaron 4 valores de τ (0.3, 0.4, 0.5 y 0.6) puesto que fuera de los mismos la precisión o el recall asumen valores muy bajos. Los resultados de los experimentos (Tabla 4) combinando los 4 valores de τ con las diversas operaciones de pre procesamiento muestran que la preparación de los datos tiene un efecto variable sobre la efectividad del reconocimiento. Por un lado, la eliminación de *stop words* y POST tienen una influencia casi imperceptible sobre precisión y recall. Por el otro lado, la normalización puede ocasionar tanto una disminución como un incremento de la precisión cuando se opta por el *stemming to a root*, pero estas variaciones no superan el valor 0.02. En cambio con la lematización la precisión es siempre creciente y se pueden lograr ganancias del orden del 0.04.

Con relación al valor de τ , a medida que se acerca a uno se produce un incremento del valor de precisión pero una reducción en el recall, en tanto que para valores próximos a cero es el valor de recall el que crece y el de precisión el que disminuye. Por consiguiente, si cuando se dispone de pocos ejemplos de entrenamiento se considera de mayor importancia el reconocer la mayor cantidad de habilidades posibles, conviene utilizar valores de τ próximos al cero. En cambio, si lo más importante es que las decisiones de clasificación sean correctas, conviene utilizar valores de τ cercanos al uno.

Tabla 4: Micro promedio de medidas de efectividad de clasificadores construidos con SVM.

Parámetros / Procesamiento	$\tau=0.3$			$\tau=0.4$			$\tau=0.5$			$\tau=0.6$		
	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1
Procesamiento 1	0.57	0.31	0.40	0.63	0.28	0.38	0.67	0.25	0.36	0.72	0.23	0.35
Procesamiento 2	0.58	0.31	0.40	0.63	0.27	0.38	0.66	0.24	0.35	0.71	0.23	0.34
Procesamiento 3	0.60	0.32	0.41	0.66	0.28	0.39	0.71	0.25	0.37	0.76	0.23	0.36
Procesamiento 4	0.57	0.30	0.39	0.64	0.28	0.38	0.69	0.26	0.37	0.70	0.24	0.36
Procesamiento 5	0.55	0.30	0.39	0.62	0.27	0.38	0.67	0.24	0.35	0.71	0.23	0.34
Procesamiento 6	0.56	0.30	0.39	0.62	0.27	0.37	0.68	0.24	0.35	0.71	0.22	0.34
Procesamiento 7	0.58	0.30	0.40	0.66	0.27	0.38	0.69	0.24	0.36	0.72	0.22	0.34
Procesamiento 8	0.58	0.30	0.39	0.64	0.27	0.38	0.68	0.25	0.36	0.68	0.23	0.35
Procesamiento 9	0.59	0.31	0.41	0.66	0.29	0.40	0.70	0.26	0.37	0.73	0.24	0.36
Procesamiento 10	0.59	0.31	0.41	0.65	0.29	0.40	0.71	0.26	0.38	0.73	0.22	0.34
Procesamiento 11	0.57	0.30	0.39	0.63	0.28	0.38	0.68	0.24	0.36	0.72	0.23	0.35
Procesamiento 12	0.59	0.31	0.40	0.64	0.27	0.38	0.69	0.25	0.36	0.75	0.23	0.35

Algoritmo PAUM. El Algoritmo PAUM es una extensión del algoritmo Perceptrón. El perceptrón es una red neuronal simple formada por una neurona con pesos sinápticos ajustables. En el algoritmo perceptrón el

aprendizaje de la tarea de clasificación se realiza mediante pequeños ajustes a los pesos de manera de reducir la diferencia entre la salida deseada y la obtenida con el perceptrón [6, 23].

Utilizando un margen positivo (τ_{+1}) y un margen negativo (τ_{-1}) el algoritmo PAUM puede hacer frente a problemas de clasificación donde los ejemplos positivos son escasos en relación a los ejemplos negativos. Los valores óptimos de τ_{-1} suelen pertenecer al conjunto $\{-1.5, -1, -0.5, 0, 0.1, 0.5, 1.0\}$ en tanto que los valores óptimos de τ_{+1} por lo general están incluidos en $\{-1, -0.5, 0, 0.1, 0.5, 1, 2, 5, 10, 50\}$ [10]. Teniendo en cuenta que existen 70 combinaciones posibles de los valores de los márgenes, y que se tienen 12 alternativas de pre procesamiento, el número total de modelos sería de 840. Para establecer el valor apropiado de los parámetros no se construyeron todos esos modelos sino que, mediante la validación cruzada 10-fold se combinaron los valores de los 2 márgenes con la primera operación de preparación de datos. De todas estas combinaciones se seleccionaron las que permitían: obtener el valor más alto de precisión, lograr el valor más alto de recall, y tener la menor diferencia entre recall y precisión. Los valores que permitieron cumplir la primera condición son $\tau_{+1}=-1$ y $\tau_{-1}=-1.5$, la segunda condición la satisfacen $\tau_{+1}=10$ y $\tau_{-1}=0.5$, y la tercera condición la cumplen $\tau_{+1}=5$ y $\tau_{-1}=0$ al igual que $\tau_{+1}=10$ y $\tau_{-1}=-0.5$ (Tabla 5). Una vez definidos los valores para τ_{+1} y τ_{-1} se procedió a construir los modelos combinando esos valores con las 12 alternativas de pre procesamiento. Tal como puede observarse en Tabla 5, la reducción del espacio de características no incrementa la precisión sino que tiende a disminuirla, mientras que el recall permanece relativamente estable. No obstante, con el *stemming to a root* y POST en algunos casos puede incrementarse la precisión hasta en 0.03. Los resultados obtenidos permiten destacar dos aspectos sobre el algoritmo PAUM. En primer lugar, para valores positivos de τ_{+1} no existen grandes diferencias entre la precisión y el recall, en tanto que, para valores negativos de τ_{+1} la precisión supera ampliamente al recall. En segundo lugar, los resultados de PAUM parecerían estar dominados por el margen positivo ya que para $\tau_{+1}=10$ se lograron los mismos valores de precisión y recall independientemente de que τ_{-1} sea igual a 0.5 o igual a -0.5.

Tabla 5: Micro promedio de medidas de efectividad de clasificadores construidos con PAUM.

Parámetros Procesamiento	$\tau_{+1}=-1$ $\tau_{-1}=-1.5$			$\tau_{+1}=10$ $\tau_{-1}=-0.5$			$\tau_{+1}=5$ $\tau_{-1}=0$			$\tau_{+1}=10$ $\tau_{-1}=0.5$		
	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1
Procesamiento 1	0.88	0.08	0.15	0.41	0.42	0.41	0.43	0.43	0.43	0.41	0.42	0.41
Procesamiento 2	0.86	0.08	0.16	0.40	0.41	0.40	0.41	0.41	0.41	0.40	0.41	0.40
Procesamiento 3	0.85	0.10	0.18	0.41	0.41	0.41	0.41	0.40	0.40	0.41	0.41	0.41
Procesamiento 4	0.86	0.06	0.12	0.44	0.43	0.44	0.42	0.41	0.41	0.44	0.43	0.44
Procesamiento 5	0.84	0.09	0.17	0.42	0.42	0.42	0.42	0.41	0.41	0.42	0.42	0.42
Procesamiento 6	0.85	0.09	0.16	0.42	0.42	0.42	0.41	0.40	0.40	0.42	0.42	0.42
Procesamiento 7	0.81	0.10	0.18	0.42	0.42	0.42	0.40	0.40	0.40	0.42	0.42	0.42
Procesamiento 8	0.86	0.07	0.13	0.43	0.44	0.43	0.42	0.40	0.41	0.43	0.44	0.43
Procesamiento 9	0.84	0.09	0.16	0.42	0.42	0.42	0.41	0.41	0.41	0.42	0.42	0.42
Procesamiento 10	0.87	0.09	0.16	0.44	0.44	0.44	0.44	0.42	0.43	0.44	0.44	0.44
Procesamiento 11	0.83	0.10	0.18	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
Procesamiento 12	0.86	0.09	0.16	0.44	0.43	0.43	0.43	0.42	0.42	0.44	0.43	0.43

Algoritmos Naïve Bayes y C4.5. Al aplicar los algoritmos Naïve Bayes y C4.5 se obtuvo un valor de precisión igual a 1 en todos los casos. Esto parecería indicar que no se cometieron errores al clasificar las oraciones, pero, dado que recall se mantuvo en 0 también en todos los casos, lo que realmente sucedió es que no se cometieron errores porque no se tomaron decisiones de clasificación (no se reconocieron habilidades). Por consiguiente, ni Naïve Bayes ni C4.5 parecerían ser apropiados para el problema de identificación automática de habilidades de un e-tutor. En el caso de Naïve Bayes es de suponer que los resultados estuvieron influenciados por el conjunto de datos usados, puesto que al ser un clasificador probabilístico su desempeño depende de la cantidad de instancias empleadas para estimar las probabilidades. Para C4.5 es probable que las características lingüísticas usadas en la representación de las interacciones no permitan discriminar entre habilidades.

4 Conclusiones

Con esta investigación se comprobó que las técnicas de clasificación de minería de textos, aplicadas sobre los mensajes publicados por los e-tutores en entornos de ACSC con interfaces no estructuras, permiten reconocer las habilidades manifestadas por los mismos. A priori se creía que las operaciones de pre procesamiento permitirían

mejorar la efectividad de los clasificadores automáticos pero, la experimentación realizada demostró que el efecto del pre procesamiento depende del algoritmo de aprendizaje. Así, para KNN el pre procesamiento puede incrementar la precisión y el recall hasta en un décimo, mientras que su influencia sobre SVM y PAUM es reducida. Esto pareciera indicar que tanto SVM como PAUM no son sensibles al tamaño del espacio de características.

Dado que SVM con márgenes desiguales y PAUM han sido pensados para problemas de clasificación con pocos ejemplos de entrenamiento positivos, es lógico que se hayan obtenido niveles de precisión que dupliquen a la máxima precisión lograda con KNN. No obstante, se debería seguir trabajando para mejorar los niveles de recall de los clasificadores pero sin que esto implique sacrificar precisión. En este sentido, un posible curso de acción consiste en ajustar los parámetros de los algoritmos a la cantidad de ejemplos disponibles de cada atributo de habilidad [10]. Si bien, se necesita seguir trabajando para mejorar el recall, los valores de precisión logrados son más que destacables si se los compara con las cifras de precisión reportadas en otros trabajos de investigación donde se utiliza minería de textos para tareas de clasificación sobre una colección de documentos considerablemente más grande. Así por ejemplo, en [10] con la colección Mod-Apte (una muestra de Reuters-21578 con 12902 documentos) se reportaron valores de precisión de 0.75, mientras que en la presente investigación se lograron niveles de precisión mayores al 0.7 con apenas 891 oraciones.

Para obtener resultados verdaderamente concluyentes, en trabajos futuros se cree conveniente incrementar el número de interacciones de entrenamiento y prueba, y también ampliar las categorías a reconocer ya que en esta investigación el reconocimiento se restringió a los atributos de la habilidad social. No obstante, los resultados obtenidos son alentadores sobre el uso de la minería de textos como parte de un sistema de software de entrenamiento personalizado de habilidades para e-tutores de ACSC. Una aplicación de este tipo tendría que analizar las interacciones grupales de modo de identificar conflictos que requieran la intervención de los e-tutores para resolverse. En estos casos, si los e-tutores no realizaron la o las intervenciones adecuadas, el sistema se encargaría de sugerirles las acciones a llevar a cabo de modo tal que mejoren el aprendizaje grupal y simultáneamente practiquen las habilidades que no hayan manifestado adecuadamente.

Referencias

- [1] Philip Barker. Skill set for online teaching. Proc. de *Word Conference on educational, Multimedia, Hipermedia, & Telecomunicatios (EDMEDIA'02)*, pp. 85-90, 2002.
- [2] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, y Rüdiger Wirth. *CRISP-DM 1.0 Step-by-step data mining guide*. SPSS Inc, USA, 2001.
- [3] W. Chen. Supporting teachers' intervention in collaborative Knowledge building. *Journal of Network and Computer Applications*, 29: 200–215, 2006. doi:[10.1016/j.jnca.2005.01.001](https://doi.org/10.1016/j.jnca.2005.01.001)
- [4] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, et al. *Developing Language Processing Components with GATE Version 7 (a User Guide)*. Department of Computer Science, The University of Sheffield, Reino Unido, 2013.
- [5] Tony W. Day. Online Collaborative Learning and Leadership Development. En: Patricia Rogers, Gary Berg, Judith Boettcher, Carole Howard, Lorraine Justice, y Karen Schenk (eds) *Encyclopedia of Distance Learning. Second Edition*. Information Science Reference, pp. 1488 -1492, 2009.
- [6] Ronen Feldman y James Sanger. *The text mining handbook. Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007.
- [7] A. Hotho , A. Nürnberger, y G. Paaß. Brief survey of text mining. *LDV Forum GLDV Journal for Computational Linguistics and Language Technology*, 20 (1): 19-62, 2005.
- [8] M. Juárez Pacheco. Recomendaciones para el uso académico de herramientas web gratuitas. *Revista Mexicana de Investigación Educativa*, 10 (25): 577-584, 2005.
- [9] Klaus Krippendorff. *Content analysis: an introduction to its methodology. Second Edition*. SAGE Publications, USA, 2004.
- [10] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, y Jaz Kandola. The Perceptron Algorithm with Uneven Margins. En Proc. de *Nineteenth International Conf. on Machine Learning*, pp. 379-386, 2002.
- [11] Yaoyong Li y John Shawe-Taylor. The SVM with uneven margins and Chinese document categorization. En Proc. de *17th Pacific Asia Conference on Language Information and Computation (PACLIC17)*, pp. 216- 227, 2003.

- [12] Edilberto Magalhães Silva. *Descubierta de Conhecimento com o uso de Text Mining: Cruzando o Abismo de Moore*. No publicado, Tesis de Master, Universidad Católica de Brasil, 2002.
- [13] M. McPherson y M. Baptista Nunes. The role of tutors as an integral part of online learning support. *European Journal of Open, Distance and E-learning*, 2004.
- [14] Orlando J. Olivares. Collaborative vs. Cooperative Learning: The Instructor's Role in Computer Supported Collaborative Learning. En: Kara L. Orvis y Andrea L.R. Lassiter (eds) *Computer-Supported Collaborative Learning: Best Practices and Principles for Instructors*. Information Science Publishing, pp. 20-39, 2007.
- [15] Kara L. Orvis y Andrea R.L. Lassiter. Computer-Supported Collaborative Learning: The Role of the Instructor. En: Sharmila Pixy Ferris y Susan H. Godar (eds) *Teaching and learning with virtual teams*. Information Science Publishing, pp. 158-179, 2006.
- [16] C. Rosé, Y. Wang, Y. Cui, J. Arguello, K. Stegmann, A. Weinberger, y F. Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning*, 3 (3): 237–271, 2008. doi: [10.1007/s11412-007-9034-0](https://doi.org/10.1007/s11412-007-9034-0)
- [17] Pablo Santana Mansilla, Rosanna Costaguta, y Daniela Missio. Habilidades de E-tutores en Grupos Colaborativos. En: Noelia Peñaranda, Susana Zazarini, y Ignacio F. Bejarano (eds) *Experiencias Innovadoras en Investigación Aplicada*. Ediciones DASS-UCSE, Jujuy, pp. 687-704, 2012.
- [18] Pablo Santana Mansilla, Rosanna Costaguta y Daniela Missio. Uso de la Arquitectura GATE para la Identificación de Sentencias en Textos en Español. *Primer Congreso Argentino de la Interacción-Persona Computador@, Telecomunicaciones, Informática e Información Científica*, 2012.
- [19] B.B. Schwarz y C.S. Asterhan. E-Moderation of Synchronous Discussions in Educational Settings: A Nascent Practice. *Journal of the Learning Sciences*, 20 (3): 395–442, 2011. doi: [10.1080/10508406.2011.553257](https://doi.org/10.1080/10508406.2011.553257)
- [20] F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1): 1–47, 2002. doi: [10.1145/505282.505283](https://doi.org/10.1145/505282.505283)
- [21] A. Soller, A. Martínez Monés, P. Jermann, y M. Muehlenbrock. From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *International Journal of Artificial Intelligence in Education*, 15 (4): 261-290, 2005.
- [22] Pierre Tchounikine, Nikol Rummel, y Bruce M. McLaren. Computer Supported Collaborative Learning and Intelligent Tutoring Systems. En: Roger Nkambou, Jacqueline Bourdeau y Riichiro Mizoguchi (eds) *Advances in Intelligent Tutoring Systems*. Springer, pp. 447-463, 2010. doi: [10.1007/978-3-642-14363-2_22](https://doi.org/10.1007/978-3-642-14363-2_22)
- [23] Sholom M. Weiss, Nitin Indurkha, Tong Zhang, y Fred J. Damerau. *Text Mining Predictive Methods for Analyzing Unstructured Information*. Springer, USA, 2005.