

## Reconocimiento de habilidades de docentes en Aprendizaje Colaborativo Soportado por Computadora mediante minería de textos

Pablo Santana Mansilla<sup>(1,2)</sup>, Rosanna Costaguta<sup>(2)</sup> y Daniela Missio<sup>(2)</sup>

(1) CONICET, Comisión Nacional de Investigaciones Científicas y Técnicas

(2) Instituto de Investigación en Informática y Sistemas de Información (IISI),  
Departamento de Informática, Facultad de Ciencias Exactas y Tecnologías,  
Universidad Nacional de Santiago del Estero (UNSE),  
Avda. Belgrano (S) 1912, Santiago del Estero, 4200, Argentina  
{psantana, rosanna, dmissio}@unse.edu.ar

**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 principalmente 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.

**Palabras clave:** Aprendizaje Colaborativo Soportado por Computadora, habilidades de e-tutor, minería de textos, identificación automática de habilidades.

### 1 Introducción

Los sistemas de ACSC no solo permiten que el aprendizaje grupal tenga lugar con independencia del tiempo y espacio donde estén localizados los miembros del grupo, sino que brindan un entorno apropiado para el desarrollo de habilidades de solución de problemas y de liderazgo [4]. 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 [14, 20]. 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 [14, 17, 20]. No obstante, la

mayoría de los e-tutores no tienen experiencia con las técnicas propias del ACSC, y además existe la falsa creencia de que 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 [13, 14].

Dado que los e-tutores de ACSC son los principales responsables de la entrega de los cursos y el soporte de los estudiantes, la selección de docentes con las habilidades apropiadas es esencial para el éxito del ACSC [12, 14]. 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 modo que las experiencias de ACSC que emprendan no conduzcan al fracaso.

Si bien en todo sistema de ACSC es posible acceder a un registro completo de las actividades desarrolladas, tanto por estudiantes como por docentes, el análisis manual de esta información es un proceso tedioso, y costoso en tiempo y esfuerzo. Debido a esto, una de las líneas de trabajo del proyecto de investigación “Sistemas de información web basados en agentes para promover el Aprendizaje Colaborativo Soportado por Computadoras” (CICYT-UNSE Código 23/C097), consistió en utilizar técnicas de minería de textos para construir clasificadores que identifiquen automáticamente habilidades manifestadas por e-tutores. Tales clasificadores 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. La sección 3 define la tarea de clasificación de textos y especifica 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 [2,15]. 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 tener que usar técnicas de procesamiento de lenguaje natural [2, 19, 20]. 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 [2, 13, 19]. Por su parte, el texto libre permite que la comunicación se realice sin restricciones pero ya no es posible usar la propia interfaz para hacer inferencias sobre el proceso de interac-

ción, 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 necesaria encontrar una técnica capaz de detectar las habilidades que poseen los e-tutores sin afectar negativamente la dinámica de trabajo de los estudiantes o 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, dada su capacidad para manejar la vaguedad, borrosidad, diversidad de estructuras, y gran cantidad de palabras que caracterizan al lenguaje natural [6]. 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 [20]. Lo que sí es novedoso y destaca la originalidad del trabajo descrito en este documento es el uso de minería de textos para construir clasificadores que reconocen las habilidades manifestadas por los 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 [6]. 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 [5]. 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 [5]. 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  $F$  [18, 5]. Cuando un clasificador se construye mediante aprendizaje de máquina un proceso inductivo examina las características de un conjunto de documentos 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 [18].

Cuatro cuestiones deben abordarse cuando se usa aprendizaje de máquina para desarrollar un clasificador automático [5]. 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

La falta de conocimiento sobre la manera en que los docentes de ACSC intervienen en las actividades de aprendizaje colaborativo de los estudiantes [13, 14] llevó a que se planteara un esquema de clasificación de las habilidades que deberían poseer los tutores para desempeñarse de manera efectiva en entornos de ACSC. Esta clasificación de habilidades, que se describe de manera detallada en Santana *et al.* [16], 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. En esta investigación, para evaluar la viabilidad del uso de la minería de textos en el reconocimiento automático de las habilidades de docentes de ACSC, se trabajó solo con 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 [13, 14]. La Tabla 1 muestra las sub habilidades y atributos de la habilidad social.

**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 las 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 buenos.
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.

### 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 [7]. 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 [8]. 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. [16]. El nivel de acuerdo entre los expertos humanos (para las 891 sentencias) fue de  $\alpha = 0.938$ , y si bien este valor se encuentra por arriba del 0.80 recomendado por Krippendorff, indica que existían oraciones a las que se les 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 que inicialmente existía desacuerdo.

### 3.3 Selección de características y algoritmo de clasificación a utilizar

Para construir los clasificadores se siguió la metodología CRISP-DM [1]. 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 [11]. Durante la aplicación de CRISP-DM se recurrió a la herramienta de minería de textos GATE [3] 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/Kea<sup>1</sup>, Pimiento<sup>2</sup>, y R/tm<sup>3</sup>) 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 o fase de despliegue se mencionan en las conclusiones.

#### **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

---

<sup>1</sup> <http://www.nzdl.org/Kea/>

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

<sup>3</sup> <http://tm.r-forge.r-project.org/>

de habilidades los mensajes de los e-tutores debían ser descompuestos en oraciones sin necesidad de intervención del usuario.

### **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 para representar los caracteres.

### **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 se aplicaron las tareas de limpieza de datos indicadas a continuación: corrección de las palabras con errores de ortografía, en los archivos de texto plano se insertaron los caracteres Unicode equivalentes a los emoticones incluidos en los archivos originales de los diálogos, eliminación de sentencias asociadas con atributos de habilidad que presentaban menos de 5 instancias en el conjunto de datos, supresión de archivos duplicados, y eliminación de archivos que contenían solo nombres de lugares o de personas por cuanto no brindan información relevante para la clasificación.

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 [5, 21]. Estas operaciones permiten transformar datos no estructurados en un formato intermedio estructurado más explícitamente [5]. 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* [5, 6, 21]. Estas operaciones se combinaron para determinar si tenían influencia en la efectividad de los algoritmos de clasificación. La Tabla 2 muestra todas las combinaciones posibles de las 5 operaciones de pre procesamiento aplicadas.

**Tabla 2.** Combinaciones de operaciones de pre procesamiento

Operaciones de pre procesamiento Combinaciones	Identificación de Token	Identificación de Sentencias	Eliminación de <i>Stop words</i>	Lematización	<i>Stemming to a root</i>	POST
Pre procesamiento 1	x	x				
Pre procesamiento 2	x	x	x			
Pre procesamiento 3 A	x	x		x		
Pre procesamiento 3 B	x	x			x	
Pre procesamiento 4	x	x				x
Pre procesamiento 5	x	x	x			x
Pre procesamiento 6 A	x	x	x	x		
Pre procesamiento 6 B	x	x	x		x	
Pre procesamiento 7 A	x	x		x		x
Pre procesamiento 7B	x	x			x	x
Pre procesamiento 8 A	x	x	x	x		x
Pre procesamiento 8B	x	x	x		x	x

Para identificar las oraciones que forman parte de las contribuciones de los e-tutores se realizaron adaptaciones a *ANNIE Sentence Splitter* [3] porque las reglas incluidas originalmente en GATE funcionan para textos en inglés. GATE tampoco cuenta con un recurso dedicado exclusivamente a la supresión de *stop words* pero, esta operación de pre procesamiento se implementó mediante *JAPE Transducer* y *ANNIE Gazetteer* [3].

#### 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 [5, 9, 18, 21]. 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 usó un umbral igual a 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 [18].

#### Algoritmo KNN.

La Tabla 3 contiene los valores de precisión ( $\pi$ ), recall ( $\rho$ ) 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 ob-

serva 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.

**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		
	$\pi$	$\rho$	F <sub>1</sub>	$\pi$	$\rho$	F <sub>1</sub>	$\pi$	$\rho$	F <sub>1</sub>
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 A	0.33	0.31	0.32	0.32	0.29	0.31	0.30	0.30	0.30
Procesamiento 3 B	0.34	0.33	0.33	0.31	0.29	0.30	0.28	0.29	0.28
Procesamiento 4	0.30	0.27	0.28	0.30	0.25	0.27	0.24	0.24	0.24
Procesamiento 5	0.30	0.27	0.28	0.32	0.28	0.30	0.29	0.29	0.29
Procesamiento 6 A	0.33	0.31	0.32	0.35	0.31	0.33	0.32	0.33	0.32
Procesamiento 6 B	0.34	0.32	0.33	0.32	0.29	0.30	0.28	0.29	0.29
Procesamiento 7 A	0.34	0.32	0.33	0.32	0.29	0.30	0.29	0.29	0.29
Procesamiento 7 B	0.34	0.32	0.33	0.32	0.28	0.30	0.27	0.26	0.26
Procesamiento 8 A	0.36	0.33	0.35	0.35	0.31	0.32	0.32	0.32	0.32
Procesamiento 8 B	0.37	0.33	0.35	0.36	0.31	0.33	0.30	0.31	0.31

#### Algoritmo SVM.

Para construir un clasificador mediante el algoritmo SVM es preciso que el problema sea linealmente separable, es decir, que pueda encontrarse un hiperplano que separe las instancias negativas de las positivas. 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 [10]. Este algoritmo se vale de un parámetro  $\tau$  cuyos valores óptimos usualmente se encuentran dentro de los siguientes:  $\{-0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 2\}$  [10]. No se construyeron modelos con los 15 valores listados ya que al combinarlos con las 8 alternativas de preparación de datos se tendría un total de 120 modelos. Lo que se hizo fue 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  $\tau$  (0.3, 0.4, 0.5 y 0.6) puesto que fuera de los mismos la preci-



sión o el recall asumen valores muy bajos. Los resultados de los experimentos (Tabla 4) combinando los 4 valores de  $\tau$  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 su parte, 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.

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

Parámetros	$\tau=0.3$			$\tau=0.4$			$\tau=0.5$			$\tau=0.6$		
	$\pi$	$\rho$	F <sub>1</sub>	$\pi$	$\rho$	F <sub>1</sub>	$\pi$	$\rho$	F <sub>1</sub>	$\pi$	$\rho$	F <sub>1</sub>
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 A	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 3 B	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 4	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 5	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 6 A	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 6 B	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 7 A	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 7 B	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 8 A	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 8 B	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.

Utilizando un margen positivo ( $\tau_{+1}$ ) y un margen negativo ( $\tau_{-1}$ ) 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  $\tau_{-1}$  suelen pertenecer al conjunto  $\{-1.5, -1, -0.5, 0, 0.1, 0.5, 1.0\}$  en tanto que los valores óptimos de  $\tau_{+1}$  por lo general están incluidos en  $\{-1, -0.5, 0, 0.1, 0.5, 1, 2, 5, 10, 50\}$  [9]. Teniendo en cuenta que existen 70 combinaciones posibles de los valores márgenes, y que se tienen 8 alternativas de pre procesamiento, el número total de modelos sería de 560. 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  $\tau_{+1}$  y  $\tau_{-1}$  se procedió a construir los modelos combinando esos valores con las 8 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  $\tau_{+1}$  no existen grandes diferencias entre la precisión y el recall, en tanto que, para valores negativos de  $\tau_{+1}$  la precisión duplica 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  $\tau_{-1}$  sea igual a 0.5 o igual a -0.5.

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

Parámetros	$\tau_{+1}=1$ $\tau_{-1}=-1.5$			$\tau_{+1}=10$ $\tau_{-1}=-0.5$			$\tau_{+1}=5$ $\tau_{-1}=0$			$\tau_{+1}=10$ $\tau_{-1}=0.5$		
	$\pi$	$\rho$	F1	$\pi$	$\rho$	F1	$\pi$	$\rho$	F1	$\pi$	$\rho$	F1
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 A	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 3 B	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 4	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 5	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 6 A	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 6 B	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 7 A	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 7 B	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 8 A	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 8 B	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 cometen errores al clasificar las oraciones, pero, dado que recall se mantuvo en 0 también en todos los casos, lo que realmente sucede es que no se cometen errores porque no se reconocen 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 permiten reconocer las habilidades manifestadas por e-tutores, cuando se aplican a los mensajes publicados por los mismos en entornos de ACSC con interfaces no estructuras. Inicialmente se creía que las operaciones de pre procesamiento permitirían mejorar la efectividad de los clasificadores automáticos pero, la experimenta-

ción 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 indica 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 [9].

Para obtener resultados verdaderamente concluyentes, 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, el sistema se encargaría de sugerir a los e-tutores las acciones a llevar a cabo para mejorar el aprendizaje grupal y así, simultáneamente, instarlos a practicar las habilidades que no hayan manifestado adecuadamente.

## Referencias

1. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: *CRISP-DM 1.0 Step-by-step data mining guide*. SPSS Inc, USA (2001)
2. Chen, W.: Supporting teachers' intervention in collaborative Knowledge building. *Journal of Network and Computer Applications*, 29, 200-215 (2006)
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., et al.: *Developing Language Processing Components with GATE Version 7 (a User Guide)*. Departamento de Ciencias de la Computación, Universidad de Sheffield, Reino Unido (2013)
4. Day, T.W.: Online Collaborative Learning and Leadership Development. En: Rogers, P., Berg, G., Boettcher, J., Howard, C., Justice, L., Schenk, K. (eds) *Encyclopedia of Distance Learning, 2nd Ed*, pp. 1488 -1492. Information Science Reference (2009)
5. Feldman, R., Sanger, J.: *The text mining handbook. Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press (2007)
6. Hotho, A., Nümberger, A., Paaß, G.: Brief survey of text mining. *LDV Forum GLDV Journal for Computational Linguistics and Language Technology*, 20 (1), 19-62 (2005)
7. Juárez Pacheco, M.: Recomendaciones para el uso académico de herramientas web gratuitas. *Revista Mexicana de Investigación Educativa*, 10 (25), pp.577-584 (2005)
8. Krippendorff, K.: *Content analysis: an introduction to its methodology, 2da edición*. SAGE Publications, USA (2004)
9. Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., Kandola, J.: The Perceptron Algorithm with Uneven Margins. En: *Nineteenth International Conf. on Machine Learning*, pp. 379-386 (2002)

10. Li, Y., Shawe-Taylor, J.: The SVM with uneven margins and Chinese document categorization. En: *17th Pacific Asia Conference on Language Information and Computation (PACLIC17)*, pp. 216- 227. Singapur (2003)
11. Magalhães, S. E.: *Descoberta de Conhecimento com o uso de Text Mining: Cruzando o Abismo de Moore*. No publicado, Tesis de Master, Universidad Católica de Brasil (2002)
12. McPherson, M., Baptista, N. M.: The role of tutors as an integral part of online learning support. *European Journal of Open, Distance and E-learning* (2004)
13. Olivares, O. J.: Collaborative vs. Cooperative Learning: The Instructor's Role in Computer Supported Collaborative Learning. En: Orvis, K. L., Lassiter, A. L.R (eds) *Computer-Supported Collaborative Learning: Best Practices and Principles for Instructors*. Information Science Publishing, USA, pp. 20-39 (2007)
14. Orvis, K.L., Lassiter, A.L.R.: Computer-Supported Collaborative Learning: The Role of the Instructor. En: Ferris, S., Godar, S.H. (eds) *Teaching and learning with virtual teams*, pp. 158-179. Information Science Publishing (2006)
15. Rosé, C., Wang, Y., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., Fischer, F.: 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)
16. Santana Mansilla, P., Costaguta, R., Missio, D.: Habilidades de E-tutores en Grupos Colaborativos. En: Peñaranda, N., Zazarini, S., Bejarano, I. F. (eds) *Experiencias Innovadoras en Investigación Aplicada*. Ediciones DASS-UCSE, Jujuy, pp. 687-704 (2012)
17. Schwarz, B.B., Asterhan, C.S.: E-Moderation of Synchronous Discussions in Educational Settings: A Nascent Practice. *Journal of the Learning Sciences*, 20 (3), 395-442 (2011)
18. Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1), 1–47, (2002)
19. Soller, A., Martínez, M. A., Jermann, P., Muehlenbrock, M.: 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)
20. Tchounikine, P., Rummel, N., McLaren, B.M.: Computer Supported Collaborative Learning and Intelligent Tutoring Systems. En: *Advances in Intelligent Tutoring Systems*. Springer, pp. 447-463 (2010)
21. Weiss, S. M., Indurkha, N., Zhang, T., Damerou, F. J.: *Text Mining Predictive Methods for Analyzing Unstructured Information*. Springer, USA (2005)