Clustering Iterativo de Textos Cortos con Representaciones basadas en Conceptos*

Diego Ingaramo, María V. Rosas, Marcelo Errecalde, Paolo Rosso

LIDIC, UNSL, San Luis, Argentina - NLE Lab., ELiRF, UPV, España
{daingara,mvrosas,merreca}@unsl.edu.ar, prosso@dsic.upv.es
http://www.unsl.edu.ar,http://users.dsic.upv.es/grupos/nle/

Resumen La tendencia actual a trabajar con *documentos cortos* (blogs, mensajes de textos, y otros), ha generado un interés creciente en las técnicas de procesamiento automáticas de documentos con estas características. En este contexto, el "*clustering*" (agrupamiento) de *textos cortos* es un área muy importante de investigación, que puede jugar un rol fundamental en organizar estos grandes volúmenes de textos cortos, en un número pequeño de grupos significativos. Recientemente, el uso de métodos de clustering bio-inspirados iterativos, ha producido resultados muy interesantes utilizando representaciones de vector de términos clásicas. En este trabajo, extendemos este enfoque utilizando representaciones de documentos enriquecidas con información semántica (conceptos) obtenida con métodos de desambiguación basados en conocimiento. Los resultados experimentales, permiten concluir que el enfoque de clustering iterativo utilizado puede verse beneficiado significativamente con la incorporación de información semántica en la representación de documentos, mostrando un desempeño superior al exhibido por varios de los métodos de clustering más difundidos en el área, en la mayoría de las instancias experimentales.

1. Introducción

Hoy en día, la enorme cantidad de información disponible en la Web ofrece un número ilimitado de oportunidades para utilizar esta información en diferentes aplicaciones del mundo real. Desafortunadamente, las herramientas de análisis automático que son requeridas para que esta información sea fácilmente interpretada y utilizada de manera efectiva (categorización, clustering y sistemas de extracción de información, entre otras), deben enfrentar muchas dificultades propias de las características de los documentos a ser procesados. Por ejemplo, la mayoría de los documentos Web como por ejemplo "blogs", "snippets", "chats", "FAQs", evaluaciones en línea de productos comerciales, "e-mails", noticias, resúmenes científicos y otros son "*textos cortos*". Éste es un aspecto central si consideramos los problemas que los documentos cortos usualmente plantean a las diferentes tareas de Procesamiento del Lenguaje Natural (PLN) [1,2].

* El trabajo del tercer y cuarto autor ha sido soportado por el proyecto MICINN TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 (Plan I+D+i).

En los últimos tiempos, varios trabajos han reconocido la importancia y complejidad del procesamiento de textos cortos, reportándose resultados interesantes en diversas tareas de *clustering* de este tipo de documentos [1,2,3,4,5,6,7]. En particular, un enfoque iterativo llamado ITSA*, ha demostrado ser muy efectivo en este tipo de tareas y ha logrado excelentes resultados con distintas colecciones de textos cortos [8].

Por otro lado, diferentes trabajos han analizado las ventajas de incorporar información semántica a la representación de textos, obteniéndose resultados positivos en algunos casos [9]. Es conveniente destacar, que en general estos estudios están enfocados en documentos donde es factible, en la mayoría de los casos, disponer de una colección de entrenamiento para la tarea de desambiguación del sentido de las palabras (*WSD*, las siglas en inglés para *Word Sense Disambiguation*). Este enfoque basado en corpus no siempre es factible de ser aplicado en dominios con las características planteadas previamente. Una alternativa para abordar el problema anterior, es el uso de métodos de WSD *basados en conocimiento* los cuales obtienen información desde recursos léxicos externos. Si bien este tipo de métodos suelen mostrar resultados de menor calidad que los obtenidos con métodos basados en corpus, constituyen en muchos casos la única alternativa realista si se desea hacer uso de información semántica en la representación de documentos [10]. Teniendo en cuenta esto, se puede pensar en el enfoque basado en conocimiento como una opción apropiada para el caso que nos ocupa: el clustering de textos cortos.

El principal objetivo de este trabajo es analizar, mediante un estudio experimental, cuál es el impacto que tiene en el desempeño del método iterativo ITSA* en el clustering de textos cortos, cuando se utiliza información semántica (conceptos) obtenida mediante técnicas de WSD basadas en conocimiento. A tal fin, el estudio experimental incluirá algunos de los algoritmos de clustering que han mostrado ser los más efectivos en este área, y un conjunto representativo de colecciones de documentos cortos.

El resto del trabajo está organizado de la siguiente manera: la Sección 2 presenta conceptos introductorios relacionados a nuestro trabajo; la Sección 3 detalla el diseño experimental, describiendo los conjuntos de datos utilizados y la representación de los textos con las distintas variantes en la incorporación de información semántica. También se incluyen aquí, los resultados obtenidos en el trabajo experimental. Finalmente, la Sección 4 presenta las conclusiones y posibles trabajos futuros.

2. Conceptos introductorios

El principal aporte de este trabajo es el análisis del impacto del uso de información semántica (conceptos) obtenida mediante métodos de WSD basados en conocimiento, cuando se usa el método iterativo ITSA*. Por este motivo, y a los fines de hacer este trabajo tan autocontenido como sea posible, se presentan en las siguientes dos subsecciones una breve introducción a cada uno de estos temas.

2.1. Representaciones basadas en Conceptos

En la mayoría de las tareas de categorización, clustering[1] y recuperación de la información, los documentos son representados mediante el modelo de espacio vector introducido por Salton [11], para la codificación de textos. En este enfoque, cada texto es representado por un vector de n-términos, donde n es el número de términos que aparecen en la colección de documentos, y cada término del vector es ponderado con un peso determinado usualmente en base a la frecuencia de ocurrencia del término en el documento y en la colección completa. En el sistema SMART [12], cada codificación está compuesta por tres letras: las primeras dos letras refieren, respectivamente, a TF (frecuencia de un término) e IDF (frecuencia inversa del documento) mientras que el tercer componente ($NORM$) indica si se utiliza normalización o no. Teniendo en cuenta la nomenclatura estándar SMART, se consideran cinco alternativas diferentes para la componente TF: n (natural), b (binario), l (logaritmo), m (max-norm) y a (promedio-norm); dos alternativas para el componente IDF (n y t) con n (no aplicación) y t ($tfidf$) y dos alternativas para la normalización: n (no normalización) y c (coseno). De esta forma, una codificación *ntc* representa la codificación estándar tf-idf (normalizada).

El uso de información semántica implica, en este contexto, la incorporación del *significado* de los términos a la representación. La determinación de cuál es el significado que corresponde a los distintos términos no es una tarea directa debido a los problemas de polisemia y sinonimia. Por este motivo, se requieren de métodos de WSD que, así como se explicó previamente, pueden ser clasificados en términos generales como basados en corpus o basados en conocimiento [13]. En este trabajo, nos centraremos en métodos basados en conocimiento los cuáles requieren de algún recurso externo que, en primera instancia, puede ser cualquier base de conocimiento léxica que defina los diferentes sentidos de las palabras y relaciones entre ellas (conocida como *ontología*). La ontología más utilizada es WordNet (WN) [14], que agrupa las palabras en conjuntos de sinónimos llamados *synsets*. Cada synset representa un "concepto" léxico único, que en WN puede estar relacionado semánticamente con otros conceptos a través de relaciones de sinonimia, hiperonimia, hiponimia, etc., dando origen de esta manera a una jerarquía conceptual.

En el presente trabajo se utilizaron tres enfoques diferentes de WSD basado en conocimiento:

1. *CIAOSENSO*: sistema basado en la idea de *densidad conceptual*, medida como la correlación entre el sentido de una palabra y su contexto. Para ello, utiliza la longitud del camino más corto que conecta dos synsets en la taxonomía de sustantivos que utiliza WordNet. El método utiliza las relaciones jerárquicas de hiperonimia e hiponimia presentes en WordNet [15].

[1] Al usar el término *clustering* (agrupamiento) hablamos de aquella forma de categorización no-supervisada la cual, a diferencia de la categorización standard, no dispone de información sobre las clases "correctas" ni ejemplos de asignaciones de instancias a las diferentes clases.

2. *Algoritmo de Lesk*: el procedimiento determina los sentidos de las palabras que ocurren en un contexto particular basándose en una medida de solapamiento entre las definiciones de un diccionario y dicho contexto [16]. Una variante, denominada *Lesk Mejorado*, fue propuesta en [17] que considera no sólo las definiciones de las palabras a desambiguar, sino también las definiciones de aquellos términos relacionados semánticamente en la jerarquía WordNet.
3. *Método heurístico del sentido más frecuente*: sistema basado en propiedades lingüísticas aprendidas. Esta es la técnica más simple de desambiguación asignando a una palabra el sentido que ocurre más a menudo de todos los posibles sentidos de esa palabra. En este caso, los sentidos han sido obtenidos a partir de las frecuencias de ocurrencia de las palabras reportadas por WordNet.

El uso de información semántica plantea distintas alternativas respecto a la manera en que esta información puede ser incorporada en la representación de los documentos. En este trabajo, el enfoque tradicional basado en términos[2] será comparado con dos esquemas semánticos diferentes que referenciaremos como *"conceptos"* y *"términos+conceptos"*. En la primera estrategia denominada "conceptos", se genera un nuevo vector reemplazando todo término de la representación original por su concepto en WN ("synset") y eliminando aquellos términos cuyo synset no existe o no pudo ser desambiguado. Cuando se habla de "términos+conceptos", al vector de términos original se le incorporan todos los conceptos de WN obtenidos en la primera estrategia.

2.2. El algoritmo ITSA*

El algoritmo ITSA* (cuyo nombre deriva de *ITerative PAntSA**), es la versión iterativa de *PAntSA**, un método bio-inspirado diseñado para mejorar los resultados obtenidos con algoritmos de clustering arbitrarios. PAntSA* es la versión particional del algoritmo AntTree [18] que además incorpora información sobre el *Coeficiente de Silueta*[3] y el concepto de *atracción* de un cluster. En PAntSA*, los datos (en este caso *documentos*) son representados por *hormigas* que se mueven sobre una estructura de árbol de acuerdo a su similaridad con las otras hormigas ya conectadas al árbol. Cada nodo en la estructura del árbol representa una única hormiga y cada hormiga representa un único documento.

La colección completa de hormigas es representada inicialmente por una lista $\mathcal{L}$ de hormigas que esperan ser conectadas. Comenzando desde un soporte artificial a_0, todas las hormigas son conectadas de manera incremental ya sea en

[2] Con un proceso previo de eliminación de palabras de paro (o "stopword") y lematizado de las palabras.

[3] El componente fundamental de esta medida es la fórmula usada para determinar el coeficiente de cualquier objeto i, $s(i)$ y es definido como: $s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$ con $-1 \leq s(i) \leq 1$. El valor $a(i)$ denota el promedio de disimilitud del objeto i con el resto de los objetos en su propio agrupamiento, y $b(i)$ de disimilitud del objeto i a todos los objetos en el agrupamiento más cercano [19].

el soporte o bien a otra hormiga ya conectada. Este proceso continúa hasta que todas las hormigas están conectadas a la estructura. Las dos componentes principales de PAntSA* son: 1) la disposición inicial de las hormigas en la lista $\mathcal{L}$, y 2) el criterio utilizado por una hormiga arbitraria a_i ubicada sobre el soporte, para decidir cuál será la hormiga a_+ (conectada al soporte) hacia la que debería moverse. Para el primer paso, PAntSA* toma como entrada el agrupamiento obtenido con algún algoritmo de clustering arbitrario y usa información sobre el Coeficiente de Silueta (CS) de este agrupamiento para determinar el orden inicial de las hormigas en $\mathcal{L}$. Para el segundo proceso, PAntSA* usa un criterio más informado basado en el concepto de *atracción*. Aquí, si $\mathcal{G}_{a^+} = \{a^+\} \cup \mathcal{A}_{a^+}$ es el grupo formado por una hormiga a^+ conectada al soporte y sus descendientes; esta relación entre el grupo $\mathcal{G}_{a^+}$ y la hormiga a_i es referida como la *atracción de* $\mathcal{G}_{a^+}$ *sobre* a_i y será denotada como $att(a_i, \mathcal{G}_{a^+})$.

PAntSA* se diferencia de AntTree en otro aspecto importante: no construye estructuras jerárquicas con raíces (hormigas) directamente conectadas al soporte. De esta manera, se simplifica el algoritmo al suprimirse algunos parámetros necesarios para la generación de la jerarquía. En PAntSA*, cada hormiga a_j conectada al soporte (a_0) y sus descendientes (el grupo $\mathcal{G}_{a_j}$) es considerado simplemente como un *conjunto*. De esta manera, cuando una hormiga arbitraria a_i debe ser incorporada en el grupo de la hormiga a^+ que más atracción ejerce sobre a_i, este paso es implementado como un simple agregado de a_i al conjunto $\mathcal{G}_{a^+}$. El algoritmo PAntSA* resultante es mostrado en la Figura 1, donde es posible observar que toma un clustering arbitrario como entrada y lleva a cabo los siguientes tres pasos para obtener el nuevo clustering: 1) Conección al soporte, 2) Generación de la lista $\mathcal{L}$ y 3) Agrupamiento de las hormigas en $\mathcal{L}$.

En el primer paso, la hormiga más representativa de cada grupo del agrupamiento recibido como entrada es conectada al soporte a_0. Esta tarea, consiste en seleccionar la hormiga a_i con el valor de CS más alto de cada grupo C_i, y conectar cada una de ellas al soporte, formando un conjunto de un único elemento $\mathcal{G}_{a_i}$. El segundo paso, consiste en generar la lista $\mathcal{L}$ con las hormigas que no fueron conectadas en el paso previo. Este proceso también considera el ordenamiento basado en el CS obtenido en el paso previo, y mezcla las hormigas restantes (ordenadas) de cada grupo tomando, en forma iterativa, la primera hormiga de cada fila no vacía, hasta que todas las filas están vacías. En el tercer paso, el orden en el cual las hormigas serán procesadas es determinado por sus posiciones en la lista $\mathcal{L}$. El proceso de clustering de cada hormiga arbitraria a_i simplemente determina cuál es la hormiga conectada a^+ que ejerce más atracción sobre a_i[4] y entonces incorpora a a_i en el grupo de a^+ ($\mathcal{G}_{a^+}$). El algoritmo finalmente retorna un agrupamiento formado por los grupos de las hormigas conectadas al soporte.

Una vez que el algoritmo PAntSA* es implementado, su versión iterativa llamada ITSA*, puede ser fácilmente lograda. ITSA* sólo deberá proveer un agrupamiento inicial a PAntSA*, luego tomar la salida de PAntSA* que servirá co-

[4] Actualmente, como medida de atracción usamos la similitud promedio entre a_i y todas las hormigas en $\mathcal{G}_{a^+}$ pero otras medidas alternativas también serían válidas.

función PAntSA$^\star(\mathcal{C})$ **retorna** un clustering $\mathcal{C}^\star$
Entrada: $\mathcal{C} = \{C_1, \ldots, C_k\}$, un agrupamiento inicial

1. Conección al soporte
1.a. Crear un conjunto $\mathcal{Q} = \{q_1, \ldots, q_k\}$ de k filas de datos (una fila por cada grupo $C_j \in \mathcal{C}$).
1.b. Ordenar cada fila $q_j \in \mathcal{Q}$ en orden decreciente de acuerdo al Coeficiente de Silueta de sus elementos. Sea $\mathcal{Q}' = \{q'_1, \ldots, q'_k\}$ el conjunto de filas ordenadas que resulta de este proceso.
1.c. Sea $\mathcal{G}_\mathcal{F} = \{a_1, \ldots, a_k\}$ el conjunto formado por la primer hormiga a_i de cada fila $q'_i \in \mathcal{Q}'$. Por cada hormiga $a_i \in \mathcal{G}_\mathcal{F}$, remover a_i de q'_i e inicializar $\mathcal{G}_{a_i} = \{a_i\}$ (conectar a_i al soporte a_0).

2. Generación de la lista $\mathcal{L}$
2.a. Sea $\mathcal{Q}'' = \{q''_1, \ldots, q''_k\}$ el conjunto de filas resultante del proceso previo de remoción de la primer hormiga de cada fila en $\mathcal{Q}'$.
Generar la lista $\mathcal{L}$ mezclando las filas en $\mathcal{Q}''$.

3. Proceso de Clustering
3.a. Repetir
3.a.1 Seleccionar la primer hormiga a_i de la lista $\mathcal{L}$.
3.a.2 Sea a^+ la hormiga con el valor más alto de $att(a_i, \mathcal{G}_{a^+})$.
$\mathcal{G}_{a^+} \leftarrow \mathcal{G}_{a^+} \cup \{a_i\}$
Hasta que $\mathcal{L}$ esté vacía

retornar $\mathcal{C}^\star = \{\mathcal{G}_{a_1}, \ldots, \mathcal{G}_{a_k}\}$

Figura 1. El algoritmo PAntSA*.

mo nueva entrada para el mismo algoritmo en la siguiente iteración, y repetir este proceso hasta que no se observen cambios en el agrupamiento generado por PAntSA* con respecto a la iteración previa. Es importante notar que tanto PAntSA* como ITSA* son en realidad *métodos de mejora* que trabajan sobre un agrupamiento inicial generado por un algoritmo de clustering separado. Sin embargo, en trabajos recientes [8], se ha observado que ITSA* puede independizarse de este requerimiento y obtener resultados muy competitivos aún cuando es provisto con *agrupamientos iniciales aleatorios*. Esta variante, que será la utilizada en el trabajo experimental, permite obtener una versión totalmente autocontenida de ITSA* como algoritmo de clustering, e independizarse de esta manera del algoritmo de clustering particular utilizado para generar el agrupamiento inicial.

3. Diseño Experimental y Análisis de los Resultados

Para el trabajo experimental se utilizaron cuatro colecciones con diferentes niveles de complejidad respecto al tamaño, longitud de los documentos y sola-

pamiento de los vocabularios: CICling-2002, EasyAbstracts, R8+ y R8-. CICling-2002 es una colección de textos cortos muy popular que ha sido reconocida como de alta complejidad debido a que sus documentos son resúmenes científicos que pertenecen a un dominio muy restringido (lingüística computacional). La colección EasyAbstracts está compuesta de documentos de corta longitud que también son resúmenes científicos, pero que tratan sobre tópicos bien diferenciados entre sí, por lo que es una colección considerablemente más fácil para trabajar que CICling-2002. Las colecciones previas, son colecciones pequeñas que han permitido en trabajos previos, realizar un análisis detallado que sería dificultoso llevar a cabo si se trabaja con colecciones de gran tamaño. Desafortunadamente, si sólo estos conjuntos de datos fueran considerados no sería posible determinar si las conclusiones también son extensibles a colecciones de mayor tamaño. Por esta razón, en los experimentos también se consideraron otras dos colecciones, R8+ y R8-, dos subconjuntos del corpus R8-Test el cual, a su vez, es una subcolección muy utilizada de la popular Reuters-21578. La diferencia entre R8+ y R8- es que en el primer caso se tomó el 20 % de los documentos más largos de cada clase mientras que en el segundo se seleccionó el 20 % de los documentos más cortos de cada una de ellas. De esta manera, la longitud de los documentos de R8+ es, en promedio, 10 veces la longitud de los documentos de R8-. [5]

Los documentos fueron representados mediante el modelo de espacio vector (Sección 2.1), y la representación utilizada para codificar cada texto fue enriquecido a partir de la incorporación de información semántica, obteniéndose los vectores de "conceptos" y "términos+conceptos". Los "conceptos" fueron obtenidos mediante los tres enfoques descriptos previamente: CIAOSENSO (CIAO), Lesk Mejorado (LM) y el método heurístico del sentido más frecuente (MFS, por sus siglas en inglés). Los documentos fueron representados con la codificación stantard (normalizada) tf-idf (codificación SMART ntc) luego de un proceso de eliminación de *palabras de paro (stop-word)*. Como vectores de conceptos se utilizaron aquellos que reportaron los mejores resultados en trabajos previos [23] los cuales corresponden, en algunos casos, a distintos métodos de WSD dependiendo de la colección considerada. Para estimar la similitud entre documentos se utilizó la popular *medida coseno*. La inicialización de parámetros para CLUDIPSO y los restantes algoritmos usados en las comparaciones corresponden a los parámetros derivados empíricamente en [3].

3.1. Resultados Experimentales

Los resultados of ITSA* fueron comparados con los resultados de PAntSA* y otros cuatro algoritmos de clustering: K-means, K-MajorClust [24], CHAMELEON [25] y CLUDIPSO [3]. Todos estos algoritmos han sido usados en estudios similares y en particular, CLUDIPSO ha obtenido en trabajos previos [3] los mejores resultados con dos de las colecciones de textos cortos que utilizaremos en los

[5] Restricciones de espacio nos impiden brindar una descripción detallada de estas colecciones pero en [20,21,22,2,3] es posible obtener más información sobre las características de las mismas y sus enlaces de acceso.

experimentos, CICling-2002 y EasyAbstracts. La calidad de los resultados fue evaluada mediante la medida (externa) clásica denominada *medida F* (F-measure) sobre los agrupamientos que cada algoritmo generó en 50 ejecuciones independientes por colección. Los resultados reportados corresponden a los valores de medida F mínimo (F_{min}), máximo (F_{max}) y promedio (F_{avg}). Los valores resaltados en negrita en la tabla con los resultados indican los mejores resultados obtenidos en cada caso.

La Tabla 1 muestra los valores F_{min}, F_{max} y F_{avg} que K-means, K-MajorClust, CHAMELEON y CLUDIPSO obtuvieron con las cuatro colecciones. También se incluyen los resultados obtenidos con PAntSA* e ITSA* tomando como entrada los agrupamientos generados por un proceso simple (denotado R-Clustering) que determina de manera aleatoria el grupo de cada documento. Estos resultados de PAntSA* e ITSA* utilizando la representación de términos tradicional son denotados como R-Clust* y R-Clust** respectivamente. Los resultados de ITSA* utilizando vectores de conceptos son denotados R-Clust**-C mientras que aquellos que combinan términos y conceptos son referenciados como R-Clust**-T+C.

Los resultados obtenidos son concluyentes respecto al buen desempeño de ITSA* con las colecciones con mayor número de documentos (R8+ y R8-) cuando se incorpora información semántica en la representación de los textos cortos. En este sentido, se debe notar que ITSA* obtiene los mejores valores de F_{min}, F_{max} y F_{avg} utilizando representaciones que sólo utilizan conceptos (R-Clust**-C) en el caso de R8+ y con representaciones que utilizan conceptos y términos (R-Clust**-T+C) en el caso de R8-. Los resultados con las otras dos colecciones más pequeñas, si bien no son tan categóricos, son altamente competitivos con los obtenidos por los restantes algoritmos. En el caso de CICling-2002 (una colección que ha exhibido una alta complejidad en trabajos previos) ITSA* obtiene los mejores valores de F_{max} y F_{avg} con representaciones que sólo utilizan conceptos, y sólo es mínimamente superado en el valor de F_{min} por el algoritmo CLUDIPSO. Con respecto a EasyAbstracts, la variante R-Clust**-T+C obtiene el mejor valor de F_{avg}, si bien es superada en F_{min} por CLUDIPSO, y levemente superada por la variante de ITSA* que sólo utiliza términos (R-Clust**).

Tabla 1. Mejores valores de medida F por colección.

	CICling-2002			EasyAbstracts			R8-			R8+		
Algoritmos	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
R-Clust*	0.54	0.42	0.71	0.76	0.54	0.96	0.63	0.52	0.73	0.64	0.54	0.7
R-Clust**	0.6	0.46	0.75	0.92	0.67	**1**	0.66	0.57	0.74	0.65	0.57	0.72
R-Clust**-C	**0.62**	0.46	**0.77**	0.92	0.72	0.98	**0.72**	**0.63**	**0.82**	0.73	0.58	0.78
R-Clust**-T+C	0.59	0.46	0.72	**0.93**	0.71	0.98	0.71	0.59	0.78	**0.74**	**0.62**	**0.79**
K-Means	0.45	0.35	0.6	0.54	0.31	0.71	0.64	0.55	0.72	0.60	0.46	0.72
K-MajorClust	0.39	0.36	0.48	0.71	0.48	0.98	0.61	0.49	0.7	0.57	0.45	0.69
CHAMELEON	0.46	0.38	0.52	0.74	0.39	0.96	0.57	0.41	0.75	0.48	0.4	0.6
CLUDIPSO	0.6	**0.47**	0.73	0.92	**0.85**	0.98	0.62	0.49	0.72	0.57	0.45	0.65

4. Conclusiones y Trabajo Futuro

En este trabajo, se analizó el impacto del uso de información semántica (conceptos) en la representación de documentos, combinada con el uso del algoritmo iterativo ITSA*, en tareas de clustering de textos cortos. Los resultados obtenidos en el trabajo experimental, son altamente prometedores respecto a la efectividad de este enfoque, obteniéndose resultados muy competitivos en la mayoría de las instancias experimentales consideradas y, en algunos casos, alcanzándose los mejores resultados reportados en la literatura específica en el área. Este último aspecto, no es una observación menor, si consideramos que los métodos de WSD utilizados para obtener los conceptos (WSD basado en conocimiento) son considerados como métodos débiles en relación a otros métodos supervisados más elaborados, por lo que cualquier mejora introducida en el proceso de WSD subyacente podría resultar en una mejora del proceso de clustering completo.

Como posibles extensiones de este trabajo nos proponemos, en primer lugar, aplicar este enfoque combinado (conceptos + ITSA*) a otras colecciones de documentos cortos y también a colecciones con documentos de mayor longitud. Si bien en este trabajo sólo nos concentramos en el uso de clusterings aleatorios como entrada a ITSA*, también sería interesante analizar las posibilidades de este algoritmo como método de mejora general, y también considerar las mejoras logradas con este esquema utilizando las salidas generadas por los restantes algoritmos, mediante un estudio similar al realizado en [8], pero utilizando en este caso información semántica en la representación de los documentos. Finalmente, se debe aclarar que, si bien en el trabajo experimental sólo se consideraron los conceptos directamente obtenidos del proceso de desambiguación, también sería factible analizar el efecto de considerar aquellos conceptos disponibles siguiendo la relación de *hiperonimia* de WN. Este enfoque ya ha sido considerado en trabajos previos que utilizan información semántica con resultados favorables [26].

Referencias

1. Pinto, D., Rosso, P.: On the relative hardness of clustering corpora. In: Proc. of TSD07. Volume 4629 of LNAI., Springer-Verlag (2007) 155–161
2. Errecalde, M., Ingaramo, D., Rosso, P.: Proximity estimation and hardness of short-text corpora. In: Proceedings of TIR-2008, IEEE CS (2008) 15–19
3. Ingaramo, D., Errecalde, M., Cagnina, L., Rosso, P.: Particle Swarm Optimization for clustering short-text corpora. In: Computational Intelligence and Bioengineering. IOS press (2009) 3–19
4. Liu, W., Xiaojun, Q., Min, F., Bite, Q.: A short text modeling method combining semantic and statistical information. Inf. Sci. **180** (2010) 4031–4041
5. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2007) 787–788
6. Ingaramo, D., Errecalde, M., Rosso, P.: A new anttree-based algorithm for clustering short-text corpora. Journal of CS&T **10** (2010) 1–7

7. Ingaramo, D., Errecalde, M., Rosso, P.: A general bio-inspired method to improve the short-text clustering task. In: Proc. of CICLing 2010. LNCS 6008, Springer-Verlag (2010) 661–672
8. Errecalde, M., Ingaramo, D., Rosso, P.: Itsa*: an effective iterative method for short-text clustering tasks. In: Proc. of IEA-AIE 2010. LNAI 6096, Springer-Verlag (2010) 550–559
9. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: SIGIR. (2007) 787–788
10. Vázquez, S.: Resolución de la ambigüedad semántica mediante métodos basados en conocimiento y su aportación a tareas de PLN. PhD thesis, Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante (2009)
11. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24** (1988) 513–523
12. Salton, G.: The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall, Inc. (1971)
13. Agirre, E., Edmonds, P., eds.: Word Sense Disambiguation: Algorithms and Applications. Volume 33 of Text, Speech and Language Technology. Springer (2006)
14. Miller, G.: Wordnet: a lexical database for english. Communications of the ACM **38** (1995) 39–41
15. Buscaldi, D., Rosso, P., Masulli, F.: The upv-unige-ciaosenso wsd system. In: SENSEVAL-3: 3rd International Workshop on the Evaluation of Systems, Association for Computational Linguistics for the Semantic Analysis of Text, Barcelona, Spain (2004) 77–82
16. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th International Conference on Systems Documentation. (1986)
17. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Proc. of CICLing 2002. Volume 2276 of LNCS., Springer-Verlag (2002) 136–145
18. Azzag, H., Monmarche, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A new model for clustering with artificial ants. In: Proc. of the CEC2003, Canberra, IEEE Press (2003) 2642–2647
19. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20** (1987) 53–65
20. Makagonov, P., Alexandrov, M., Gelbukh, A.: Clustering abstracts instead of full texts. In: Proc. of TSD-2004. Volume 3206 of LNAI., Springer-Verlag (2004) 129–135
21. Alexandrov, M., Gelbukh, A., Rosso, P.: An approach to clustering abstracts. In: Proc. of NLDB-05. Volume 3513 of LNCS., Springer-Verlag (2005) 8–13
22. Pinto, D., Benedí, J.M., Rosso, P.: Clustering narrow-domain short texts by using the Kullback-Leibler distance. In: Proc. of CICLing 2007. Volume 4394 of LNCS., Springer-Verlag (2007) 611–622
23. Rosas, M.V., Errecalde, M.L., Rosso, P.: Un análisis comparativo de estrategias para la categorización semántica de textos cortos. Revista del Procesamiento del Lenguaje Natural (SEPLN) **44** (2010) 11–18
24. Stein, B., Meyer zu Eißen, S.: Document Categorization with MajorClust. In: Proc. WITS 02, Technical University of Barcelona (2002) 91–96
25. Karypis, G., Han, E.H., Vipin, K.: Chameleon: Hierarchical clustering using dynamic modeling. Computer **32** (1999) 68–75
26. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: ICDM. (2003) 541–544