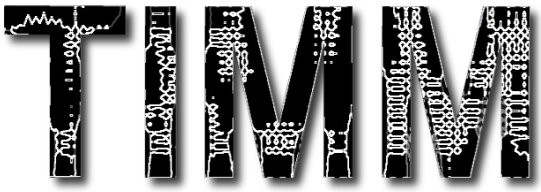
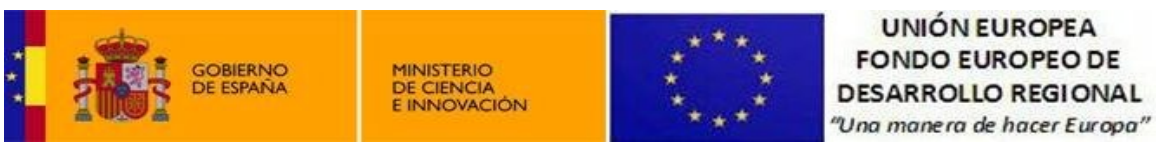
IV Jornadas TIMM
Tratamiento de la Información Multilingüe y Multimodal
7 y 8 de abril de 2011
Torres, Jaén



ACTAS

Editores: L. Alfonso Ureña y Fernando Martínez



Financiada por el MICIIN

Prólogo

La Red Temática TIMM (Tratamiento de Información Multilingüe y Multimodal), con referencia TIN2009-06135-E, dentro del programa de acciones complementarias da soporte tanto a las IV Jornadas TIMM, como a su organización en Torres (Jaén).

El objetivo general de las jornadas es promover la difusión de las actividades de investigación, desarrollo e innovación entre los diferentes grupos de investigación de ámbito nacional en el ámbito del Tratamiento de Información Multilingüe y Multimodal. Concretamente se persiguen los siguientes objetivos:

- Crear un foro donde los investigadores en formación puedan presentar y discutir su trabajo en un ambiente que facilite el intercambio de ideas y la colaboración.
- Organización de un seminario y una mesa redonda con el objetivo de hacer una puesta en común para conocer en qué estado se encuentra cada grupo participante y hacia dónde se dirige con el fin de que los grupos puedan interactuar y reutilizar los recursos de cada uno. Concretamente un seminario sobre sistemas de recomendación y minería de opiniones y una mesa redonda sobre proyectos.
- Difusión de los resultados científicos y tecnológicos mediante trabajos presentados.
- Realizar un catálogo de recursos lingüísticos y herramientas desarrolladas en los diferentes grupos de investigación para fomentar su uso y difusión entre otros grupos.

Quiero agradecer al comité de programa y a los diferentes revisores el apoyo y el trabajo realizado. Igualmente debe ser reconocida la labor realizada por el comité organizador, especialmente a Fernando Martínez. Asimismo, agredecer a Eugenio Martínez Cámara técnico de TIMM el trabajo realizado en la compilación de estas actas. Finalmente agradecer a la Red Temática TIMM, en cuyo marco se organiza por cuarta vez estas jornadas. Estas actas han sido cofinanciadas por la Red Temática (TIN2009-06135-E) del Ministerio de Ciencia e Innovación y por el Fondo Europeo de Desarrollo Regional (FEDER).

L. Alfonso Ureña

Presidente Comité Organizador y de Programa

Comité Organizador

L. Alfonso Ureña López
Fernando Martínez Santiago
Maite Martín Valdivia
Miguel A. García
Manuel C. Díaz
Arturo Montejo
Manuel García
José M. Perea
Eugenio Martínez

Comité de programa

Arantxa Díaz. Universidad del País Vasco
Mª Teresa Martín. Universidad de Jaén
Fernando Martínez . Universidad de Jaén
Patricio Martínez. Universidad de Alicante
Lidia Moreno. Universidad Politécnica de Valencia
Rafael Muñoz. Universidad de Alicante
Paolo Rosso. Universidad Politécnica de Valencia
Jose Antonio Troyano. Universidad de Sevilla
L. Alfonso Ureña López. Universidad de Jaén

Índice

Clasificación de texto

Estudio Comparativo sobre Métodos de Combinación de Clasificadores en PLN

Fernando Enríquez, José A. Troyano, Fermín Cruz, and F. Javier Ortega

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Av. Reina Mercedes s/n 41012, Sevilla (Spain)
{fenros,troyano,fcruz,javierortega}@us.es

Resumen Existen múltiples herramientas de clasificación que pueden ser utilizadas para diversas tareas del PLN, aunque ninguna de ellas puede considerarse la mejor en términos generales ya que cada una posee una lista particular de virtudes y defectos. Los métodos de combinación pueden servirnos tanto para rentabilizar al máximo las virtudes de los clasificadores base, obteniendo mejores resultados en términos de precisión, como para disminuir los errores provocados por sus defectos. Aquí se presenta un estudio comparativo sobre los más relevantes.

Keywords: Combinación de Clasificadores, Aprendizaje Automático

1. Fundamentos de la Combinación

En Hansen y Salamon [4] se establecen la precisión y la diversidad como requisitos necesarios y suficientes para llevar a cabo con éxito la combinación de dos o más sistemas de clasificación. Por su parte Dietterich [2] justifica la combinación desde tres puntos de vista como son el estadístico, el computacional, y el de representación, dejando claro que se cubre mucho mejor el espacio de búsqueda para aproximarnos a la solución óptima.

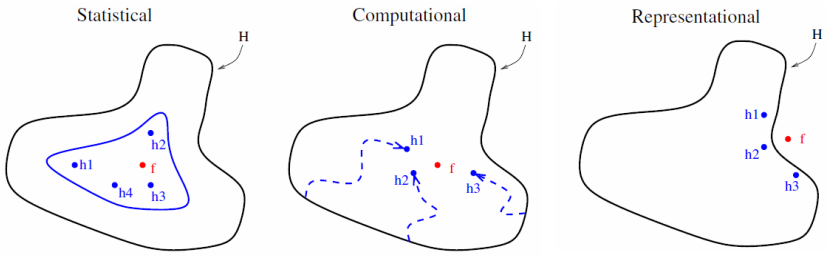


Figura 1. Justificación para la combinación según Dietterich.

En [6] se organizan los métodos de combinación en base a cuatro niveles que representan el punto del proceso donde recae el peso de la combinación,

pudiendo hacer uso de diferentes colecciones de datos (*data level*), diferentes subconjuntos de características empleadas para representar los ejemplos (*feature level*), diferentes clasificadores (*classifier level*) o distintas técnicas de combinación (*combiner level*).

Aún así, no todos los métodos de combinación existentes son aplicables a cualquier conjunto de clasificadores. Es importante considerar el tipo de información que estos producen como salida. En [7] se describen tres posibilidades: el 'abstract level' (la salida es una única etiqueta o un subconjunto de las etiquetas posibles), el 'rank level' (se devuelven las etiquetas o un subconjunto de ellas ordenadas según el orden de preferencia) y el 'measurement level' (el clasificador atribuye a cada etiqueta un valor indicativo de la confianza que se tiene en ella).

2. La Combinación y el PLN

A partir de 1998, con la publicación de los trabajos [3] y [1], fue cuando un mayor número de investigadores desarrollaron sus trabajos aplicando las técnicas de combinación a tareas del PLN. Ambos artículos se dedicaban al etiquetado POS, y aunque les sucedieron múltiples y variados trabajos, se echa en falta un estudio comparativo que abarque un mayor número de métodos y sirva para guiar al investigador a la hora de seleccionar el más adecuado.

Tras un análisis bibliográfico sobre una selección de setenta trabajos que hacen uso de alguna técnica de combinación en tareas de PLN, comprobamos la distribución de clasificadores, métodos y tareas que se muestra en la figura 2. En cuanto a las técnicas de clasificación y de combinación apreciamos un uso dispar, ya que en lo referente a los algoritmos de clasificación, si bien hay algunos métodos que destacan ligeramente del resto, existe un mayor equilibrio en cuanto a la frecuencia de uso. En los métodos de combinación sin embargo, los métodos de votación y *stacking* acaparan la mayor parte de los trabajos, dejando entrever una posible falta de experimentación con el resto de métodos que hemos comentado y que podrían ofrecer mejoras en algunas tareas del PLN.

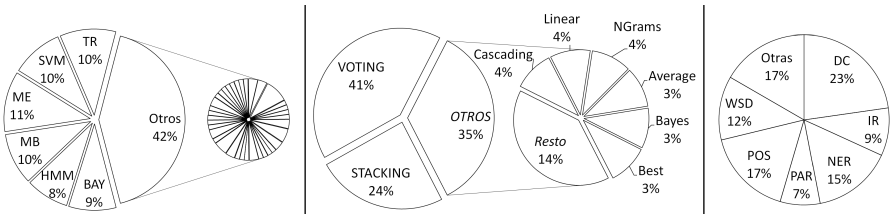


Figura 2. Clasificadores, métodos de combinación y tareas de las referencias seleccionadas.

En cuanto a los resultados presentados en los trabajos, resulta difícil establecer comparaciones debido a la gran variedad de métodos de clasificación y combinación, además de las tareas y los datos, contabilizándose cerca de 50 corpus

distintos. Aún así hemos confeccionado la tabla 1 donde se reflejan las mejoras mínima, máxima y media alcanzadas en los trabajos que aplican combinación de sistemas a alguna tarea del PLN.

	mínimo	máximo	media
DC	0,01	8,10	2,02
NER	1,30	6,41	3,52
PAR	0,03	2,30	1,12
POS	-0,58	1,75	0,75
WSD	1,70	7,00	3,34

Cuadro 1. Resumen de los resultados de las referencias seleccionadas.

3. Estudio Comparativo

Para lograr establecer un escenario más propicio para comparar los distintos métodos, hemos realizado experimentos de combinación para una tarea y clasificadores base concretos. Hemos elegido la tarea POS en la que (gracias al buen rendimiento de los etiquetadores base) podemos estar seguros de que las mejoras obtenidas por la combinación no son consecuencia de la baja calidad de los clasificadores base. Como base se han utilizado tres herramientas diseñadas para esta tarea, TnT, TreeTagger y MBT junto a un clasificador basado en características e implementado haciendo uso del software SVM^{light}[5][1] Los métodos de combinación implementados son: Bayes (BAY), *behavior knowledge space* (BKS), *stacked generalization* (SG), combinación simple de probabilidades (SPC), votación (VT) y bagging (BAG). Además se permite que la salida de un método se pueda volver a introducir como entrada en otro nivel de combinación, como si de un clasificador base se tratara, dando lugar a un esquema de *cascading* (CAS). En este caso hemos probado con dos niveles de combinación, utilizando un método de combinación para recibir las salidas del resto de métodos, que a su vez trabajan con las etiquetas propuestas por los clasificadores base. Todos los métodos han sido evaluados mediante cinco corpus muy diferentes, tanto por idioma como por su tamaño y por el conjunto de etiquetas que utilizan.

En la tabla 2 se muestran los resultados obtenidos por los clasificadores y las mejoras logradas por los diferentes métodos de combinación. Podemos comprobar que las mejoras son significativas en todos los casos, siendo *stacking* el método que mejores resultados obtiene, mostrándose como el que mejor se adapta a los diferentes tipos de datos. También *cascading*, con sus dos niveles de combinación, hace gala de una robustez que destaca al conseguir muy buen resultado. No obstante hay que destacar también los buenos resultados de métodos más

[1] http://www.cs.cornell.edu/People/tj/svm_light/

CORPUS	Idioma	Clasificadores				Combinación						
		FV	MBT	TnT	TT	BAY	BKS	SG	SPC	VT	BAG	CAS
Brown	Inglés	96,18	95,82	96,55	95,64	0,39	0,63	0,64	0,51	0,49	0,32	0,67
Floresta	Portugués	96,52	95,81	97,02	96,66	0,55	0,72	0,78	0,60	0,63	0,36	0,71
Susanne	Inglés	92,26	91,16	93,61	91,27	0,67	1,36	1,26	1,16	0,71	0,81	1,52
Talp	Español	94,59	94,80	95,82	95,62	0,96	1,08	1,10	1,10	0,76	0,75	1,18
Treebank	Inglés	96,28	95,67	96,21	95,52	0,27	0,47	0,59	0,44	0,45	0,35	0,55
PROMEDIO		95,17	94,65	95,84	94,94	0,57	0,85	0,87	0,76	0,61	0,52	0,93

Cuadro 2. Resultados obtenidos.

sencillos, como el *behavior knowledge space*, que pueden resultar muy útiles en sistemas donde prima la velocidad en lugar de la precisión.

Referencias

1. E. Brill and J. Wu. Classifier combination for improved lexical disambiguation. *Proceedings of the 17th international conference on Computational linguistics*, pages 191–195, 1998.
2. T. G. Dietterich. Ensemble methods in machine learning. *In J. Kittler and F. Roli, editors, Multiple Classifier Systems, Lecture Notes in Computer Science*, 1857:1–15, 2000.
3. H.V. Halteren, J. Zavrel, and W. Daelemans. Improving data driven wordclass tagging by system combination. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 1:491–497, 1998.
4. L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
5. T. Joachims. *Making large-Scale SVM Learning Practical*, chapter 11. MIT Press, 1999.
6. L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.
7. L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:418–435, 1992.

Actas IV Jornadas TIMM, pp 13-16

Detección de Spam en la Web mediante el análisis de texto y de grafos

F. Javier Ortega, José A. Troyano, Fermín Cruz, and Fernando Enríquez

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Av. Reina Mercedes s/n 41012, Sevilla (Spain)
{javierortega,troyano,fcruz,fenros}@us.es

Resumen El spam en la web representa un grave problema para los sistemas de Recuperación de Información, debido al perjuicio que puede ocasionar en la calidad de los resultados de los mismos. En este trabajo se presenta un sistema de detección de spam en la web basado en un algoritmo de ranking que ordena las páginas web de acuerdo a su relevancia, penalizando aquellas páginas susceptibles de ser consideradas spam. La novedad de este sistema reside en conjugar técnicas de procesamiento de textos con técnicas de análisis de grafos. Las técnicas de procesamiento de textos se utilizan para asignar a determinadas páginas una puntuación a priori, de acuerdo a la probabilidad de que sean spam o no, según su contenido. Nuestro algoritmo de ranking procesará el grafo de las páginas web y las puntuaciones a priori para obtener el ranking de webs. En los experimentos se comprueba que nuestro sistema mejora los resultados de otras técnicas muy utilizadas.

Keywords: Detección de Spam, Recuperación de Información, Algoritmos de Ranking

1. Introducción

El spam en la web consiste en la creación o modificación de páginas web con el objetivo de conseguir que los sistemas de recuperación de información indexen dichas webs, y obtener así un determinado beneficio gracias al aumento del tráfico web hacia esas páginas. Existen dos mecanismos de spam: el basado en contenido, y el basado en enlaces. El primero consiste en la modificación del contenido textual de las webs, generalmente añadiéndole a la misma una gran cantidad de palabras clave (*keyword stuffing*). El spam basado en enlaces trata de burlar a los sistemas de recuperación basados en el número de enlaces de las páginas webs. Una manera sencilla de hacer esto es crear un gran número de páginas web enlazadas entre sí, formando lo que se conoce como *granja de enlaces*. De esta forma, cada página creada tendrá un gran número de enlaces entrantes y, por tanto, su reputación se verá incrementada.

Existen muchos trabajos sobre detección de spam. En general se suelen enfocar hacia uno de los dos tipos de spam mencionados antes. Así podemos encontrar

sistemas basados en la detección de spam mediante el análisis de grafos. Dentro de este grupo se encuentra el TrustRank [5], basado en el cálculo de un ranking de páginas, utilizando un conjunto de páginas fiables seleccionadas a mano (semillas) para introducir un sesgo en la ejecución del algoritmo, de forma que dichas semillas tengan un mayor peso respecto al resto de páginas. Otro trabajo en esta línea es el presentado en [1].

Otras técnicas se basan en el contenido textual de las páginas para determinar si una web es o no spam. Estos métodos normalmente analizan la distribución de determinadas heurísticas sobre el contenido de páginas de spam y no spam, para generar un clasificador de documentos [4], [3]. Algunas métricas utilizadas para esta tarea son el número de palabras en una web, el contenido HTML invisible, la longitud media de las palabras de una página, etc.

Nuestro sistema combina conceptos de ambos tipos de técnicas, de forma que el contenido textual y los enlaces ayuden a detectar las webs de spam.

El resto del trabajo se organiza de la siguiente forma. En la Sección 2 explicamos nuestra propuesta. Seguidamente, en la Sección 3 se muestran los resultados experimentales de nuestro sistema. Finalmente en la Sección 4 presentamos las conclusiones y trabajos futuros.

2. Combinando enlaces y contenidos

Tanto las técnicas basadas en contenido como las basadas en enlaces han mostrado ser métodos fiables en la detección de spam, si bien ambos métodos presentan ciertas debilidades. Las técnicas basadas en contenido fallan a la hora de detectar granjas de enlaces. Por su parte las técnicas basadas en enlaces no tienen en cuenta el contenido de las páginas web en sus cálculos, lo que puede ocasionar que una página con un determinado contenido no deseado sea considerada relevante de acuerdo a sus enlaces.

Nuestro sistema consta de dos partes: un algoritmo de ranking y un conjunto de heurísticas basadas en el contenido. El objetivo de estas heurísticas es obtener cierta información a priori sobre la probabilidad de que una página sea o no spam, atendiendo a su contenido. Los valores de dichas métricas se incluirán dentro del algoritmo de ranking para añadir un sesgo en los cálculos, de forma que las páginas relacionadas con una web de spam vean disminuido su ranking global, y viceversa. En nuestro sistema hemos implementado dos métricas: la longitud media de las palabras de una página y el número de palabras repetidas. Por su parte, el algoritmo de ranking es una adaptación del PageRank [**?**], pero a diferencia de éste calcula dos puntuaciones para cada web: PR^+, que representa la relevancia de una página, y PR^-,que es la probabilidad de que dicha página sea spam. El ranking de webs se calcula según la diferencia entre PR^+ y PR^-. Estas puntuaciones se calculan siguiendo las Ecuaciones (1) y (2).

$$PR^+(v_i) = (1-d)e_i^+ + d \sum_{j \in In(v_i)} \frac{PR^+(v_j)}{|Out(v_j)|} \quad (1)$$

$$PR^{-}(v_i) = (1-d)e_i^{-} + d \sum_{j \in In(v_i)} \frac{PR^{-}(v_j)}{|Out(v_j)|} \quad (2)$$

donde v_i es un nodo del grafo (una página web), $In(v_j)$ es el conjunto de webs apuntando a v_j, y $Out(v_j)$ es el conjunto de nodos hacia los que apunta v_j. El algoritmo itera sobre los nodos del grafo, aplicando las ecuaciones (1) y (2), hasta que la diferencia máxima entre las puntuaciones de los nodos de dos iteraciones consecutivas sea menor que un umbral dado, t. Los vectores e_i^{+} y e_i^{-} son los encargados de incluir en el algoritmo las métricas basadas en contenido. De esta forma, según se inicialicen ambos vectores, podremos dar mayor importancia a algunas webs frente a otras. Estas webs son las *semillas* de nuestro algoritmo. Hemos implementado tres métodos de selección de semillas:

- Páginas Más Positivas y Páginas Más Negativas (MPN): seleccionamos como semillas las N páginas con mayores y menores valores de las métricas. Cada semilla se inicializa con un valor de $e_i = 1/N$.
- Más Positivas y Páginas Más Negativas con Métricas (MPN-M): se eligen como semillas las mismas webs que en el apartado anterior, pero sus puntuaciones se inicializan según los valores de las métricas, de forma que $e_i = Metricas_i/N$.
- Todas las Páginas como Semillas (TPS): se utilizan todos los nodos como semillas, aplicando la misma puntuación que la vista para MPN-M.

3. Experimentación y resultados

Dado que nuestro sistema no clasifica las páginas webs entre spam y no spam, sino que genera un ranking de las mismas, no podemos aplicar como método de evaluación las métricas típicas de los sistemas de clasificación. En su lugar, hemos implementado una técnica de evaluación muy utilizada en otros trabajos de detección de spam en la web, conocida como *PR-buckets* [5]. Este método da mayor relevancia a los fallos producidos por webs de spam que consiguen burlar al sistema colocándose en las primeras posiciones del ranking, dado que estas posiciones son las más importantes. Para tener una idea de la validez de los resultados de nuestro sistema, hemos ejecutado experimentos con una conocida técnica de detección de spam, TrustRank. Como hemos comentado, esta técnica se basa también en un algoritmo de ranking, aunque la selección de semillas se realiza a mano, al contrario que en nuestro sistema, en el que la selección de semillas es automática.

Los experimentos se han realizado con el corpus WEBSPAM-UK2006 Dataset [2], compilado expresamente para la tarea de la detección de spam en la web. El corpus está formado por unos 98 millones de páginas web, y unos 120 millones de enlaces. Un subconjunto de unos 11000 sitios web fueron etiquetados como spam o no spam, conteniendo un total de 10 millones de páginas de spam. Como marco de trabajo se ha utilizado el sistema de Recuperación de Información Terrier[1] para el indexado del corpus.

En la tabla 1 mostramos los resultados de nuestra técnica, con sus tres métodos de selección de semillas, junto con el algoritmo de TrustRank. Mostramos los resultados de los 10 primeros buckets, relativos a las primeras posiciones del ranking.

[1] `http://terrier.org/`

#	Páginas	TR	MPN	MPN-M	TPS
1	14	**0**	2	**0**	**0**
2	68	2	5	**1**	3
3	212	17	16	**4**	8
4	649	40	48	**21**	32
5	1719	73	104	**66**	101
6	3849	155	244	**124**	199
7	6513	254	392	**180**	297
8	9291	371	557	**255**	416
9	12102	448	742	**350**	537
10	14914	511	937	**440**	650

Cuadro 1. Número de errores acumulados para cada método: TrustRank (TR), Páginas Más Positivas/Negativas (MPN), Más Positivas/Negativas con Métricas (MPN-M) y Todas las Páginas como Semillas (TPS). El mejor resultado se muestra resaltado.

4. Conclusiones y trabajo futuro

En este trabajo hemos presentado un sistema de detección de spam en la web basado tanto en el contenido textual de las páginas como en los enlaces de las mismas. El sistema se ha evaluado con un corpus específico para la tarea de detección de spam, obteniendo muy buenos resultados, e incluso mejorando los de la técnica de TrustRank.

Como trabajos futuros nos planteamos la mejora de este método, implementando nuevas métricas basadas en contenido, y estudiando el impacto de dichas métricas en el rendimiento global del sistema, y en la complejidad en tiempo del mismo. También resultaría interesante evaluar la influencia del conjunto de semillas en los resultados finales del algoritmo, y estudiar posibles mejoras a los métodos de selección propuestos.

Referencias

1. Luca Becchetti, Carlos Castillo, Debora Donato, Ricardo Baeza-YATES, and Stefano Leonardi. Link analysis for web spam detection. *ACM Transactions on the Web*, 2(1):1–42, February 2008.
2. Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.
3. Gordon V Cormack, Mark Smucker, and Charles L A Clarke. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *Computing Research Repository*, abs/1004.5, 2010.
4. Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases*, pages 1–6, New York, NY, USA, 2004. ACM.
5. Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases*, volume 30, pages 576–587, Toronto, Canada, 2004. VLDB Endowment.

Actas IV Jornadas TIMM, pp 17-20

Algoritmos bio-inspirados aplicados a tareas de clasificación de textos cortos

Leticia Cagnina[1], Marcelo Errecalde[1], Paolo Rosso[2] *

[1] LIDIC (Research Group). Universidad Nacional de San Luis. Argentina.
{lcagnina,merreca}@unsl.edu.ar
[2] Natural Language Engineering Lab. - ELiRF, DSIC, Universidad Politécnica de Valencia. España. prosso@dsic.upv.es

Resumen El agrupamiento de textos cortos como así también la atribución de autoría son dos problemas típicos y de gran interés en el área de Procesamiento del Lenguaje Natural. Trabajos previos han demostrado que algoritmos bio-inspirados tales como los basados en Particle Swarm Optimization han resultado efectivos y eficientes para la resolución de problemas de agrupamiento de textos cortos. Con base en estas experiencias, se pretende aplicar este tipo de algoritmos para resolver problemas de atribución de autoría.

Palabras Clave: Agrupamiento de Textos Cortos, Atribución de Autoría, Particle Swarm Optimization.

1. Trabajo Previo: Agrupamiento de Textos Cortos

El *agrupamiento de textos cortos* es una tarea importante del Procesamiento del Lenguaje Natural ya que está presente en aplicaciones como minería de textos, extracción de información de la web, generación de textos y otras derivadas del uso de lenguajes reducidos en blogs y mensajes de textos. El objetivo de un problema de agrupamiento de textos es clasificar un conjunto de documentos en diferentes grupos. Si los documentos son cortos, el problema se dificulta debido a la baja frecuencia de términos presentes en cada texto. Este último problema es una aplicación corriente ya que permite organizar grandes volúmenes de información (expresada en textos cortos) en un número reducido de grupos significativos. En problemas de agrupamiento de textos cortos no se cuenta con información referida a los grupos ni la clasificación correcta de los documentos, dificultándose así la evaluación de una potencial solución a través de medidas externas como la *Medida F* o la *Entropía*. Como consecuencia de ello, la calidad de la solución debe ser evaluada con respecto a propiedades estructurales expresadas en medidas de validación interna como el *coeficiente de Silhouette Global*

* La investigación de la primera autora está parcialmente financiado por el programa de Estancias en la UPV de Investigadores de Prestigio PAID-02-10 N 2257. La investigación de los dos últimos autores está parcialmente financiado por el proyecto MICINN TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 (Plan I+D+i).

(GS)[17] y la *Medida de Densidad Esperada* (DEM)[11]. Estas dos medidas son utilizadas frecuentemente ya que proveen una buena estimación de la calidad de los grupos de la solución obtenida además de aportar un buen grado de correlación con respecto a la correcta clasificación realizada por una persona. Estos dos motivos han llevado a que las medidas GS y DEM sean utilizadas como una función objetivo para optimizar por distintos algoritmos de agrupamiento.
En [10] dos versiones distintas de algoritmos basados en la técnica de optimización bio-inspirada Particle Swarm Optimization (PSO)[5] fueron presentadas para resolver el problema de agrupamiento de 3 colecciones pequeñas: *Micro4News*, *EasyAbstract* y *CICLing-2002* [6]. Una de las versiones es un algoritmo PSO discreto denominado CLUDIPSO cuya representación de soluciones es un vector de n números enteros, indicando cada uno de ellos el grupo al cual pertenece cada uno de los n documentos de la colección. La otra versión es un algoritmo PSO continuo denominado CLUCOPSO cuya representación de soluciones es un vector $(K \times T)$ de números reales donde K indica el número de centroides (uno por cada grupo de la colección) de T términos. Ambas versiones utilizan las medidas GS y EDM como función objetivo a optimizar. Las conclusiones del trabajo indican que CLUDIPSO tuvo un desempeño consistente en las 3 colecciones evaluadas logrando superar a algoritmos efectivos representativos del área como K-Means, MajorClust [18] y DBSCAN [7], cuando la medida GS fue empleada. Los mejores resultados de CLUCOPSO fueron alcanzados con la medida DEM pero siendo competitivos sólo en colecciones de mediana y alta complejidad (EasyAbstract y CICLing-2002). Los Cuadros 1, 2 y 3 ilustran los resultados obtenidos con los diferentes algoritmos. Notar que los mejores valores fueron resaltados.

Cuadro 1. Micro4News: Valores medios, mínimos y máximos de DEM y GS.

Algoritmo	DEM med	DEM min	DEM max	GS med	GS min	GS max
K-Means	0.99	0.89	1.07	0.39	0.05	0.74
MajorClust	1.08	1.05	1.10	0.69	0.64	0.74
DBSCAN	1.05	1.01	1.10	0.54	0.36	0.67
CLUDIPSO	1.07	1.06	1.07	**0.72**	**0.69**	**0.74**
CLUCOPSO	**1.11**	**1.10**	**1.12**	0.26	0.19	0.36

Cuadro 2. EasyAbstract: Valores medios, mínimos y máximos de DEM y GS.

Algoritmo	DEM med	DEM min	DEM max	GS med	GS min	GS max
K-Means	0.9	0.86	0.92	0.08	-0.05	0.29
MajorClust	0.94	**0.93**	0.96	0.31	-0.01	**0.50**
DBSCAN	0.93	0.91	0.94	0.23	0.08	0.32
CLUDIPSO	0.94	**0.93**	0.95	**0.47**	**0.44**	**0.50**
CLUCOPSO	**0.98**	0.92	**1.03**	0.25	0.21	0.32

2. Atribución de Autoría de Textos Cortos

La atribución de autoría es la tarea de determinar el autor de un texto considerando un conjunto de documentos de varios autores candidatos. Esta tarea

Cuadro 3. CICLing-2002: Valores medios, mínimos y máximos de DEM y GS.

Algoritmo	DEM med	DEM min	DEM max	GS med	GS min	GS max
K-Means	0.87	0.84	0.91	0.07	-0.06	0.22
MajorClust	0.92	0.91	0.94	0.14	-0,24	0.36
DBSCAN	0.91	0.88	0.95	0.08	-0.11	0.21
CLUDIPSO	0.92	0.91	0.93	**0.39**	**0.36**	**0.41**
CLUCOPSO	**1.07**	**1.06**	**1.11**	0.16	0.14	0.18

puede ser empleada en disputas sobre autoría de obras literarias [14], para investigaciones criminales [2] y verificación de autoría de mensajes o correos electrónicos [3], entre otras.

Una forma de resolver la tarea de atribución de autoría es utilizando las características estilográficas de escritura que posee cada autor. En [1] se identifican 3 tareas estilográficas fundamentales en aplicaciones de recuperación de la información: la caracterización del autor a través de un estilo único, la detección de similitudes y luego, la atribución de autoría. La caracterización del autor refleja información específica como género, educación, nivel social, etc. y debe ser invariante entre textos del mismo autor. La detección de similitudes se enfoca en la comparación de varios textos de forma tal de detectar propiedades comunes entre ellos. Finalmente, la atribución de autoría permitirá identificar el autor de un texto utilizando alguna medida de similitud con otros textos de autoría conocida.

Con base en las 3 tareas enunciadas, es posible automatizar la atribución de autoría mediante algoritmos que permitan: (1) representar los textos de forma tal que queden reflejadas características estilográficas propias del autor, (2) utilizar una función que permita medir similitudes entre textos de autores candidatos y el texto de autoría desconocida y, (3) decidir el autor del texto más probable.

Comúnmente, la forma más utilizada de automatizar la atribución de autoría es considerando éste como un problema de clasificación [13,8,15]. Haciendo uso del conocimiento de la buena prestación de algoritmos bio-inspirados en tareas de agrupamiento de textos cortos, actualmente se está estudiando la manera de resolver el problema de atribución de autoría, particularmente de textos cortos, utilizando estos algoritmos. El problema se transformaría en uno de agrupamiento más que de clasificación, empleando los documentos candidatos como grupos predefinidos. Luego, a través de alguna medida de similitud, el algoritmo debería ser capaz de seleccionar el grupo (uno por cada autor candidato) más probable para el documento de autoría desconocida. Se están estudiando varias formas de representación de los documentos de forma tal que se capturen las principales características estilográficas del autor. Las que se evaluarán son: uso de palabras cortas (dos o tres letras) [8], frecuencia de ocurrencia de palabras funcionales [15], utilización de $n-$gramas [9] o algunas específicas que emplean un ordenamiento de frecuencias de palabras [4]. Como función de similitud a optimizar se pretende evaluar las siguientes medidas: SVM basada en la frecuencia de las palabras [12] y la medida *Relative Hardness* [16] que utiliza el grado de

solapamiento de vocabulario entre grupos. Como conclusión de este estudio se persigue determinar si el problema de atribución de autoría puede ser abordado como uno de clasificación, con un algoritmo bio-inspirado.

Referencias

1. G. Bonanno, F. Moschella, S. Rinaudo, P. Pantano, and V. Talarico. Manual and evolutionary equalization in text mining. In *Proc. of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, pages 262–267, 2007.
2. C. Chaski. Empirical evaluations of the language-based author identification techniques. *Forensic Linguistics*, 8(1):1–65, 2001.
3. O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD Record*, 30(4):55–64, 2001.
4. L. Dinu and M. Popescu. Ordinal measures in authorship identification. In *PAN'09*, pages 62–66, 2009.
5. R. Eberhart and Y. Shi. A modified particle swarm optimizer. In *International Conference on Evolutionary Computation*. IEEE Service Center, 1998.
6. M. Errecalde and D. Ingaramo. Short-text corpora for clustering evaluation. Technical report, LIDIC, 2008.
7. M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.
8. D. I. Holmes. The Evolution of Stylometry in Humanities Scholarship. *Literary and Linguist Computing*, 13(3):111–117, 1998.
9. J. Houvardas and E. Stamatatos. N-Gram Feature Selection for Authorship Identification. volume 4183 of *LNCS*, chapter 10, pages 77–86. 2006.
10. D. Ingaramo, M. Errecalde, L. Cagnina, and P. Rosso. *Computational Intelligence and Bioengineering*, chapter Particle Swarm Optimization for Clustering short-text Corpora, pages 3–19. IOS Press, 2009. F. Masulli et al. (Eds.).
11. D. Ingaramo, D. Pinto, P. Rosso, and M. Errecalde. Evaluation of internal validity measures in short-text corpora. *Proc. of the CICLing 2008 Conference. LNCS*, 4919:555–567, 2008. Publisher Springer-Verlag.
12. M. Koppel, J. Schler, and E. Bonchek-Dokow. Measuring Differentiability: Unmasking Pseudonymous Authors. *J. of Mach. Lear. Research*, 8:1261–1276, 2007.
13. R. Matthews and T Merriam. Neural computation in stylometry i: An application to the works of shakespeare and fletcher. *Literary and Linguist Computing*, 8(4):203–209, 1993.
14. F. Mosteller and D. L. Wallace. Inference in an authorship problem. *Journal of the American Statistical Association*, 58(302):275–309, 1963.
15. F. Mosteller and D. L. Wallace. *Inference and Disputed Authorship: The Federalist.* Addison-Wesley, 1964.
16. D. Pinto and P. Rosso. On the relative hardness of clustering corpora. In *TSD*, pages 155–161, 2007.
17. P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20:53–65, 1987.
18. B. Stein and O.Ñiggemann. On the nature of structure and its identification. In *Proc. of the 25th International Workshop on Graph Theoretic Concepts in Computer Science - WG99, LNCS*, volume 1665, pages 122–134. Springer-Verlag, 1999.

Actas IV Jornadas TIMM, pp 21-24

Detección de reuso de código fuente entre lenguajes de programación con base en la frecuencia de términos

Enrique Flores, Alberto Barrón-Cedeño, Paolo Rosso, and Lidia Moreno

Universidad Politécnica de Valencia, Dpto. de Sistemas Informáticos y Computación
Camino de Vera s/n, E-46022 Valencia, España
{eflores,lbarron,prosso,lmoreno}@dsic.upv.es
http://www.dsic.upv.es/

Resumen En la actualidad, hay muchos repositorios públicos donde cualquiera puede acudir y obtener un código fuente o parte de él y utilizarlo en sus programas sin permiso del autor o sin citarlo. Otro caso más común se da en el mundo académico, ya que cuando varios estudiantes deben de realizar el mismo trabajo algunos de ellos pueden copiar o modificar el trabajo de los compañeros para que parezca un trabajo diferente.
En este artículo proponemos un sistema que intenta detectar la reutilización de código fuente incluso entre lenguajes de programación distintos, estudiando la influencia de los comentarios, los nombres de variables y las palabras del lenguaje. Las técnicas aplicadas para la detección de la similitud han sido Frecuencia de Términos (tf) y Frecuencia Inversa (tf-idf) considerando como términos los n-gramas de carácteres.

Keywords: Reuso de Código Fuente, Detección de Plagio Multilíngüe, Procesamiento del Lenguaje Natural

1. Introducción

La popularización del uso de Internet ha posibilitado la existencia de enormes cantidades de información al alcance de cualquier usuario. Las obras publicadas en este medio digital están expuestas a copias y reutilización de todo o parte de ellas. Por ello, existe un gran interés en identificar la autoría y el posible plagio en las obras publicadas. Sin embargo, no es trivial la localización y comparación de toda la información relevante para asegurar un plagio, o al menos cierto grado de similitud entre obras.

La ingeniería lingüíística, más conocida en el área de la Inteligencia Artificial como Procesamiento del Lenguaje Natural (PLN), facilita el tratamiento automático de la documentación textual. Un desafío interesante cosiste en aplicar recursos y técnicas de PLN con el fin de detectar similitud, incluso para texto traducido entre distintas lenguas [7].

Actualmente, la disponibilidad de código fuente en la red es muy amplia facilitando la reutilización total o parcial de programas que han sido previamente implementados y testeados por el propio autor o bien por autores externos.

Un campo de especial aplicación de los problemas expuestos se da en el ámbito académico, medio propenso a la copia de las tareas correspondientes a una determinada materia por parte del alumnado.

Otra forma de reuso de código fuente consiste en encontrar dentro de un repositorio de código fuente algún algoritmo implementado en un lenguaje de programación y traducirlo a otro lenguaje de interés para el programador que ha realizado la búsqueda. La disponibilidad de código fuente traducido a otro lenguaje de programación puede dar lugar a una nueva línea de investigación dentro del campo de atribución de autoría o detección de plagio. Para dar solución a este tipo de problema, se debería disponer de sistemas con capacidad de detección de un código fuente similar a otro escrito en un lenguaje de programación diferente.

La carencia de estudios dirigidos a este tipo de reuso de código fuente traducido entre diferentes lenguajes de programación ha motivado la realización de este trabajo. Los autores pretenden demostrar que es posible detectar la similitud de código fuente monolíngüe o multilíngüe aplicando técnicas de PLN a través de los experimentos realizados.

2. Método propuesto

El método que vamos a utilizar para la detección de similitud entre códigos fuente es *term frequency* (tf). Este método consiste en dividir el documento que contiene el código fuente en términos (en nuestro caso en n-gramas de carácteres), calcular la frecuencia de aparición de esos términos y normalizar dichas frecuencias de forma que todas ellas sumen 1 con el fin de poder comparar documentos de diferentes tamaños. Finalmente, para poder comparar dos documentos se ha aplicado la similitud del coseno .

Para poder probar esta propuesta se ha hecho uso de una colección de documentos que contienen programas en C++, Java y Python de un Sistema Multiagente, que permiten desarrollar agentes con sus respectivos comportamientos. Para cada lenguaje tenemos una colección de programas que tienen cierta correspondencia con programas en los otros lenguajes. Esta correspondencia puede ser total o parcial en funcionalidad.

3. Experimentación

El objetivo general de este trabajo es proponer un método para detectar similitud entre códigos fuente escritos en distintos lenguajes de programación, es decir multilingüe. Para ello se han realizado los experimentos sobre los lenguajes C, Java y Python con el corpus SPADE[1] de la siguiente forma. Se ha utilizado

[1] El corpus SPADE consiste en tres versiones de una interfaz para diseñar sistemas multiagentes. Estas versiones estan escritas en C, Java y Python y se corresponden en funcionalidad parcialmente.

distintos tipos de comparación como son: el texto entero, el texto sin los comentarios, solamente los comentarios, el texto sin las palabras del lenguaje, las palabras del lenguaje únicamente y el texto sin los comentarios y sin las palabras del lenguaje.

Estos experimentos se han realizado en base a term frequency (tf) y para term frequency-inverse document frequency (tf-idf) para n-gramas de carácteres con valores de N de 1 a 5.

En la mayoría de los experimentos entre los pares de lenguajes de programación mencionados se han obtenido los mejores resultados teniendo en cuenta el texto entero y el texto entero sin los comentarios con los mismos valores en ambos casos. Además los mejores valores de n-gramas de carácteres han sido de tamaño 3 hasta 5. En términos de promedio y desviación de la posición de los documentos correspondientes a los códigos fuente utilizados los mejores resultados para trigramas han sido de 1.00 ± 0.00 entre Java y C++, de 1.44 ± 0.83 entre Python y C++, y de 1.62 ± 1.10 entre Java y Python.

Otro resultado interesante que hemos detectado ha sido el que tf-idf funcionara de manera bastante pareja a tf, dado que para aplicar tf-idf se necesita conocer el corpus de referencia y que hay que realizar un preproceso de éste para obtener la tabla de frecuencias de términos. Esto puede ser debido al tamaño del corpus, por lo que habría que comprobarlo con un corpus mucho más grande del utilizado en este estudio previo.

4. Conclusiones

Después de realizar todos los experimentos, una primera conclusión que se puede extraer es que tf-idf funciona bastante similar a tf. En el futuro se espera volver a realizar los experimentos con un corpus más grande con el fin de comprobar si se producen cambios al respecto. De no ser así, la opción más conveniente será utilizar tf dado que no se necesita ningún corpus de referencia y ahorra tiempo de cálculo.

Los comentarios no parecen relevantes a la hora de determinar la similitud, ya que en la mayoría de casos funciona mejor el texto entero y el texto entero sin los comentarios para valores de 3 o más n-gramas de carácteres. Esto sugiere que por lo que se pueden obviar, ganando tiempo de cálculo y evitando manipulaciones por parte del usuario en los comentarios en el nivel 1 sugerido por Faidhi en [4]. Además, que los mejores resultados se consigan con 3 o más n-gramas de carácteres sugiere que se puede detectar el estilo de programación al igual que sucede en escritura como se comenta en [9].

El hecho de que los resultados sean similares con trigramas que con tetragramas y pentagramas, nos hace pensar que el sistema se ha estabilizado y que no va a mejorar más. Si partimos de este supuesto, convendría trabajar con el tamaño mínimo de n-grama dado que es menos costoso el cálculo de éstos n-gramas.

El presente trabajo se considera como unas pruebas preliminares con la intención de comprobar los métodos y técnicas de similitud de textos que pueden aplicarse a la similitud de código fuente. Se considera que aún quedan muchas más

por evaluar como las ventanas deslizantes [9][2] o como las técnicas CL-Explicit Semantic Analysis [7] y CL-Alignment-based Similarity Analysis [6], utilizadas entre distintos idiomas del lenguaje natural. Por otra parte, se considera que este experimento se ha realizado con un corpus muy pequeño, tanto en cantidad de documentos a comparar como en lenguajes de programación estudiados. En un corto plazo de tiempo se espera poder disponer de un corpus más amplio y que cuente con una mayor variedad de lenguajes de programación.

Agradecimientos Este trabajo ha sido desarrollado con la ayuda del proyecto de investigación MICINN TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 (Plan I+D+i)

Referencias

1. Arwin, C. and Tahaghoghi, S.M.M.: Plagiarism Detection across Programming Languages. Proceedings of the 29th Australasian Computer Science Conference vol. 48, pp. 277-286. Darlinghurst, Australia: Australian Computer Society. (2006)
2. Burrows, S., Tahaghoghi, S.M.M., and Zobel, J.: Efficient plagiarism detection for large code repositories. Software? Practice and Experience, vol. 37. pp. 151-175. (2006)
3. Clough, P.: Plagiarism in natural and programming languages: An overview of current tools and technologies. Research Memoranda: CS-00-05. Department of Computer Science. University of Sheffield, UK. (2000)
4. Faidhi, J. and Robinson, S.: An empirical approach for detecting program similarity and plagiarism within a university programming environment. Comput. Educ., vol. 11, pp. 11-19. (1987)
5. Halstead, M. H.: Natural laws controlling algorithm structure? SIGPLAN Noticies, vol. 7(2). (1972)
6. Pinto, D., Civera, J., Barrón-Cedeño, A., Juan, A., and Rosso, P. A statistical approach to crosslingual natural language tasks. Journal of Algorithms, vol. 64(1), pp. 51-60. (2009)
7. Potthast M., Barrón-Cedeño A., Stein B., Rosso P.: Cross-Language Plagiarism Detection. In: Languages Resources and Evaluation. Special Issue on Plagiarism and Authorship Analysis, vol. 45, num. 1. (2011)
8. Prechelt, L., Malpohl, G., and Philippsen, M.: Finding plagiarisms among a set of programs with jplag. Journal of Universal Computer Science, vol. 8(11), pp. 1016-1038. (2002)
9. Stamatatos, E.: Intrinsic Plagiarism Detection Using Character n-gram Profiles. In Stein et al. vol. 502, pp. 38-46. (2009)
10. Whale, G.: Software metrics and plagiarism detection. Journal of Systems and Software, vol. 13, pp. 131-138. (1990)
11. Wise, M.J.: Detection of similarities in student programs: YAPing may be preferable to Plagueing, SIGSCI Technical Symposium, Kansas City, USA, pp. 268-271. (1992)

[2] La técnica de ventanas deslizantes consiste en dividir los documentos a trozos del mismo tamaño y comparar todas las partes con todas. Así, además de evitar comparar documentos de diferente tamaño, se puede detectar si una parte del código ha sido situada en cualquier parte del otro documento.

Recuperación de información

Actas IV Jornadas TIMM, pp 27-28

Geographic Information Retrieval

Fernando S. Peregrino, David Tomás and Fernando Llopis

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante
Carretera San Vicente del Raspeig s/n - 03690 Alicante (España)

Resumen Nuestra investigación se centra en la rama de la recuperación de información geográfica, concretamente se está investigando sobre las entidades geográficas vagas tales como "el norte de España", "cerca de la costa mediterránea", etc. Hasta la fecha solamente hemos hecho una aproximación a este tema, la cual fue hecha desde un punto de vista geográfico, más que de PLN. Para dicho artículo se desarrolló una aplicación que construía una imagen ráster o bitmap de la entidad geográfica concisa que aparecía en una consulta. Sobre esta imagen se asignaban valores a cada una de las cuadrículas que la formaban dando un mayor valor a aquellas que se encontraban más próximas a la entidad vaga que en la consulta se solicitaba. A modo de ejemplo, en el artículo se mostraron consultas del estilo de "Estaciones de esquí en el norte de España", de la cual, el sistema obtenía la imagen raster de España (la entidad concisa) y daba un mayor peso a las cuadrículas que estaban en el norte (la expresión vaga) de esta entidad concisa. En definitiva, centrándonos en el ejemplo anterior, la imagen ráster nos permitía ponderar la consulta realizada (Estaciones de esquí) mediante la entrada como parámetro de las coordenadas geográficas de cualquier estación de esquí del mundo, discriminando fácilmente el sistema aquellas que estaban en el norte de España del resto, incluso las que estaban muy próximas en la frontera de la entidad difusa como podían ser las estaciones de Andorra, Pirineos franceses o incluso las del sistema central español. Tras este enfoque pretendemos abordar la investigación de las entidades geográficas vagas mediante la captación de información en lenguaje natural en Internet, concretamente en páginas de agencias de viaje, turismo, opinión, redes sociales, tales como Booking, TripAdvisor, Flickr etc., y con ayuda de la Wikipedia.

Por otra parte, si lo expuesto anteriormente es el trabajo que hemos realizando y en el que nos vamos a centrar en un futuro próximo, hay que hablar de lo que actualmente estamos preparando, un sistema con el que vamos a participar en el Workshop del NTCIR, en la rama de GeoTime (recuperación de información geográfica y temporal), para lo cual se cuenta con un sistema de recuperación de información temporal (TIPSem) desarrollado en nuestro grupo, y se está implementando la parte geográfica mediante Lucene y el API de Yahoo! PlaceMaker. Por último, y como objetivo final, lo que se pretende es implantar un sistema GIR (Geographic Information Retrieval) acoplando la parte de recuperación geográfica general que vamos a presentar en el NTCIR y la específica de la entidades difusas, así como cualquier otro problema relacionado con la recuperación de información geográfica.

Keywords: Geographic Information Retrieval, Fuzzy Entities, Vernacular Language.

Actas IV Jornadas TIMM, pp 29-38

UBC at Slot Filling TAC-KBP 2010

Ander Intxaurrondo, Oier Lopez de Lacalle, Eneko Agirre
aintxaurrond001@ikasle.ehu.es, oier.lopezdelacalle@ehu.es,
e.agirre@ehu.es

IXA NLP Group, University of the Basque Country, Donostia, Basque Country

Abstract. This paper describes our submissions for the slot filling and *surprise* slot filling tasks of TAC-KBP. The system is based on the distant supervision strategy presented by [3]. We did a straightforward implementation, trained using snippets of the document collection containing both entity and filler from the KB provided by the organizers (a subset of Wikipedia infoboxes). Our system does not use any other external knowledge source, with the exception of closed lists of words for religion, causes of death, charges and religious/political affiliation, plus the use of Geonames to distinguish between cities, countries, states and provinces. We submitted three runs based on different post-processing options of the output of our classifiers, with results below the median. We did expect low results, as our system is still under development, and we still have plenty of room for improvement.

1 Introduction

This paper describes our participation in the TAC-KBP 2010 slot-filling and surprise slot-filling tasks. Our system is a straightforward implementation of a distant supervision system [3]. The system was trained using snippets of the document collection containing both entity and filler from the KB provided by the organizers (a subset of Wikipedia infoboxes). Our system does not use any other external knowledge source, with the exception of closed lists of words for religion, causes of death, charges and religious/political affiliation, plus the use of Geonames to distinguish between cities, countries, states and provinces.

The paper is structured as follows. In Section 2 the tasks of *slot filling* and *surprise slot filling* will be described. In Section 3 the main components for the distant supervision system will be explained, including slot preparation, extraction of training examples, classifiers and the post-processing to produce the output. Next, we will focus on the results obtained by the three runs for slot filling, and for the unique run for surprise slot filling. Section 5 is devoted to error analysis, and finally, in Section 6, we draw some conclusions.

2 Slot Filling

The *slot filling task* in TAC-KBP consists on learning a set of predefined relationships and attributes for named entities (people or organizations) based on a pre-existing

knowledge base extracted from Wikipedia Infoboxes. The learned information is then used to extract new facts from a large document base (1,7 million documents) for a set of target entities. The main objective is thus to feed Wikipedia Infoboxes with new additional values extracted from the document collection.

When we developed this system, in 2010, the TAC-KBP track was on its second edition.

The information in the KB is organized around *entity-slot-filler* triples. An entity is the name of the article of Wikipedia, and can include people or organizations. The slot is the type of information of the entity, for example the birthplace of a person. The filler is the value of the slot. An example of an *entity-slot-filler* triple could be *Paul Newman - date of birth - January 26, 1925*. The target slots were defined by the organizers, including which are possible values, as made explicit in the task guidelines.

2.1 Surprise Slot Filling

The surprise task was optional in TAC-KBP 2010. In this task, participants had to find information for new entities and new slots, specified by TAC-KBP organizers. The main idea of the task consisted on giving portability to the information extraction systems developed by participants.

3 Distant supervision system

We tried a straightforward strategy for slot filling, designed around distant supervision [3] and joint work by Stanford and UBC in TAC-KBP 2009 [1]. Our system is very similar to the later, with the difference that we used freely available tools and that our system is still under development.

Our systems has a training phase and an application (or test) phase. For training we perform the following steps:

- Slot preparation, including the extraction of entity-slot-filler triples from infoboxes, mapping them to *official* KB slots, and assigning a named-entity type or a closed list depending on the expected fillers.
- Example extraction, where we retrieve text fragments which include both the entity and filler in the triples
- Training of classifiers using the extracted examples

When applying the system we perform the following steps:

- Search of examples of mentions to the target entities
- Identification of potential fillers for possible slots
- Applying the classifiers to each filler in each mention
- Collation of results, where for each entity and slot the system returns the filler with maximum weight from classifiers[1]. When no filler is classified positively, the system returns NIL.

[1] We tried slightly different post-processing strategies in the three submissions (cf. Section 4) following this idea.

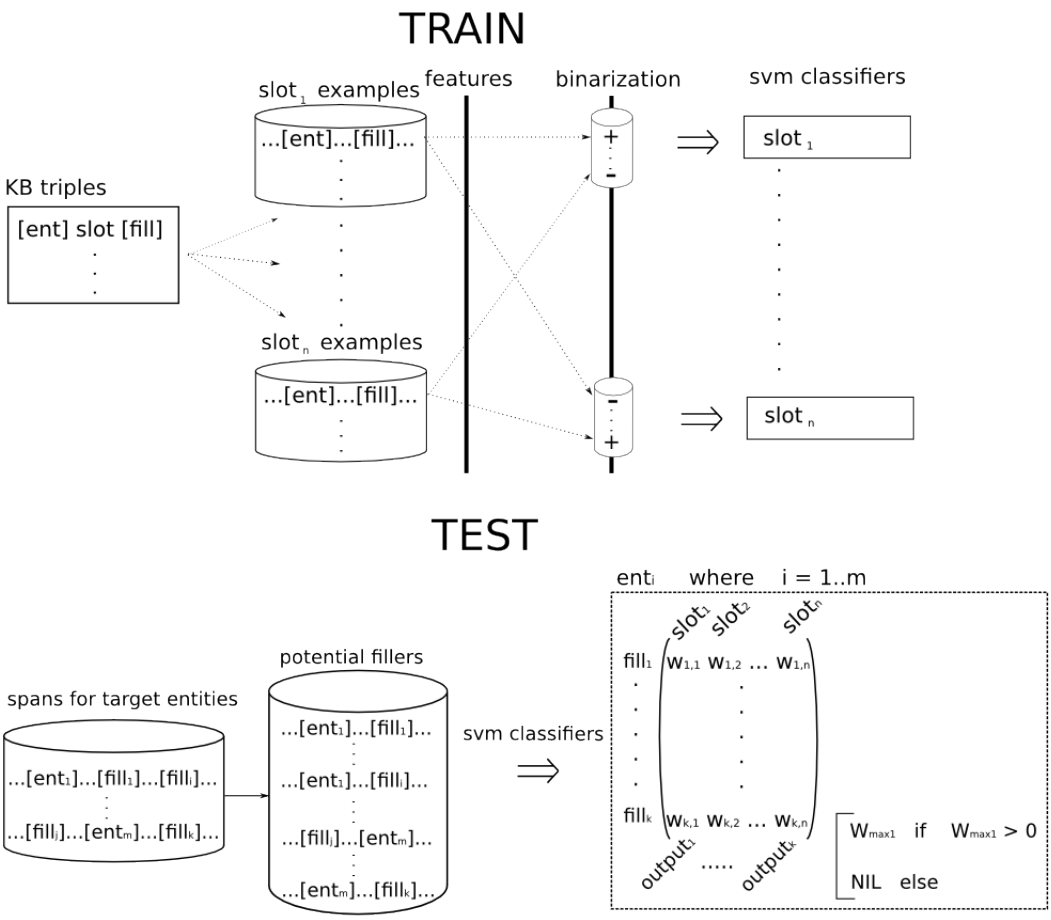


Fig. 1. The architecture of the slot filling system. TRAIN: Extraction of KB triples, which are used to acquire training examples for each slot ($1..n$ slots) from the document base, followed by featurization and binarization. We finally train n classifiers, one per slot. TEST: examples containing mentions to the target entities (m entities) are retrieved from the document base (m target entities). Potential fillers are identified, and then each example containing one entity-filler is classified, obtaining a weighted prediction for each slot. Predictions are collated and the result returned.

For the *Surprise Slot Filling* task, the organizers did not provide any training triples from Wikipedia infoboxes, and training examples were directly provided. Given the very small number of examples provided, we looked for additional examples containing those entity-filler pairs in the document base.

The development of the system did not involve manual curation of data, except assigning named entity classes (e.g., date, person) or closed lists of fillers (e.g., religions, countries, products, diseases) to each slot.

Below, we first present the details of how we prepared the slot information, then how we extracted the textual fragments (examples) of entity occurrences, followed by the method to train the classifiers. The application of the classifier to produce the slot filling results is explained next.

3.1 Slot Preparation

In order to prepare the training data for the slot classifiers, we first extracted entity-slot-filler triples from Wikipedia infoboxes using the mapping provided by the organizers.

As part of slot preparation, different slots based on the expected NE type were categorized (see Table 1: ORG, PER, LOC, DATE, and NUMBER. The NE type is used to help assign ambiguous infobox values to the appropriate slot, as well as to identify potential fillers for a text fragment for a slot. For org:website, regular expressions were used; nothing was done for per:title.

In the *Surprise Slot Filling* task, closed lists of fillers were used for all new slots.

NE (ORG)	org:alternate_names, org:founded_by, org:member_of, org:members, org:parents, org:shareholders, org:subsidiaries, per:employee_of, per:member_of, per:schools_attended
NE (PER)	org:founded_by, org:shareholders, org:top_members/employees, per:alternate_names, per:children, per:other_family, per:parents, per:siblings, per:spouse
NE (LOC)	org:headquarters, per:place_of_birth, per:place_of_death, per:residences
NE (DATE)	org:dissolved, org:founded, per:date_of_birth, per:date_of_death
NE (NUMBER)	org:number_of_employees/members, per:age
Closed List	org:political/religious_affliation, per:cause_of_death, per:charges, per:origin, per:religion
RegExp	org:website
NIL	per:title

Table 1. Mapping of slot to NE type or closed list. *LOC* slots belong to the 2009 TAC-KBP Slot Filling task, these slots will be later adaptted to fit with the 2010 task (see section 3.4)

Due to the ambiguity and noisiness of the infobox to slot mapping, we processed the infobox values for the entity-slot-filler triple as follows:

– We run a named-entity recognition and classification system [2] on the entity itself to determine if the entity is ORG/PER. Because the slots are specific for the two entity types, we can safely ignore any entity that is not ORG/PER. Besides, note that some entities in the knowledge base have been tagged as UNK by the organizers (instead of ORG/PER). We also run NER on this to determine the entity type.

– Run NER on infobox fillers to extract fillers for ambiguous slots. The mapping from the Wikipedia infobox to the TAC-KBP slots can be ambiguous. For instance, the Wikipedia infobox "born" can map to `date_of_birth`, `country_of_birth`, `stateorprovince_of_birth` and `city_of_birth`:

  Carrie Underwood
  Born March 10, 1983 (1983-03-10) (age 6) Muskogee, Oklahoma, USA

After obtaining the entity-slot-filler triples, we extract examples from the document base for training and development.

3.2 Example Extraction

The training examples were drawn from the 2009's TAC KBP Entity Linking Sample Corpus. Due to time limitation we were not able to build a training set based on the 2010 document base. We indexed the document base using the *KBP_Toolkit* search tool provided by NIST, which had Lucene on its base.

In order to extract the training examples, we used the known entity and filler pairs, and looked for occurrences of these in the document base. Exact string match is used for both the entities and fillers. We looked for examples with up to 10 tokens between the entity and filler, and five words to surrounding the entity and filler. The examples are of the form:

```
5w entity 0-10w filler 5w
5w filler 0-10w entity 5w
```

where Nw corresponds to N words/tokens; for the middle span, this ranged from zero to ten.

Note that because we look for exact matches for the entity and filler, we miss examples that contain variations of the entity or filler strings.

For target entities for slot filling, we extracted examples that matched the string of the entity exactly. These examples are of the form:

```
30w entity 30w
```

Similarly, we miss examples that use different names for the target entity.

3.3 Training the Classifiers

For each slot, we trained a binary classifier that takes a text fragment with the entity and potential filler and decides whether or not the potential filler is an actual filler for the slot. We used Support Vector Machines (SVM) trained on the entity-slot-filler examples extracted from the document base (cf. Section 3.2). As explained in the previous section, the development of the system was done using the TAC-KBP 2009 dataset. Basically, our development consisted of feature set selection and setting of the SVM cost parameter (C).

For positive examples, we used examples containing the known entity and filler pairs based on slots derived from Wikipedia infoboxes. To avoid misleading infoboxes, we only used examples that had an entity type matching the entity type of the slot.

We did not use the Participant Annotation samples as positive examples due to time problems.

For negative examples, we distinguish between persons and organizations. For instance, given a specific classifier of slot i for person entity, the rest of the person slots were considered as negative examples. We followed the same strategy for slots of organization entities.

Regarding learning features, we considered three sets of features in order to develop the final system. The first set was based on the features introduced by Mintz et al. [3]. The second set was based on the features proposed by Zhou et al. in [4]. Finally, the third feature set was build by joining the previous two sets in one.

This way, for the first set we extracted the following feature types:

- The sequence of words between the entity and filler (10 words maximum).
- The part-of-speech tags of these words.
- The name-entity types of the entity and filler.
- A window of *k* words to the left of the first entity/filler and their part-of-speech tags
- A window of *k* words to the right of the second entity/filler and their part-of-speech tags.

Each lexical feature consists of a conjunction of all this components. We generate a conjunctive feature for each $k \in \{0, 1, 2\}$. Thus, Table 2 shows the resulting lexical feature (note that the each row in the table represents a single lexical feature).

ENTITY - SLOT - FILLER : Kim Il-Sung - `date_of_birth` - April 15, 1912				
SPAN: *Kim Jong-Il's late father <entity> Kim Il-Sung </entity> , who was born <filler> April 15, 1912 </filler>.*				
LEFT WINDOW	NE1	MIDDLE	NE2	RIGHT WINDOW
[]	PERSON	,/, who/WP was/VB born/VB	DATE	[]
[father/NN]	PERSON	,/, who/WP was/VB born/VB	DATE	[]
[late/JJ father/NN]	PERSON	,/, who/WP was/VB born/VB	DATE	[]
		...		

Table 2. Result of the conjuntive lexical features.

For the second-type features (those based on [4]), we extracted the following feature types:

- A flag indicating there is no word between the entity and filler.
- A flag indicating there is only one word between the entity and filler.
- The first word after the first-coming entity/filler.
- The last word before the second-coming entity/filler.
- All words between the entity and filler, except the first and last.
- The first word before the first-coming entity/filler.
- The second word before the first-coming entity/filler.
- The first word after the second-coming entity/filler.

– The second word after the second-coming entity/filler.
– The name-entities of the entity and filler.

Finally, the third feature set contained the features from both [3] and [4]. Among all three features set, the best results were obtained deploying the first set (those introduced in [3]). We think that this is because the conjunction of features yields high-precision features (but low-recall). With a small amount of data, this approach would be problematic, since most features would only be seen once, rendering them useless to the classifier. Since we use large amounts of data, even complex features appear multiple times, allowing our high precision features to work as intended. For Surprise Slot-Filling task we use the second set of features [4]. Due to the lack of training data, it is unlikely that complex features occur enough times for learning. So that we would expect higher recall by the use of independent features.

We used *svmperf*[2], which is an extension of *svmlight* to manage large sets of data, as implementation of a linear SVM classifier. We tried different values of C to tune the classifiers. The used values of C were 0.01, 1, 10, 20, 50, 100 and 200. We obtained the best results with $C = 10$ in the development dataset (cf. Section 3.2).

3.4 Applying the classifiers

Once the classifiers was trained, we used them to determine the most likely fillers for the target entities. Using the examples extracted from the document base for each entity, we identified potential fillers using a NER module or closed lists of strings (see Table 1). After identifying potential fillers within the span, we expanded the examples for target entities in entity-filler pairs (see Figure 1, test part). For each entity-filler pair extraction of features was carried out, and the prediction of the classifier in the slot was obtained deciding whether the filler was positive or negative.

For each entity-slot, we selected the positive the top-scoring filler for single-valued slots. Depending on the run (cf. Section 4) for multi-valued slot we returned the list of all positive fillers. If a slot had all the fillers with negative predictions, the system would return a NIL value for that slot (see Figure 1, "Output" part).

In the 2009 edition of TAC-KBP, slots like `place_of_birth`, `place_of_death`, `residences` and some others relation with locations were used. In 2010, this slots were separated into 3 parts; for example, instead of `place_of_birth` we got the following slots: `country_of_birth`, `stateorprovince_of_birth` and `city_of_birth`. We distinguish between countries, states and cities, after applying the classifiers . We used the GeoNames geographical database[3] to determine if the filler value was a city, country, state or province; also taking control if the filler value contains, for example, a country and a city.

In cases like the *Carrie Underwood* example mentioned before, our system will determine that "March 10, 1983" is a `DATE` while "Muskogee", "Oklahoma" and "USA" are `LOC`. We then map "March 10, 1983" to the `date_of_birth` slot "Muskogee" to the `city_of_birth` slot, "Oklahoma" to the `stateorprovince_of_birth` slot, and "USA" to the `country_of_birth` slot.

[2] `http://www.cs.cornell.edu/People/tj/svm_light/svm_perf.html`
[3] `http://www.geonames.org`

4 Results

Due to the limited time we had to build the entire slot-filling system, we were not able to tune our system. We submitted three runs based on different post-processing of the output of the classifiers. For the first run (UBC1), we submitted a basic system which, for each entity and slot, takes the filler that maximizes the prediction of the slot classifier. The system returns NIL if the SVM prediction is negative for all potential fillers within a slot.

In the second run (UBC2), for each entity, if its slot is single-valued we return the potential filler that maximizes the prediction of the classifier, as we did in the first run. But if the slot type is multi-valued, the system returns all positive potential fillers. For this submission we removed the slots that had to be ignored, as specified by the organization.

Finally, in the last run (UBC3), for each entity, we run all the slots looking for the one which maximizes the entity-slot-filler triple (as in UBC1 and UBC2). Depending on the type of the slot, it is treated differently. Given an entity, for each we first check if the slot is single-valued. In that case, we select the filler which maximize the slot. In the case that the slot is a location slot (org:headquarters, per:place_of_birth, per:place_of_death, per:residences), we select up to three countries, cities or states above a threshold of prediction confidence given by the classifiers. For the rest of the multiple valued slot we return the filler that maximizes the triple, in the same way we did for unique valued. Based on some preliminary results on the development dataset, we set the confidence threshold in -0.8 in order to increase the recall of the system. Again, we ignored some entity-slots, as specified by the organization.

	MAIN TASK				SURPRISE TASK	
	UBC1	UBC2	UBC3	median	UBC	median
# filled slots in key	1034	1034	1034		505	
# filled slots in response	37	6398	109		3	
# correct non-NIL	1	3	5		1	
# incorrect/spurious	35	6380	103		2	
# inexact	0	9	0		0	
Recall	0.0009	0.0029	0.0048	0.1412	0.0019	0.1544
Precision	0.0270	0.0004	0.0458	0.2141	0.3334	0.5032
F1	0.0018	0.0008	0.0087	0.1054	0.0039	0.2363

Table 3. TAC 2010 KBP Slot Filling Results.

Table 3 shows in the first three columns the official results in TAC 2010 KBP Slot Filling task, followed by the median. Although all the runs are very low, they show that the more sophisticate is the post-processing the more accurate are the results.

The last columns show the results for the Surprise slot-filling task. For this subtask we post-process the output of the classifiers as we did for UBC1: We returned for each entity and slot the filler with the maximum weight given by the SVM classifier.

5 Analysis

Although we were expecting low results, the obtained results are far from satisfactory. This lead us to analyze the outputs of our system. We next list some issues, and their possible solutions.

- **Lack of positive examples**. There were some slots without no positive examples in the training set.
- **Noisy positive examples**. Many of the gathered training examples were inaccurate for appropriate automatic learning. This means that we should apply some kind of filtering or instance weighting technique to get rid of useless examples.
- **Negative examples**. We generated too many negative example producing an unbalanced training set. Unbalanced training sets introduce undesirable biases in the learning process. Smart filtering of negative examples or weighted SVM classifiers might be a desirable solution to the problem. In Table 4 is possible to compare the number of tuples, positive spans and negative spans between slots.
- **Post-processing**. We treated the output of the classifiers equally. In other words, we did not take into account that each slot would need to tune its own threshold independently. This caused the system to select too many fillers for some slots like `per:title`. In addition, the post-processing phase could be improved by using semantic classes to constrain the final output of our system.
- **Suprise slot filling task**. The slots provided for the *surprise slot filling* task did not appear in Wikipedia infoboxes, so we could not apply our distant supervision strategy. On the other hand, the training provided by the organizers data was too small to train our classifiers.

slot	triples	pos. examples	neg. examples	slot	triples	pos. examples	neg. examples
per:age	54	131	99790	org:alternate_names	1121	6690	182235
per:alternate_names	590	1755	98166	org:dissolved	227	1486	187439
per:cause_of_death	0	0	99921	org:founded_by	253	1225	187700
per:charges	10	25	99896	org:founded	1243	5379	183546
per:children	157	598	99323	org:headquarters*	13352	93277	95648
per:date_of_birth	146	249	99672	org:member_of	665	3924	185001
per:date_of_death	83	162	99759	org:members	137	855	188070
per:employee_of	1676	13871	86050	org:number_of_employees/members	422	2843	186082
per:member_of	2623	18440	81481	org:parents	4304	31233	157692
per:origin	207	1529	98392	org:political/religious_affiliation	1145	7268	181657
per:other_family	11	13	99908	org:shareholders	0	0	188925
per:parents	11	77	99844	org:subsidiaries	87	348	188577
per:place_of_birth*	4613	24426	75495	org:top_members/employees	6109	31502	157423
per:place_of_death*	1125	5554	94367	org:website	1265	2894	186031
per:religion	223	857	99064				
per:residences*	2835	17256	82665				
per:schools_attended	83	156	99765				
per:siblings	6	10	99911				
per:spouse	776	3391	96530				
per:title	2302	11416	88505				

Table 4. Statistics for all slots, including number of triples, positive and negative examples. The slots with an asterisk (*) belong to the 2009 TAC-KBP Slot Filling track, before separating them to cities, states, provinces or countries.

6 Conclusions

We have participated with a preliminary implementation of a distant supervision system. The idea was to train the system using snippets of the document collection containing both entity and filler from the KB provided by the organizers (a subset of Wikipedia infoboxes). Our system does not use any other external knowledge source, with the exception of closed lists of words for religion, causes of death, charges and religious/political affiliation, plus the use of Geonames.

Our main goal was to setup a preliminary system, and we submitted three runs based on different post-processing options of the output of our classifiers, with results below the median.

The low results of our system in the main slot filling task, although expected, are far from satisfactory. The core Information Extraction module of our system is preliminary and buggy, with a few unsolved issues. One of the lessons that we have learned is that we first need to develop a traditional Information Extraction system and evaluate it on standard datasets (e.g. ACE relation detection task). We would then explore the challenges posed by the slot filling task proper, which include issues like getting false positive examples for training, or treating each slot as a separate problem.

Regarding the surprise slot filling exercise, the organizers released a few training examples for each target slot, where the examples where snippets of text where the entity, slot and filler were explicitly attested. Given the spirit of the main task (where a large number of entity-slot-filler triples from the KB had been made available), we were expecting such entity-slot-filler examples for the surprise task as well. This might partially explain our low results.

References

1. Agirre, E., Chang, A.X., Jurafsky, D.S., Manning, C.D., Spitkovsky, V.I., Yeh, E.: Stanford-ubc at tac-kbp. In: Proceedings of the Second Text Analysis Conference (TAC 2009). Gaithersburg, Maryland, USA (November 2009), pubs/subctackbp.pdf
2. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). pp. 363–370. Association for Computational Linguistics, Ann Arbor, Michigan (June 2005), http://www.aclweb.org/anthology/P05-1045
3. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of ACL-IJCNLP 2009 (2009), http://www.stanford.edu/~jurafsky/mintz.pdf
4. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 427–434. Association for Computational Linguistics, Morristown, NJ, USA (2005)

Minería de opiniones y análisis de sentimientos

Actas IV Jornadas TIMM, pp 41-44

Extracción de opiniones sobre características adaptable al dominio *

Fermín L. Cruz, José A. Troyano, F. Javier Ortega and Fernando Enríquez

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Av. Reina Mercedes s/n 41012, Sevilla (Spain)
{fcruz,troyano,javierortega,fenros}@us.es

Resumen La extracción de opiniones sobre características es una tarea relacionada con la minería de opiniones, que consiste en extraer a partir de textos opiniones individuales acerca de las características de un objeto determinado. En los últimos años, esta tarea ha sido abordada desde una perspectiva no supervisada y sin concretar un dominio de aplicación específico. Nuestra propuesta, sin embargo, se centra en el desarrollo de un sistema de extracción que tenga en cuenta las particularidades de cada dominio de aplicación, y que se adapte con facilidad a los distintos dominios mediante la definición de una serie de recursos específicos. Los experimentos realizados muestran que el conocimiento aportado por estos recursos supone una valiosa ayuda para la construcción de sistemas precisos de extracción de opiniones.

1. Introducción

Durante los últimos años, la extracción de opiniones sobre características de productos ha sido estudiada en varios trabajos. La primera definición del problema se encuentra en [4]:

> Given a set of customer reviews of a particular product, the task involves three subtasks: (1) identifying features of the product that customers have expressed their opinions on (called product features); (2) for each feature, identifying review sentences that give positive or negative opinions; and (3) producing a summary using the discovered information.

Esta definición ha sido la base de distintos trabajos de dichos autores y otros investigadores [5],[8],[7],[10],[3]. En todos estos trabajos se aborda la tarea desde una perspectiva general, sin concretar ningún dominio de aplicación. Los sistemas tratan de identificar las menciones a características de los productos y las palabras que expresan la opinión de los usuarios sin tener en cuenta las particularidades del producto. Además, estos trabajos se apoyan en su mayoría

* Parcialmente financiado por el Ministerio de Educación y Ciencia (HUM2007-66607-C04-04).

en métodos automáticos, sin hacer apenas uso de ningún recurso manual que aporte conocimiento de calidad al sistema de extracción.

En contraposición a estos trabajos, nuestra propuesta se basa en un acercamiento más aplicado al problema: (1) antes de construir el sistema, se debe concretar un dominio de aplicación; (2) sólo se tendrán en cuenta las características incluidas en una taxonomía específica; y (3) el sistema de extracción se apoyará en un conjunto de recursos específicos del dominio de aplicación, generados automáticamente a partir de un conjunto de documentos anotados.

2. Nuestra propuesta

En nuestro trabajo, hemos partido de una definición distinta de la tarea de extracción de opiniones sobre características: dado un conjunto de documentos de opinión de un dominio concreto, se trata de reconocer las opiniones vertidas acerca de una serie de características opinables disponibles para el dominio, y clasificar las opiniones reconocidas según la polaridad (positivas/negativas). Llamamos a la primera tarea *reconocimiento de opiniones* y a la segunda *clasificación de opiniones*. En nuestro planteamiento es fundamental la participación como entrada al sistema de las características en las que estamos interesados. Dichas características son descritas en una taxonomía, en la que además se dispone de relaciones de especialización/generalización entre las mismas. Por ejemplo, la característica *sound quality* para el dominio *headphones* puede descomponerse a su vez en varias características, por ejemplo *bass*, *mids* y *highs*. La construcción de la taxonomía de características es un paso previo que es llevado a cabo de manera semiautomática, utilizando un algoritmo de *bootstrapping* a partir de un conjunto de documentos de opinión del dominio y con la participación de un experto.

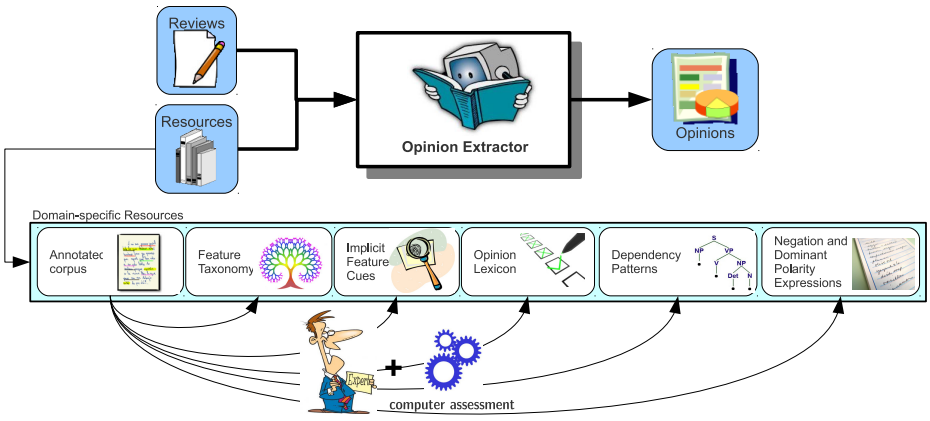


Figura 1. Esquema conceptual de nuestra propuesta

Uno de los pilares de nuestra propuesta es la generación de recursos dependientes del dominio, que facilitan la tarea de extracción de las opiniones. Dichos recursos incluyen, entre otras cosas, patrones de dependencias sintácticas que permiten conectar las *palabras de característica* (menciones a las características de la taxonomía) con las *palabras de opinión* (palabras a partir de las cuáles se decide la polaridad de una opinión), lexicones de palabras de opinión del dominio (incluyendo estimaciones de la orientación semántica de las mismas y de la probabilidad de ser utilizadas en una opinión) y listas de términos indicativos de características implícitas (una característica implícita es aquella que no aparece mencionada en el texto, sino que se deduce de las palabras de opinión utilizadas, como ocurre por ejemplo con la palabra de opinión *expensive*, que asociaremos a la característica *price*). Los recursos son inducidos a partir de un conjunto de documentos anotados. Dado que el proceso de anotación de las opiniones puede ser costoso, hemos investigado métodos para facilitar dicho proceso, así como algoritmos de ampliación automática que nos permiten construir los recursos a partir de unos pocos documentos anotados y un conjunto mayor de documentos sin anotar. Además, en la construcción del sistema extractor hemos incluido implementaciones independientes del dominio (y que por tanto no hacen uso de los recuros) de algunas de las subtareas identificadas. De esta forma pretendemos, por un lado, permitir la construcción rápida de sistemas para nuevos dominios y, por otro lado, evaluar la aportación real de los recursos a la resolución de la tarea. En [2] se describen con mayor detalle los recursos, los métodos utilizados para su generación y la arquitectura del sistema extractor.

3. Experimentación

En este apartado incluimos algunos resultados experimentales realizados sobre un dominio concreto (*headphones*), con una taxonomía de 31 características. Para su realización, dispusimos de un corpus de 587 documentos anotados de análisis (*reviews*) de auriculares, en inglés, extraídos de Epinions.com. El corpus utilizado, incluyendo también los dominios *hotels* y *cars*, está disponible para uso público[1]. Los experimentos se realizaron usando validación cruzada sobre diez particiones, utilizando en cada una de las ejecuciones una parte para evaluación y el resto para la generación de los recursos.

En la tabla 1 mostramos los resultados obtenidos por cinco aproximaciones distintas. Las tres primeras no hacen uso de los recursos del dominio; en su lugar, usan un método léxico basado en ventanas para enlazar las palabras de característica y de opinión, y clasifican la polaridad de las opiniones utilizando distintos algoritmos de la literatura : información mutua entre términos de opinión y semillas (algoritmo PMI-IR) [9], cálculo de distancias en WordNet [6], y el recurso léxico SentiWordNet [1]. El cuarto sistema se basa en los recursos del dominio para llevar a cabo la extracción. Finalmente, el quinto sistema representa un acercamiento híbrido en el que se utilizan componentes basados en recursos y algunos independientes del dominio. Como se puede observar, los

[1] http://www.lsi.us.es/ fermin/index.php/Datasets

sistemas que hacen uso de los recursos mejoran significativamente los resultados de los sistemas que no hacen uso de ellos.

	Opinion Recognition			Opinion Classification	Opinion Recognition + Classification		
Experimento	p	r	$F_{\frac{1}{2}}$	accuracy	p	r	$F_{\frac{1}{2}}$
PMI-IR	0,6092	0,3039	0,5073	0,8706	0,5512	0,2754	0,4593
WordNet	0,6756	0,3002	0,5405	0,8940	0,6111	0,2720	0,4892
SentiWordnet	0,6744	0,3643	0,5763	0,8688	0,5972	0,3230	0,5105
Resource-based	0,7869	0,5662	0,7300	0,9503	0,7557	0,5436	0,7010
Hybrid	0,7836	0,5736	**0,7301**	**0,9572**	0,7573	0,5543	**0,7056**

Cuadro 1. Resultados obtenidos por los *pipelines* basados en recursos

Referencias

1. S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).
2. F. L. Cruz, J. A. Troyano, F. Enríquez, J. Ortega, and C. G.Vallejo. A knowledge-rich approach to feature-based opinion extraction from product reviews. In *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*, pages 13–20. ACM, 2010.
3. X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 231–240, New York, NY, USA, 2008. ACM.
4. M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.
5. M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of AAAI*, pages 755–760, 2004.
6. J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke. Using wordnet to measure semantic orientation of adjectives. In *National Institute for*, volume 26, pages 1115–1118, 2004.
7. B. Liu, M. Hu, and J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of WWW*, 2005.
8. A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.
9. P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 417–424, 2002.
10. L. Zhuang, F. Jing, X. Zhu, and L. Zhang. Movie review mining and summarization. In *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, 2006.

Sistemas de Recomendación basados en Lenguaje Natural: opiniones vs. valoraciones

John A. Roberto[1], Ma. Antònia Martí[1], Paolo Rosso[2]
[1] Dpto. de Lingüística, Universidad de Barcelona,
Gran Via de les Corts Catalanes, 585. 08007 Barcelona, España
[2] NLE Lab. - ELiRF, DSIC, Universidad Politécnica de Valencia
{roberto.john, amarti}@ub.edu, prosso@dsic.upv.es

Resumen. La construcción de los perfiles de usuario es una de las fases más críticas en el desarrollo de los Sistemas de Recomendación. Actualmente, la mayoría de estos sistemas construyen los perfiles de usuario siguiendo modelos explícitos o implícitos de adquisición de datos. En oposición a estos modelos clásicos, nuestra investigación se centrará en la construcción de perfiles a partir de las opiniones de los usuarios expresadas en lenguaje natural.

Palabras Clave: Sistemas de Recomendación, perfiles de usuario, PLN, opiniones de usuario.

Descripción del trabajo

Los Sistemas de Recomendación (SR) son un tipo particular de aplicaciones y de técnicas especialmente desarrolladas para filtrar la información. Los SR trabajan efectuando predicciones sobre un ítem o conjunto de ítems que podrían ser de interés para un usuario particular. Estos sistemas están enfocados a la eliminación de ítems irrelevantes del flujo de datos. Los SR basan su funcionamiento en el conocimiento que tienen sobre las preferencias de los usuarios y que es almacenado en los perfiles de usuario. La construcción de los perfiles de usuario es una de las fases más críticas del desarrollo de los SR y ahí centraremos nuestra investigación.

En la actualidad, la mayoría de los SR construyen los perfiles de usuario según dos modelos [1] [6] [5]: preguntando directamente a las personas cuales son sus preferencias (modelo explícito) o infiriéndolo de sus acciones (modelo implícito). La ponderación de un ítem en base a una escala ordinal o cualitativa es típica del primer modelo mientras que la selección de un ítem o el tiempo invertido es su visualización, es característica del segundo.

En general, la lógica que gobierna ambos modelos es muy elemental y por lo tanto impide adquirir información compleja. Los modelos explícitos exigen al usuario un esfuerzo cognitivo importante [3] pues, dependiendo el dominio, las personas pueden no estar cualificadas para valorar un producto. Los modelos implícitos, por su parte, tienen limitaciones a la hora de interpretar algunas conductas ambiguas [5]; esto hace

que no sean bien recibidos por los usuarios quienes muchas veces no comprenden el motivo de la recomendación.

En oposición a los modelos clásicos, nuestra investigación se centrará en la construcción de perfiles a partir de las opiniones en lenguaje natural. El empleo del lenguaje natural para la creación de perfiles permite adquirir información compleja [11] [12] [9], que va más allá de la simple ponderación o la selección de un ítem. Adicionalmente, esta técnica mejora la forma en que el usuario se comunica con el sistema puesto que las opiniones, a diferencia de las puntuaciones, no exigen un conocimiento en profundidad de los ítems. El resultado es, pues, una reducción significativa del esfuerzo por parte del usuario y el potencial incremento en la calidad de los datos.

En tareas de recomendación la información lingüística se ha empleado básicamente para la representación del contenido textual de los ítems pero solo en contados casos se ha hecho servir a modo de estrategia de retroalimentación. Los MLD[1] [7] [10], por ejemplo, permiten que los usuarios expresen las valoraciones con palabras ("bonito", "malo", etc.) en lugar de valores numéricos y los SR Conversacionales (SRC) [4] formulan preguntas que los usuarios deben responder ordenadamente para ir refinando la recomendación de forma gradual. Una limitación importante de estos sistemas es que no trabajan con texto libre y para que funcionen tienen que imponer algún tipo de restricción sobre el lenguaje que utiliza el usuario. Nuestra investigación pretende arrojar luz en este sentido, ofreciendo alternativas al tratamiento de texto libre para crear y mantener los perfiles de usuario.

Para crear los perfiles de usuario a partir de opiniones en lenguaje natural utilizaremos estereotipos[2] basados en la forma en que diferentes grupos de usuarios tienen de opinar. Un estereotipo, según nuestro modelo, se compone de un conjunto de características lingüísticas que evidencian la forma de opinar de un grupo y las preferencias asociadas al grupo. De esta manera el proceso de recomendación consiste en hacer que el usuario activo[3] herede las preferencias de los usuarios que se expresen de modo similar.

Proponemos dos métodos alternativos para la obtención de los rasgos lingüísticos que definirán cada estereotipo. El primero radica en el análisis lingüístico de un conjunto significativo de críticas de productos agrupados según las puntuaciones asignadas por los usuarios para dichos productos. La fuente de datos que emplearemos será HOpinion (http://clic.ub.edu), un corpus en castellano de críticas de hoteles recuperadas de TripAdvisor[4]. El corpus tiene un total de 4750 textos (cerca de un millón de palabras) de críticas de hoteles, con una medida promedio de 135

[1] SR basados en Modelos Lingüísticos Difusos.

[2] Los estereotipos son mecanismos empleados para efectuar descripciones parciales de situaciones que se suceden frecuentemente. En los SR los estereotipos se emplean para asignar un usuario a un grupo de usuarios similares del que hereda sus preferencias.

[3] Usuario que es objeto del proceso de recomendación actual.

[4] TripAdvisor es la mayor comunidad de viajeros *on line* del mundo y cuentan con más de 20 millones de opiniones sobre establecimientos y destinos basadas en la experiencia personal de sus usuarios.

palabras por crítica. La valoración de las críticas va en una escala del 10 al 50 en la siguiente relación: 10 / pésimo, 20 / malo, 30 / normal, 40 / muy bueno y 50 / excelente. HOpinion ha sido anotado con información morfosintáctica y corregido manualmente. Además, se le ha aplicado un *chunking* para identificar los constituyentes básicos.

El análisis lingüístico de estas críticas busca caracterizar los usuarios por su forma de utilizar el lenguaje, de manera que se puedan agrupar en una tipología de registros: culto (CU), coloquial (CO) y neutro (NE). Asumimos, por tanto, una relación directa entre la forma de expresarse (registro) y el perfil del usuario. En el análisis lingüístico se aplicarán técnicas de PLN tales como el análisis morfosintáctico y el *chunking* empleando las herramientas disponibles. Adicionalmente y valiéndonos de la información obtenida en los procedimientos anteriores, se prevé un estudio más superficial orientado a la búsqueda de segmentos recurrentes (n-gramas). Los datos resultantes (léxico, *chunks*, expresiones, etc.) serán clasificados manualmente para asignarles el registro.

El segundo método para obtener los patrones consiste en anotar las criticas de hoteles del mismo corpus con nuestra tipología de registros (CU, CO, NE) y, de la misma manera que se hizo en el procedimiento anterior, analizar lingüísticamente los textos para obtener los patrones asociados a cada registro. Un punto que puede parecer delicado de esta aproximación es, justamente, la razón que nos lleva a asociar cada texto de opinión a un registro. No obstante, creemos que se trata de una inferencia equivalente a la que se suele asumir en la literatura sobre el análisis de sentimiento cuando se hacer corresponder la valoración vertida por el usuario en la crítica con la valoración numérica global [8]. La coincidencia o no de los patrones obtenidos en los dos procedimientos nos servirá para constatar el grado de fiabilidad que podemos atribuir a dichos patrones para discriminar los textos de opinión por registro.

Como resultado de aplicar esta metodología se dispondrá de un recurso compuesto por un léxico de palabras, frases y expresiones que nos servirá para predecir el registro al que pertenece un determinado texto de opinión. Para evaluar su capacidad de predicción en otros dominios, clasificaremos de forma automática un conjunto de críticas de películas del corpus MuchoCine[5] [2] y compararemos los resultados con las etiquetas asignadas de forma manual por un grupo de anotadores. Las conclusiones obtenidas con esta investigación nos servirán, entre otras cosas, para determinar si es viable ampliar la tipología de registros propuesta a otros de diferente matiz: directo / indirecto, experto / inexperto, etc., lo que supondría contar con un número mayor de perfiles de usuarios.

[5] El corpus de MuchoCine (http://www.lsi.us.es/~fermin/corpusCine.zip) tiene un total de 3.878 críticas y aproximadamente 2 millones de palabras, con una media de 546 palabras por crítica. Cada crítica ha sido procesada con FreeLing para obtener información léxica, morfosintáctica y semántica codificada en diferentes ficheros.

Agradecimientos. Los autores agradecen a los proyectos de investigación: MICINN TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 y TEXT-KNOWLEDGE 2.0. TIN2009-13391-C04-04 (Plan I+D+i).

Referencias

[1] Brusilovsky, P. y Maybury, M. From Adaptive Hypermedia to the Adaptive Web. *Communications of the ACM*. 45(5): 31-33, 2002.
[2] Cruz Mate, Fermín, et al. Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento del lenguaje natural*. N. 41 (sept. 2008). ISSN 1135-5948, pp. 73-80
[3] De Gemmis, M., Iaquinta, L., Lops, P., Musto, C., Narducci, F. y Semeraro, G.. Preference learning in recommender systems. *Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.
[4] Derek, Bridge. *Towards conversational recommender systems: A dialogue grammar approach*. Proceedings of the Workshop in Mixed-Initiative Case-Based Reasoning, 2002.
[5] Kelly, D. Implicit feedback: Using behavior to infer relevance. A. Spink and C. Cole (Eds.) *New Directions in Cognitive Information Retrieval*. Springer Publishing: Netherlands. 2005: 169-186.
[6] Montaner, M., López, B. y J. L. D. L. Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19(4):285-330, 2003.
[7] Morales-del-Castillo, J.M., Herrera-Viedma, E. Peis. *Modelo semántico-difuso de un sistema de recomendaciones de información para bibliotecas digitales universitarias*. II Simposio sobre Lógica Fuzzy y Soft Computing (LFSC 2007). pages. 73–80, 2007.
[8] Moreno Ortiz, A., Pineda Castillo, F. y Hidalgo García, R. *Análisis de Valoraciones de Usuario de Hoteles con Sentitext: un sistema de análisis de sentimiento independiente del dominio* . Procesamiento del Lenguaje Natural, Revista no45, septiembre 2010, pp 31-39.
[9] Reitter, D., Covaci, S., Oltean, F., Bacanu, C. y Serbanuta, T. Hybrid natural language processing in a customer-care environment. *Proc. of the 11th TaCoS*, 2001.
[10] Sáchez, Pedro José. *Modelos para la combinación de preferencias en toma de decisiones: herramientas y aplicaciones*. PhD thesis, Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada. 2007.
[11] Schmitt S. y Bergmann R. A Formal Approach to Dialogs with Online Customers. *The 14th Bled Electronic Commerce Conference*. Bled, Slovenia, 2001:309-28.
[12] Zukerman, I. y Litman, D. Natural Language Processing and User Modeling: Synergies and Limitations. *User Modeling and User-Adapted Interaction*. Vol. 11: 129-158, 2001

EmotiBlog: Towards a Finer-Grained Sentiment Analysis and its Application to Opinion Mining

1 Introduction and Motivation

The exponential growth of the subjective information freely available on the Web and the employment of new textual genres has created an explosion of interest in Sentiment Analysis (SA). This is a task of Natural Language Processing (NLP) in charge of identifying the opinions related to a specific target (Liu, 2006). Subjective data has a great potential. It can be exploited by business organizations or individuals, for ads placements, but also for the Opinion Retrieval/Search, etc (Liu, 2007). Our research is motivated by the lack of resources, methods and tools to properly threat subjective data. Our main purpose is to demonstrate that EmotiBlog - a corpus annotated with the EmotiBlog annotation schema **for detecting** subjectivity in the new textual genres- can be successfully employed to overcome the challenges of fine-grained SA. We also want to demonstrate that it contributes to solve the shortage of coarse-grained annotation data and improves the Opinion Mining (OM) task. In order to achieve this, we train our Machine Learning system with *EmotiBlog Kyoto*[1] and *EmotiBlog Phones,* but also with the *JRC*[2] corpus[3]. Then, we train with the *EmotiBlog corpus* finer-grained features (not available in the *JRC* annotation) and we integrate *SentiWN* (Esuli and Sebastiani, 2006) and *WordNet* (Miller, 1995). We also employ NLP techniques (stemmer, lemmatiser, bag of words, etc.) to improve the results obtained with the supervised ML models. After that, we apply the trained system to the OM task (using a collection of reviews from *Amazon*[4]), to automatically detect the users' points of view about a mobile phone or one/more of its features. In previous works it has been showed that *EmotiBlog* is a beneficial resource for Opinionated Question Answering (OQA), as stated Balahur et al. (2009 c and 2010a,b) and for Automatic Summarization of subjective content (Balahur et al. 2009a). Thus, the first objective of our research is to demonstrate that *EmotiBlog* is a useful resource to train ML systems for OM applications. Most work done in OM only concentrated on classifying polarity of sentiments into *positive/negative*, thus our second objective is to demonstrate that the combination of training (*EmotiBlog* and *JRC)* is beneficial since we have more data for the common elements, but also a finer-grained analysis, assured by *EmotiBlog*. As a consequence, our third purpose is to demonstrate that a deeper text classification for the OM task is essential. There is the need for *positive/negative* text categories, but also *emotion intensity* (*high/medium/low)*, the *emotion type* (Boldrini et al, 2009a) and the annotation of the linguistic elements that give the subjectivity to the discourse. The complete list of elements is presented in Boldrini et al. (2010). Finally the fourth objective of this research is the implementation of an OM application prototype (which will reinforce the system utility) for retrieving opinions on a product or its features continuing the work proposed by Balahur et al. (2009b).

2 Corpora

The corpus (in English) we mainly employed in this research is *EmotiBlog Kyoto* extended with the collection of mobile phones reviews extracted from *Amazon* (*EmotiBlog Phones*)[5]. It allows the annotation at *document/sentence/element level* (Boldrini et al. 2010), distinguishing between *objective/subjective* discourse. A list of tags for the subjective elements is available

[1] The EmotiBlog corpus is composed by blog posts on the Kyoto Protocol, Elections in Zimbabwe and USA election, but for this research we only use the EmotiBlog Kyoto (about the Kyoto Protocol)

[2] http://langtech.jrc.ec.europa.eu/JRC_Resources.html

[3] feasible since they have common tags and this will allow us to have a larger data set for the common annotated elements

[4] www.amazon.com

[5] Available on request from authors

(Boldrini et al, 2009a) – *source*, *topic*, *verbs*, *nouns*, *adjectives*, *adverbs*, *sayings*, *collocations*, etc. For all of these elements, common attributes are annotated: *polarity*, *degree* & *emotion*. Table 1 presents the size of the corpus and its subjective elements.

Table 1: corpus overview

EB Kyoto	**Ws**: 12328 **Sub**: 210 **Ps**: 62 **Ng**: 141 **Obj**: 347 **Ph**: 692					
	Adj	**Noun**	**Adv**	**Prep**	**Pron**	**Verb**
	161	154	70	13	52	140
EB Phones	**Ws**: 7759 **Sub**: 246 **Ps**: 198 **Ng**: 47 **Obj**: 172 **Ph**: 521					
	Adj	**Noun**	**Adv**	**Prep**	**Pron**	**Verb**
	212	61	94	0	0	39
EB Full	**Ws**: 20087 **Sub**: 455 **Ps**: 260 **Ng**: 188 **Obj**: 519 **Ph**: 1213					
	Adj	**Noun**	**Adv**	**Prep**	**Pron**	**Verb**
	373	215	164	13	52	179
JRC	**Ws**: 39214 **Sub**: 427 **Ps**: 193 **Ng**: 234 **Obj**: 863 **Ph**: 427					
	Adj	**Noun**	**Adv**	**Prep**	**Pron**	**Verb**
	0	0	0	0	0	0

Where *Ws*, *Sub*, *Ps*, *Ng*, *Obj*, *Ph*, *Adj*, *Noun*, *Adv*, *Prep*, *Pron* and *Verb* correspond to the number of words, subjective/positive/negative/objective sentences, total of phrases, adjectives, nouns, adverbs, prepositions, pronouns and verbs which have been annotated. We also used the *JRC* quotes[6], a set of 1590 English language quotations extracted automatically from the news and manually annotated for the sentiment expressed towards entities mentioned inside the quotation. The *JRC* is labelled in a coarse-grained way –if compared with *EmotiBlog*- thus, we use it to train our ML system for the element it has in common with *EmotiBlog,* and we then improve the training adding the *EmotiBlog* finer-grained elements.

3 ML Experiments

In order to demonstrate that *EmotiBlog* is valuable resource for ML, we perform a series of experiments with different approaches, corpus elements and resources. First, we employ the bag of word extracted from the train corpus (*EmotiBlog*) and use basic techniques: tokenisation and dimensionality reduction by term selection (TSR) methods. Table 2 shows the most significant results. We used Support Vector Machine (SVM) due to the promising results obtained by Boldrini et al. (2009b). For TSR, we compared Information Gain (IG) and Chi Square (X2) for reducing the dimensionality substantially with no loss of effectiveness (Yang and Pedersen, 1997). For the feature weight needed by SVM we adopted the binary weight, assigning 1 to the feature that appears in the sample and 0 otherwise; tf/idf, which sets the tf/idf value (Salton and Buckley, 1988) of each feature if it appears in the sample and 0 otherwise. For tf/idf approach, we have also used the normalized one, tf/idfn (Sebastiani, 2002).

Table 2: ML experiments results

EM elements	**F-measure**	**Precision**	**Recall**	**Classes**
objectivity	0.6223	0.6601	0.642	**2**
polarity	0.6196	0.7209	0.6612	**2**
degree	0.5709	0.5985	0.6026	**3**
emotion	0.5712	0.6096	0.6433	**3**
obj+pol	0.5431	0.5771	0.5866	**3**
obj+pol+deg	0.4922	0.5018	0.5612	**9**

Table 2 shows the best results obtained using lemmatiser or stemmer. The stemmer improves the results in evaluation with few features and the lemmatiser when features are reduced. The tf/idf performs better in each evaluation, except for the polarity where td/idf normalised set is used. TSR systems obtain high results in each case without any significant differences between X2 and IG and the range of featured has changed between 100 and 800 depending on the number of classes. From the results in the mix of elements (*objectivity*/*polarity*) or *objectivity*/*polarity*/*degree* we can deduce that learning a model, which combines such elements improves the performance. To evaluate the degree we will first determine if the sentence is

[6] http://langtech.jrc.ec.europa.eu/JRC_Resources.html

subjective/objective, its *polarity* and *intensity*, thus increasing the possibility of mistakes. In order to check the impact of including the semantic relation as learning features, we believe that, grouping features by their semantic relations will increase the coverage in the test corpus a part from reducing the samples' dimensionality. The challenge at this point is Word Sense Disambiguation (WSD) due the poor results that these systems traditionally obtain in international competitions (Agirre et al. 2010). Choosing the wrong sense of a term would introduce noise in the evaluation and thus a low performance. The question is that if we include all senses of a term in the set of features, if the TSR would choose the correct ones. If we use all *WordNet* senses of each term as learning features, then the TSR methods IG/X2 could remove the not useful senses to classify the sample in the correct class. In this case this disambiguation methods would be adequate. The evaluation summarized in Table 3 focuses on solving these questions.

Table 3: Results with lexical resources

EM elements	F1	Precision	Recall	Resources
objectivity	0.6261	0.6538	0.6409	**swn+wn**
polarity	0.6195	0.6809	0.6481	**swn+wn**
degree	0.6101	0.6287	0.6381	**swn1+wn1**
emotion	0.5637	0.6114	0.6239	**swn+wn**
obj+pol	0.5493	0.5946	0.5959	**swn+wn**
obj+pol+deg	0.4802	0.4724	0.5458	**swn+wn**

We used two lexical resources: *WordNet* and *SentiWordNet*. The first one because it contains a huge quantity of semantic relations between English terms; and the second one since, the use of this specific OM resource demonstrated to improve the results of OM systems. *SentiWordNet* also assigns to each synset of WordNet 3 sentiment scores: *positivity/negativity/objectivity*. As we can observe in Table 3, experiments have been carried out with 5 different configurations using: i) *SentiWN* synsets, ii) WN synsets, iii) a combination of both, iv) *SentiWN* synsets+scores and v) *SentiWN* synset+scores combined with *WN* synsets. In configuration i) if the lemma of a word appeared in *SentiWN*, it was replaced by the related synset. If a word is not found the lemma it will be left. For the experiment ii) we use WN instead of *SentiWN* and repeat the previous process. In the next experiment (iii), we first compare terms with *SentiWN* and, only if, terms are not found, *WN* is used. In the fourth case, if a term appeared in *SentiWN* it was replaced by the related synset and their associated polarity scores as new attributes. Finally, in the last configuration (iv), we applied the previous process but, if a term does not appear in *SentiWN*, we checked if it does in *WN* and if found, it was replaced only by its synset. As always, in case the word was not found, its lemma was left. In order to solve the ambiguity, 3 techniques have been adopted: taking into account only the most frequently sense, including all senses, or including all senses but using both TSR techniques (IG & X2) with the goal of checking these methods as disambiguators. As we can see in previous table 3, most of experiment using *SentiWN* and *WN* improve slightly the results if compared to Table 2. Methods, which use IG and X2 improve the majority of the results confirming our hypothesis they are adequate for disambiguation. Finally, we have applied these models with the *JRC* corpus. These experiments obtain **0.70** and **0.66** of f-measure for *objectivity* and *polarity* respectively. Although the results with *JRC* are slightly higher than the ones with *EmotiBlog*, this is because *EmotiBlog* has a finer-grained text analysis and is much smaller than the *JRC*. Moreover, *JRC* is based on more formal texts, which do not have the language variability that *EmotiBlog* has. In the future we hope to improve the results increasing the *EmotiBlog* corpus with more samples and domains.

4 GPLSI EmotiReview

After having performed the previous experiments, we created an on-line application (*GPLSI EmotiReview*[7]) for exploiting the learnt models to the real life –adapting it with a domain onthology. *GPLSI EmotiReview* is the first version of a prototype of an OM system that could be employed to extract the overall opinion or some features of mobile phone. The system is divided into 2 modules: i)the intelligent crawler tracks user's opinions in specialised Web pages

[7] http://intime.dlsi.ua.es:8080/emotireview

(offline); and ii) the users queries are processed and the requested opinions given back the user (real-time). The crawler includes ML tools to detect which comments are subjective, discriminates them into *positive/negative/level of subjectivity* of the emotion expressed. In order to detect the emotion target, we follow the approach by Qiu et al. (2006) who use *Minipar*[8] to detect the syntactic relation between terms. In our case, thanks to *EmotiBlog* we have the subjective words annotated and we use Minipar to find their syntactic relations. In order to improve this process we link the subjective terms with an ontology we manually created (which includes all the features of mobiles) and in this way we understand better which adjective is related with which feature of mobiles. If we cannot find any relation, the target will be the general one of the document. If some feature product or one of its feature (screen, battery, memory) is detected inside the window near to a subjective expression, this expression will be about this target and if not it will refer to the product in general. In order to obtain the relation of a product and its features a specific ontology has been built (about the smart phones domain) which will also be extended in the future to be also useful for other areas. Once the information is collected, the system uses Lucene[9] (Hatcher and Gospodnetic, 2004) as search engine to find (between the stored products), to retrieve the products (similar to the query) and give back the related opinions as well as the general evaluation and its specific evaluations, if applicable.

5 Conclusions and Future Works

The first contribution this paper brings is the employment of *EmotiBlog* –a collection of blog posts labelled with the homonymous annotation schema- and the *JRC* corpus. They have been employed to train and test our ML system for the automatic detection of subjective data in the *EmotiBlogPhones* corpus, an extension of the *EmotiBlog*. We used both corpora to train the system regarding their common labelled elements and then *EmotiBlog* for a finer-grained text analysis, since it contains a finer-grained annotation. We processed all the combinations of TSR, tokenisation and term weight for a total of 660000 experiments, but due to space reasons we showed the most significant. Another contribution is the implementation of an OM application prototype for retrieving the general opinions about a phone and its features. Due to the complexity of OM, there is room for the improvement for this task. First, in order to improve the target detection mechanism our intention is to use learning models based on sequence of words (n-gram, Hidden Markov Models, etc.) to detect the topic of published opinion and thus, making a comparative assessment of different techniques, which will be also employed to detect linguistic phenomena based on the consequentiality mechanisms for expressing denials, irony and sarcasm. We also intend to use temporality resolution techniques for detecting lines of argument in different posts from the same source to find possible incoherence and abstract irony or sarcasm. Last but not least, another future work line includes the extension of *EmotiBlog* annotation (data and languages) in order to have at disposal more data for the ML training and test.

References[10]

1. Agirre, E., Lopez de Lacalle, O., Fellbaum, C., Hsieh, S., Tesconi, M., Monachini, M., Vossen, P., Segers, R. 2010. SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain.
2. Abulaish, M., Jahiruddin, M., Doja, N. and Ahmad, T. 2009. Feature and Opinion Mining for Customer Review Summarization. PReMI 2009, LNCS 5909, pp. 219–224, 2009. Springer-Verlag Berlin Heidelberg.
3. Balahur A., and Montoyo A. 2008. Applying a Culture Dependent Emotion Triggers Database for Text Valence and Emotion Classification. In Proceedings of the AISB

[8] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm
[9] http://lucene.apache.org

[10] included in the paper and for the related work in general

2008 Symposium on Affective Language in Human and Machine, Aberdeen, Scotland.
4. Balahur A., Lloret E., Boldrini E., Montoyo A., Palomar M., Martínez-Barco P. 2009a. Summarizing Threads in Blogs Using Opinion Polarity. In Proceedings of ETTS workshop. RANLP.
5. Balahur, A., Boldrini, E., Montoyo, A., Martínez-Barco, P. 2009c. Opinion and Generic Question Answering systems: a performance analysis. In Proceedings of ACL, 2009, Singapore.
6. Balahur, A., Boldrini, E., Montoyo, A., Martínez-Barco, P. 2010b. Opinion Question Answering: Towards a Unified Approach. In Proceedings of the ECAI conference.
7. Balahur, A., Boldrini, E., Montoyo, A., Martínez-Barco 2009b. P. Cross-topic Opinion Mining for Realtime Human-Computer Interaction. ICEIS 2009.
8. Boldrini, E., Balahur, A., Martínez-Barco, P., Montoyo, A. 2010. EmotiBlog: a finer-grained and more precise learning of subjectivity expression models. In Proceedings of LAW IV, ACL.
9. Boldrini, E., Balahur, A., Martínez-Barco, P., Montoyo, A. 2009a: EmotiBlog: an Annotation Scheme for Emotion Detection and Analysis in Non-traditional Textual Genres. In Proceedings of DMIN, Las Vegas.
10. Balahur, A., Boldrini, E., Montoyo, A., Martínez-Barco, P. 2010a. A Unified Proposal for Factoid and Opinionated Question Answering. In Proceedings of the COLING conference.

11. Boldrini E., Fernández J., Gómez J.M., Martínez-Barco P. 2009b. Machine Learning Techniques for Automatic Opinion Detection in Non-Traditional Textual Genres. In Proceedings of WOMSA 2009. Seville, Spain.
12. Chaovalit P, Zhou L. 2005. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. In Proceedings of HICSS-05.
13. Cui H., Mittal V., Datar M. 2006. Comparative Experiments on Sentiment Classification for Online Product Reviews. In Proceedings of the 21st National Conference on Artificial Intelligence AAAI.
14. Cerini S., Compagnoni V., Demontis A., Formentelli M., and Gandini G. 2007. Language resources and linguistic theory: Typology, second language acquisition. English linguistics (Forthcoming), chapter Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. Franco Angeli Editore, Milano, IT.
15. Dave K., Lawrence S., Pennock, D. 2003. "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews". In Proceedings of WWW-03.
16. Esuli A., Sebastiani F. 2006. SentiWordNet: A Publicly Available Resource for Opinion Mining. In Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy.
17. Gamon M., Aue S., Corston-Oliver S., Ringger E. 2005. Mining Customer Opinions from Free Text. Lecture Notes in Computer Science.
18. Hatcher E. and Gospodnetic O. 2004. Lucene in Action. Manning Publications.
19. Hatzivassiloglou V., Wiebe J. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In Proceedings of COLING.
20. Liu 2006. Web Data Mining book. Chapter 11
21. Liu, B. (2007). Web Data Mining. Exploring Hyperlinks, Contents and Usage Data. Springer, first edition.
22. Miller, G.A. 1995. WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41
23. Mullen T., Collier N. 2004. Sentiment Analysis Using Support Vector Machines with Diverse Information Sources. In Proceedings of EMNLP.
24. Ng V., Dasgupta S. and Arifin S. M. 2006. Examining the Role of Linguistics Knowledge Sources in the Automatic Identification and Classification of Reviews. In the proceedings of the ACL, Sydney.
25. Ohana, B.,Tierney, B. 2009. Sentiment classification of reviews using SentiWordNet, T&T Conference, Dublin Institute of Technology, Dublin, Ireland, 22nd.-23rd.

26. Pang B and Lee L. 2003 Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting of the ACL, pages 115–124.
27. Pang B., Lee L, Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of EMNLP-02, the Conference on Empirical Methods in Natural Language Processing.
28. Qiu, G., Liu, B., Bu, J., Chen, C. 2006. Opinion Word Expansion and Target Extraction through Double Propagation. Association for Computational Linguistics
29. Riloff E. and Wiebe J. 2003. Learning Extraction Patterns for Subjective Expressions. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing.
30. Salton, G. and Buckley, C. 1988. Term-weighting Approaches in Automatic Text Retrieval. Inform. Process. Man. 24, 5, 513–523
31. Sebastiani F. 2002. Machine Learning in Automated Text Categorization. ACM Computing Surveys.
32. Strapparava C. Valitutti A. 2004. WordNet-Affect: an affective extension of WordNet. In Proceedings ofthe 4th International Conference on Language Resources and Evaluation, LREC.
33. Turney P. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. ACL 2002: 417-424.
34. Yang Y. and Pedersen J.O. 1997. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of ICML-97, 14th International Conference on Machine Learning.

Actas IV Jornadas TIMM, pp 55-56

Trabajo de doctorado: Recuperación de información orientada a la minería de opiniones

Javi Fernández, José M. Gómez, and Patricio Martínez-Barco

Universidad de Alicante

1. Introducción

El objetivo principal del trabajo de doctorado es el desarrollo de un sistema de recuperación de información orientado a la minería de opiniones, en el que se investiguen las influencias positivas entre las tres áreas que abarca: *recuperación de información* (RI), *rastreo web* (RW) y *minería de opiniones* (MO).

2. Descripción

Hasta la fecha se han realizado diversos trabajos relacionados en las diferentes áreas mencionadas, con el fin de realizar un acercamiento inicial a cada una de ellas y preparar las técnicas y herramientas que serán necesarias para alcanzar el objetivo final.

- Respecto a la MO, los estudios realizados se centran en la categorización de textos según categorías relacionadas con la subjetividad, como pueden ser la polaridad o la emoción. En ellos se han realizado diferentes experimentaciones utilizando recursos semánticos como *SentiWordNet* además de otras técnicas conocidas en PLN.
- Respecto a la RI, también se han estudiado las influencias de recursos semánticos en el cálculo de la relevancia de los documentos recuperados, tales como *clases semánticas* y *WordNet*. También se han realizado diferentes experimentaciones y posteriores evaluaciones de un sistema de RI con usuarios reales dentro de un ámbito concreto, oportunidades de negocio y exportación inteligente.
- Respecto al RW, se ha desarrollado un sistema de crawling muy adaptable y escalable, con el que se estudiarán las influencias de las relaciones entre documentos en la Web dentro de la RI y la MO. Debido a la falta de recursos que evalúen este tipo de sistemas desde el punto de vista científico, no se ha podido evaluar convenientemente. Este será uno de los objetos de estudio en posteriores trabajos.

En el último trabajo se ha realizado un estudio intensivo sobre la influencia de varias de las técnicas existentes en PLN en la tarea de clasificación de opiniones. Esta clasificación incluye, además de la tradicional categorización de objetividad y polaridad, categorías que reflejan otros aspectos relevantes de las opiniones:

la intensidad de la opinión y el tipo de emoción expresada. Evaluar este tipo de categorías ha sido posible gracias a la utilización del corpus *EmotiBlog*, un corpus de granularidad fina y una gran variedad de información anotada.

3. Trabajo futuro

En la actualidad se está desarrollando un sistema de recuperación de opiniones en el marco de la telefonía móvil. El objetivo de este sistema es el de obtener, para un producto dado, todas las opiniones y su valoración general dentro de un conjunto de documentos pertenecientes a dominios web seleccionados. Para ello se han utilizado tanto herramientas existentes como desarrolladas específicamente para esta tarea, utilizando *InTime* como sistema de integración. Entre las herramientas existentes podemos destacar las siguientes: *Minipar* para el análisis léxico y búsqueda de relaciones entre palabras; *Weka* para el aprendizaje y clasificación de documentos según polaridad; y *Lucene* para la recuperación final de documentos. Las herramientas desarrolladas se enfocan en técnicas de rastreo y obtención de documentos en la Web, modelos de selección de términos, aproximaciones para la detección semiautomática de atributos y medidas para el cálculo de las valoraciones finales. Este sistema posteriormente será trasladado al ámbito socio-económico.

Creación de un sistema de reconocimiento de emociones en alumnos de primaria

Eladio Blanco[1,1], Fernando Martínez[1] y Antonio Pantoja[1]

elbloo@gmail.com, dofer@ujaen.es y apantoja@ujaen.es

Abstract. En este artículo se presenta una experimentación preliminar realizada con el corpus e-Culturas recopilado en la edición 2009 del mismo proyecto a partir de textos escritos por niños de 10 y 11 años en los que se relata algún acontecimiento que al niño le haya producido miedo, alegría o tristeza. La finalidad del corpus es el entrenamiento de un categorizador automático para esos sentimientos. Posteriormente se prevé integrar tal categorizador dentro de la plataforma e-Culturas. Esta plataforma tiene por objetivo fomentar la interculturalidad, y es ahí donde el análisis de sentimientos puede resultar de gran ayuda: si es posible detectar automáticamente una posible reacción negativa del niño frente a determinadas situaciones específicas de la interculturalidad, es posible proponer actividades concretas orientadas a corregir esa aversión o prejuicio del niño. Los resultados obtenidos en la evaluación del categorizador son esperanzadores. Demuestran que la metodología utilizada para la creación del corpus es adecuada, y que técnicas específicas del análisis de sentimientos pueden rendir a un nivel suficientemente bueno como para ser de utilidad dentro de la Red Internacional e-Culturas.

Keywords: Análisis de sentimientos, minería de opiniones, creación de corpus, categorización automática.

1 Introducción

El análisis de sentimientos (sentiment analisys) es una tarea del Procesamiento del Lenguaje Natural que se preocupa por el tratamiento automático de opiniones, sentimientos y subjetividad presente en el texto escrito [8]. La popularización de lo que ha venido a denominarse Web 2.0 requiere de herramientas adecuadas que permitan explotar de una manera adecuada tal flujo de información. Es prometedora la confluencia de toda esta nueva fuente de texto subjetivo junto con herramientas que

[1] Please note that the LNCS Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

permitan modelar automáticamente el conocimiento allí expresado de tal forma que sea computacionalmente manipulable.

En el presente trabajo se propone la aplicación de técnicas propias de análisis de sentimientos como una herramienta valiosa para la integración intercultural de niños de 10 y 11 años dentro de la Red Internacional e-Culturas [1, 7].

1.1 El proyecto e-Culturas

La Red internacional e-Culturas® (www.e-culturas.org) tiene como principal finalidad contribuir a una mejor comprensión de los aspectos que diferencian y a la vez unen a las personas dentro de un modelo de convivencia multicultural e intercultural. Para ello se utilizan las Tecnologías de la Información y la Comunicación (TIC) aplicadas a un programa intercultural interactivo que trabajan de forma colaborativa niños hermanados de diferentes países del planeta.

En la edición del 2009 que fue la utilizada en la creación del corpus, participaron 517 alumnos de 5 países diferentes (Argentina, Brasil, Chile, España y Paraguay), agrupados en 14 centros y 18 aulas. Todos eran alumnos de 6º de Educación Primaria de España y de niveles similares en el resto de los países.

Se pretende que algunas de las tareas de la plataforma e-Culturas vayan encaminadas a reforzar positivamente aquellos aspectos que le causen rechazo al alumno, y es aquí donde es necesario aplicar análisis de sentimientos: la plataforma debe analizar los textos de los niños buscando determinadas reacciones, y una vez detectadas, proponer ciertas actividades correctoras. En este primer estadio del trabajo nos hemos centrado en tres sentimientos: alegría tristeza y miedo.

2 Experimentación preliminar

El corpus utilizado se elaboró en la edición 2009 del proyecto e-Culturas [2], estando formado por 1160 documentos escritos por alumnos de los 5 países participantes y en cada uno de los cuales el niño relata situaciones vividas en las que sintió miedo, alegría o tristeza. Se trata, en general, de textos cortos de unas 27 palabras de media.

Se han realizado numerosos experimentos teniendo en cuenta diversas características como la ocurrencia y frecuencia de palabras, n-gramas, etiquetas POS (Part Of Speech) y lemas; ocurrencia y frecuencia de trigger words, palabras enfatizadas, símbolos especiales… En este trabajo las trigger words están formadas por conjuntos de unas 10-15 palabras semilla que se han seleccionado manualmente a partir del corpus de entrenamiento (véase Tabla 1). Algunos de los experimentos en los que se han obtenido mejores resultados son los siguientes:

a. SEM: Ocurrencia de las palabras semilla.
b. DIC: Ocurrencia de las palabras de diccionario.
c. NGR: Ocurrencia de n-gramas de más de 4 letras.
d. DIC+POS: Ocurrencia de las palabras de diccionario y etiquetas POS.
e. NGR+POS: Ocurrencia de n-gramas de más de 4 letras y etiquetas POS.

Finalmente, se ha procedido a entrenar un algoritmo de aprendizaje supervisado SVM [4] con el corpus de entrenamiento, evaluando la bondad del modelo sobre el corpus de evaluación para cada uno de los experimentos anteriores.

Tabla 1. Algunas palabras semilla.

Emoción	Palabras Semilla
Alegría	Ganar, contento, regalar, reír, aprobar, familia
Tristeza	Llorar, pena, lástima, perder, suspender, morir, romper, enfadar
Miedo	Susto, secuestrar, terror, pegar, pesadilla, asustar, inquietud, morir, horror

3 Análisis de resultados

Los resultados obtenidos se describen en la Tabla 2. A pesar del sencillo modelo de categorización implementado los resultados son esperanzadores, si bien hay diferencias significativas entre cada emoción dependiendo del experimento realizado, por ejemplo, tanto en NGR y NGR+POS mejoran los resultados obtenidos para tristeza y empeoran los de miedo y alegría si los comparamos con sus experimentos predecesores DIC y DIC+POS respectivamente.

Tabla 2. Resultados de la clasificación de sentimientos.

Exp.	Emoción	Accuracy	Prec./Cobert.
SEM	Alegría	0,796	0,681/0,833
	Miedo	0,918	1/0,692
	Tristeza	0,755	0,687/0,611
DIC	Alegría	0,829	0,844/0,675
	Miedo	0,914	0,893/0,807
	Tristeza	0,762	0,765/0,382
NGR	Alegría	0,8	0,771/0,675
	Miedo	0,905	0,862/0,807
	Tristeza	0,8	0,809/0,5
DIC + POS	Alegría	0,839	0,849/0,7
	Miedo	0,905	0,862/0,807
	Tristeza	0,771	0,778/0,412
NGR + POS	Alegría	0,819	0,8/0,7
	Miedo	0,895	0,857/0,774
	Tristeza	0,781	0,789/0,441

Como es usual en las tareas de categorización, una precisión elevada se suele conseguir penalizando la cobertura, tal es el caso del miedo en SEM con una precisión de 1, pero a costa de una discreta cobertura que no alcanza el 70% de los textos marcados con tal emoción. Sin duda, este resultado nos indica que el conjunto de palabras semilla seleccionado para esa emoción es preciso, pero incompleto. En el otro extremo se encuentra la alegría, que obtiene una cobertura que supera el 80%, pero con pobre resultado en cuanto a la precisión: un 0,68. Esto es, posiblemente superar esa cobertura a partir de las sencillas características léxicas y morfosintácticas

utilizadas sea muy difícil, si no imposible, pero sí pensamos que es posible reducir la lista de palabras semilla sin que la cobertura se resienta. Finalmente, la tristeza se muestra como la emoción más difícil, con un resultado discreto.

4 Conclusión y trabajo futuro

Para validar el corpus e-Culturas (creado para el idioma español a partir de textos escritos por niños de 10 y 11 años relativos a experiencias en las que hayan sentido miedo, tristeza o alegría) se ha procedido a realizar una experimentación preliminar, orientada a la categorización automática de los sentimientos. Los resultados que hemos obtenido son consistentes con el corpus tal cual está etiquetado manualmente, alcanzando una precisión media de 82.7% para una cobertura del 60%. El análisis de errores realizado por el categorizador desarrollado muestra que el enfoque para crear el corpus es el adecuado: los errores tipo cometidos requieren ampliar el conjunto de palabras semillas o añadir algunas características basadas en reglas morfo-sintácticas. Sin embargo, algunos errores como la distinción entre miedo y tristeza son más sutiles, y posiblemente requiera de técnicas más afinadas.

El corpus ampliado con el de las siguientes ediciones se utilizará para entrenar un categorizador automático de reconocimiento de emociones que se integrará en la plataforma de la Red Intencional e-Culturas con el fin de detectar la emoción predominante en el alumno cuando escriba un determinado texto.

El fin de este sistema será detectar conductas negativas relacionadas con la interculturalidad, de tal modo que mediante el reconocimiento de las emociones en determinados contextos se les podrá proponer a los alumnos actividades complementarias para tratar de minimizar esas emociones en el caso de que sean negativas o potenciarlas si son positivas. Por ejemplo, si en un ejercicio sobre la inmigración se detecta la emoción *ira* en el alumno, puede que este presente comportamientos racistas hacia los inmigrantes, por lo que se le asignarán actividades que tratarán de corregir este comportamiento.

Bibliografía

1. Alcaide, M., Blanco, E., Pantoja, A., & Jiménez, A.: Capacitación de maestros en valores interculturales a través de la red internacional e-Culturas. INECE'08. Madrid (2008)
2. Blanco, E., Martínez, F., Pantoja, A., & Ureña, A.: Análisis de emociones sobre un corpus de textos escritos por niños de Educación Primaria. CISTI. Santiago de Compostela (2010)
3. Bisquerra, R.: Educación Emocional y Bienestar. Barcelona: Praxis (2002)
4. Cristianini, N., & Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)
5. e-Culturas. Recuperado el 15 de Enero de 2010, de http://www.e-culturas.org (2010)
6. Pang, B., & Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval , 2 (1-2), pp. 1--135 (2008)
7. Pantoja, A.: Internet y la construcción de una ciudadanía intercultural. Balance de una experiencia. I Jornadas Internacionales y VI Jornadas sobre Diagnóstico y Orientación "El carácter universal de la educación intercultural, pp. 72--128 (2008)

Actas IV Jornadas TIMM, pp 61-63

Análisis de Sentimientos

Eugenio Martínez Cámara, MªTeresa Martín Valdivia, L. Alfonso Ureña

SINAI - Sistemas Inteligentes de Acceso a la Información
Departamento de Informática
Universidad de Jaén
{emcamara,maite,laurena}@ujaen.es

Entre una de las múltiples tareas de las que se ocupa el PLN se encuentra la clasificación de textos, que consiste en la asignación de un conjunto de categorías a una colección de documentos, resolviéndose de esta forma la clasificación objetiva de documentos.

Existe una gran cantidad de textos en el que el contenido subjetivo es lo más relevante, y cuyo procesamiento no debería limitarse a aplicar únicamente las técnicas de la clasificación de documentos. Ante esta necesidad de clasificar la orientación, o la opinión que se expresan en los documentos, surge la área análisis de sentimientos (AS), o también denominada en la bibliografía como minería de opiniones, o en inglés, sentiment analysis u opinion mining.

El análisis de sentimientos trata de clasificar los documentos en función de la polaridad de la opinión que expresa su autor. Esta nueva área que combina PLN y minería de textos, incluye una gran cantidad de tareas que han sido tratadas en mayor o menor medida [3]. Existen principalmente dos formas distintas de enfrentarse a este problema: aplicando aprendizaje automático [2] o aplicando un enfoque semántico [1]. Dos son las aplicaciones más importantes: determinar la polaridad de las opiniones a nivel de documento, frase o característica, y determinar si un documento contiene opiniones.

Existen muchos trabajos en el campo del análisis de sentimientos, habiéndose aplicado en multitud de dominios, pero la mayor parte de ellos han sido realizados sobre corpus de documentos en inglés. La investigación en AS en español es reducida, siendo la más destacable la que está llevando a cabo el grupo ITALICA de la universidad de Sevilla. El grupo SINAI de la universidad de Jaén también lleva algún tiempo trabajando en esta temática pero centrándose hasta hace unos meses solamente en lengua árabe [5].

Al final del año pasado hicimos una primera aproximación al estudio del análisis de opiniones en español. Se eligió el enfoque de aprendizaje automático propuesto por Pang en [2], por su sencillez, por los buenos resultados que ha dado en trabajos anteriores, y para poder comparar los resultados con los obtenidos por Cruz y otros [4], ya que ellos utilizan el enfoque semántico propuesto por Turney [1]. Se aplicaron diversos algoritmos de clasificación, dos de los que se suelen utilizar en AS, SVM y Naïve Bayes, y otros tres, BBR (regresión logística bayesiana), KNN y C4.5, para determinar que algoritmo se comporta mejor para este problema. Para ello se utilizó un corpus de críticas de cine en español [4], compuesto por 3.878 críticas recogidas de la web *muchocine*[1]. Las críticas están

[1] http://www.muchocine.net

puntuadas en un rango de 1 a 5, significando el 1 una película muy mala, y el 5 una película muy buena. Solamente se utilizaron para la experimentación las puntuadas con 1, 2, 4, 5, eliminado las que tienen un 3 debido a su carácter neutro. Los mejores resultados se obtuvieron aplicando regresión logística bayesiana, es decir, con el algoritmo BBR, el cual no había sido utilizado antes en trabajos de análisis de sentimientos. También hay que destacar el hecho de que se mejoraron considerablemente los resultados obtenidos por Cruz y otros [4] sobre el mismo conjunto de datos.

En los últimos años ha aumentado exponencialmente el uso de las redes sociales en todo el mundo. En todas ellas los usuarios vierten opiniones de cualquier tipo y sobre cualquier tema. Dentro del conjunto de las redes sociales las de microblogging, en concreto *Twitter*[2], no han parado de crecer tanto en el número de usuarios como en el tráfico de datos. En España, en los últimos meses el uso de *Twitter* ha aumentado de forma muy considerable. Como se puede ver, existe una gran cantidad de información que se genera en internet y no se debe dejar de aprovechar la oportunidad de procesarla. Desde finales 2009 están apareciendo trabajos muy interesantes en los que se aplica AS a *Twitter*, pero todos ellos en inglés [6], [7] o [8]. Entre la infinidad de aplicaciones que puede tener aplicar AS a *Twitter* se encuentra el poder determinar la opinión de los usuarios sobre una marca comercial, sobre el último producto presentado por una compañía, sobre la campaña electoral de un partido político, o estimar la intención de voto de un determinado partido político.

Nuestro trabajo ahora mismo se centra en la aplicación de técnicas de análisis de sentimientos a información en español recogida de *Twitter*, sin olvidar la continuación de nuestro primer trabajo de análisis de opiniones en textos en español.

Referencias

1. Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL) pp. 417–424. ACL. Morristown, NJ, USA.
2. Pang, B., Lee, L.,Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) pp. 79–86. Association for Computational Linguistics.
3. Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. Foundation and Trends in Information Retrieval, vol. 2, nos. 1-2, pp. 1–135.
4. Cruz, F.L, Troyano, J.A, Enriquez, F., Ortega J. (2008). Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. In: Procesamiento de Lenguaje Natural, vol 41.
5. M. Saleh, A. Montejo, M.T. Martín, L.A. Ureña. (2009). Prediction of Customer Ratings on a New Corpus for Opinion Mining. Proceedings Working Notes for the WOMSA 2009 Workshop. Sevilla.

[2] http://twitter.com

6. A. Go, R. Bhayani, L. Huang. (2009). Twitter Sentiment Classification using Distant Supervision. CS224N Project Report. Stanford.
7. O'Connor, B. Balasubramanyan R. Routledge, B. Smith, N. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. International AAAI Conference on Weblogs and Social Media, North America.
8. Tumasjan, A. Sprenget, T. Sandner, P. Welpe, I. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. International AAAI Conference on Weblogs and Social Media, North America.

Interacción y ontologías

Actas IV Jornadas TIMM, pp 67-68

Propuesta de algoritmo para extender y poblar ontologías

Jorge Cruanes[⋆†], M. Teresa Romá-Ferri[‡]

[†]Departamento de Lenguajes y Sistemas Informáticos
[‡]Departamento de Enfermería
Universidad de Alicante, Apto. Correos 99, Alicante, España
jcruanes@dlsi.ua.es, mtr.ferri@.ua.es

1 Introduction

El objetivo principal de la tesis será el de diseñar y verificar un algoritmo capaz de, a partir de textos escritos en lenguaje natural y una ontología, extraer el conocimiento relevante y usarlo para ampliar dicha ontología. En este planteamiento la ontología usada será base y diana al mismo tiempo. Primero, la ontología filtrará el conocimiento contenido en los textos, etiquetándolo de forma semántica. Posteriormente la ontología se ampliará con nuevo conocimiento contenido en los mismos textos, extendiendo sus conceptos y poblándola con instancias (términos). La propuesta del método se divide en las siguientes etapas:

1. Procesar el documento mediante la herramienta de PLN FreeLing en tres fases. En la primera se usará EuroWordNet (en castellano) como base de conocimiento general para el etiquetado de los tokens del texto. En una segunda fase se usará una ontología de domino, en nuestro caso OntoFIS (dominio farmacoterapéutico). Esta última fase persigue el etiquetado semántico de los tokens conocidos. Por último se mezclan los resultados anteriores. Para ello, tendrá prioridad la etiquetación semántica obtenida usando la ontología de dominio como base de conocimiento.
2. Si un token sólo es identificado con conocimiento general entonces, mediante medidas de distancia semántica, habrá que conocer su equivalencia con los conceptos de la ontología domino. Se establecerán unos umbrales para considerar a un token como rechazado, candidato o equivalente a un concepto de la ontología.
3. Si un token es desconocido, comienza la etapa principal de nuestra propuesta, descubrir conocimiento. Esta etapa se basa en la utilización de patrones tipo tripletas (T1,R,T2), y donde T1 (sujeto) y T2 (predicado) serán conceptos o instancias, y R una relación. Para poder descubrir nuevo conocimiento, al menos uno de los elementos de la tripleta debe ser conocido, para así servir de referencia. En este punto tenemos previsto cuatro aproximaciones:

⋆ Este artículo ha sido cofinanciado por el Ministerio de Ciencia e Innovación (proyecto TIN2009-13391-C04-01), y la Conselleria d'Educació de la Generalitat Valenciana (proyectos PROMETEO/2009/119, ACOMP/2010/286 y ACOMP/2011/001).

- Si conocemos T1 o T2 y R es taxonómica, el token desconocido se comprobará que es un sustantivo común para determinar, mediante el lugar en la tripleta, si es hiperónimo o hipónimo.
- Si conocemos T1 o T2 y R no es taxonómica, el token desconocido será etiquetado semánticamente el dominio y rango de R.
- Si sólo conocemos la relación, pero ésta tiene un único dominio y rango, entonces la clasificación semántica de los tokens desconocidos será acorde a su posición en la relación.
- Si sólo R es desconocida, tendrá como dominio la categoría semántica de T1 y como rango la categoría de T2.

Para refinar los patrones se usarán valores de umbrales para ser aceptados conforme se vaya incrementando los documentos procesados.

4. Finalmente se hará un control de calidad en dos fases. Primero se comprobará que la ontología resultante es válida, es decir, carece de bucles de herencia y no existen instancias pertenecientes a clases disjuntas. En segundo lugar, si existen conflictos serán resueltos de acuerdo a un algoritmo de toma de decisiones, volviendo a realizar el control de calidad.

Actas IV Jornadas TIMM, pp 69-72

PATHS: Personalised Access To cultural Heritage Spaces

Eneko Agirre, Oier Lopez de Lacalle
University of the Basque Country
{e.agirre,oier.lopezdelacalle}@ehu.es
Paul Clough, Mark Stevenson
University of Sheffield
p.d.clough@sheffield.ac.uk M.Stevenson@dcs.shef.ac.uk

Abstract. This paper describes a European project called PATHS (Personalized Access To cultural Heritage Spaces) that aims to support information exploration and discovery through digital cultural heritage collections. Significant amounts of cultural heritage material are now available through online digital library portals, wich can also be overwhelming for many users who are provided with little or no guidance on how to find and interpret this information. The PATHS project will create a system that acts as an interactive personalised tour guide through existing digital library collections. The system will offer suggestions about items to look at and assist in their interpretation.

1 Introduction

Content and users are setting an exciting agenda for innovation in digital libraries. Growing quantities of digital content and information are becoming available and are being produced in increasingly sophisticated forms. In todays society both individuals and organisations are confronted with growing quantities of content that needs to be made accessible and usable. This requires new services to enable people to create, explore and share content, and personalise their experiences of digital libraries. The success of the European Digital Library initiative depends in part on the ability to unlock its users abilities to access, manipulate, use and share cultural heritage resources.

Significant amounts of cultural heritage material are now available through online digital library portals. However, this vast amount of cultural heritage material can also be overwhelming for many users who are provided with little or no guidance on how to find and interpret this information. Potentially useful and relevant content is hidden from the users who are typically offered simple keyword-based searching functionality as the entry point into a cultural heritage collection. The situation is very different within traditional mechanisms for viewing cultural heritage (e.g. museums) where items are organized thematically and users guided through the collection. Users of cultural heritage portals have diverse information needs and exhibit highly individualistic information seeking behaviours (e.g. information encountering and foraging) which are not well supported in standard search interfaces. Recent trends in information access services have recognized the necessity of providing support for more exploratory and

serendipitous search behaviours if services are to be effective in helping users with discovering and assimilating knowledge [3, 5, 4].

The PATHS project suggests the metaphor of paths through a collection as a powerful and flexible model for navigation that can enhance the users experience of cultural heritage collections and support them in their learning and information seeking activities. As a result will provide users with innovative ways to access and utilise the contents of digital libraries that enrich their experiences of these resources. The system will make user-specific recommendations about items of potential interest as individuals navigate through the collection. The user will be offered links to information both within and outside the collection that provide contextual and background information, individually tailored to each user and their context. Users can construct their own paths (independent paths) which can be saved for future reference, edited or shared with other users. These paths will be more than a simple list of items in a collection that the user has visited; they will also contain information about the links between the items, details of others items connected to them and connections to information both within and outside the collection that provides context. Groups of users can work collectively to create paths (collaborative paths), adding new routes of discovery and annotations that can build upon the contributions made by others. Users can also follow pre-defined paths (guided paths) created by domain experts, such as scholars or teachers. Guided paths provide an easily accessible entry point to the collection that can be either followed in their entirety or left at any point to create an independent path. Guided paths can be based around any theme, for example artist and media (paintings by Picasso), historic periods (the Cold War), places (Venice), famous people (Muhammed Ali), emotions (happiness), events (the World Cup) or any other topic (e.g. Europe, food).

PATHS will work directly with leading cultural heritage initiatives to advance understanding of user requirements and the research objectives of the project:

- **Alinari 24 ORE SpA**. The Alinari Archives hold a collection of 5,500,000 photographs which is growing at a rate of about 30,000 images each year.
- **Europeana** (http://www.europeana.eu/) is the prototype website of the European digital library. Europeana is incorporating millions of digitised items from Europe's archives, museums, libraries and audio visual collections and providing access through a single portal.

2 Objectives

The PATHS project will take a user-centred approach to development by bringing users into the research cycle from the beginning of the project, gathering their input at all stages in the development on how it can help to meet their needs and feedback on the functionality as prototypes are field-tested. The PATHS project consists of several separate, but connected, packages of work, including the following:

- **Gathering user requirements** and creating functional specifications from a broad range of users including those belonging to different groups, e.g. students, family historians and photographers and of different types, e.g. learning styles and needs from Cultural Heritage collections. These requirements will be used to develop

a functional specification for the systems developed during the project. These requirements will build upon those identified in previous work for cultural heritage information access systems [6, 2].
- **Processing cultural heritage content** and enriching it through identifying connections between items within a collection and complementing connections with existing relations and providing links to material both within and external to the collection that provides background information (e.g. to Wikipedia).
- **Designing effective user interfaces** through which users will interact with the PATHS system. These interfaces will provide users with personalised navigation through Cultural Heritage collections that is enriched with the additional information added through processing the digital content. The user interface will allow users to follow pathways created by other users and to share their own. This will build on previous work on personalisation in museums and digital libraries [7, 1].
- **Designing evaluation methodologies** and conducting of field trials to assess the performance (effectiveness, efficiency and satisfaction) of the systems implemented in PATHS in realistic scenarios. Evaluation will culminate in field trials in end-user scenarios. Particular focus will be on evaluating users search sessions and the value of paths generated by the user. Also, focusing on the evaluation of browsing techniques will form part of this research.

The vision is to build a system that:

- Exploits existing knowledge of users to optimise the effectiveness of interacting with digital heritage resources.
- Enables the testing and refinement of such knowledge.
- Enables new knowledge to be discovered.

This system will provide personalized access to resources by adapting suggested routes to the personalized requirements of individual users and groups. It will seek to:

- Respond to users in a cognitively ergonomic way i.e. by matching navigation to a learners preferred style and minimising any mismatch and consequent additional cognitive processing load. In this way, the learner will find exactly what s/he wants with the least effort. Navigation entails travelling the shortest path between starting point and desired end point.
- Challenge and stretch the user by via controlled and constructive mismatching. In this way, learners may develop increasing autonomy and versatility i.e. the ability autonomously to thrive in information environments not necessarily matched to their own preferred style. PATHS will also explore the extent to which users may be encouraged and helped to engage in cognitive processing in which they are less strong. For example, the extreme divergent thinker may usefully be encouraged (in certain learning circumstances) not to underplay complementary convergent processing. Cognitive research suggests that s/he may, without such complementary processing, exhibit over-generalisation and lack of grasp of detail. Conversely, the strong convergent thinker may be encouraged to explore and think more divergently (creatively) to avoid fragmented learning and failing to see the wood for the trees. PATHS will explore the potential of suggesting sub-optimal, but constructive paths to users.

3 Conclusions

The PATHS project aims to investigate and implement pathways in a naturalistic setting for a range of users and groups that regularly make use of cultural heritage information. A large-scale operational system will developed for navigating on-line cultural heritage collections in a more effective manner than current searching functionalities. Pathways will be used to guide and assist individuals and user communities with information discovery and exploration within cultural heritage information spaces for learning and information seeking activities. This will support multiple information seeking behaviours and enhance the users information access experience of digital library resources.

PATHS will focus on using content from Europeana. The breadth and depth of material provides a challenging data source, together with Europeanas status as a centralised portal for European cultural heritage material. Experimenting with user-adaptivity in this domain will benefit ongoing work on providing semantic enrichment to Europeana content and showcase the kinds of technologies which would make Europeana more accessible to a wider range of users and communities. However, PATHS will also show the generalisability of technologies developed in the project in developing prototype systems for content from Alinari. Contributions made by PATHS will be expected to benefit scholars and citizens alike in providing personalised information access.

4 Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n 270082. We acknowledge the contribution of all project partners involved in PATHS (see: http://www.paths- project.eu).

References

1. Bonis, B., Stamos, J., Vosinakis, S., Andreou, I., Panayiotopoulod, T.: A platform for virtual museums with personalized content. Multimedia tools and applications, 42(2), 139-159 (2009)
2. Clough, P., Ireson, N., Marlow, J.: Extending domain-specific resources to enable semantic access to cultural heritage data. Journal of Digital Information 10(6) (2009)
3. Eaglestone, B., Ford, N., G.Brown, Moore, A.: Information systems and creativity: an empirical study. Journal of Documentation 63(4), 443–464 (2007)
4. Ford, N.: Information retrieval and creativity: towards support for the original thinker. Journal of Documentation 55(5), 528–542 (1999)
5. Foster, A., Ford, N.: Serendipity and information seeking: an empirical study. Journal of Documentation 59(3), 321–340 (2003)
6. Minelli, S., Marlow, J., Clough, P., Recuero, J.C., Gonzalo, J., Oomen, J., Loschiavo, D.: Gathering requirements for multilingual search of audiovisual material in cultural heritage. In: Proceedings of Workshop on User Centricity - state of the art (16th IST Mobile and Wireless Communications Summit). Budapest, Hungary (2007)
7. Schmitz, P., Black, M.: The delphi toolkit: enabling semantic search for museum collections. In: Proceedings of the International Conference for Culture and Heritage Online 2008. Montreal, Quebec, Canada (2008)

Análisis morfológico y sintáctico

Actas IV Jornadas TIMM, pp 75-82

First experiments with developing an unsupervised method for learning morphology of variants

Mans Hulden
University of Helsinki, Language Technology
mans.hulden@helsinki.fi
Iñaki Alegria, Izaskun Etxeberria, Montse Maritxalar
IXA taldea, UPV-EHU
{i.alegria izaskun.etxeberria montse.maritxalar}@ehu.es

Abstract. A long-standing open question in computational morphology is how to combine linguistic and machine-learning approaches. In our work with the Basque language we try to infer a morphological description of variant using the standards description and small standard/variant parallel corpus. The key task is the inference of phonological rules. Although the results obtained in the experiments are encouraging, it seems necessary to improve upon them if we want to use the application for real tools.

1 Introduction

Computational morphology has traditionally been carried out in two ways:

- The linguist approach: experts model a lexicon, paradigms and phonological alternations producing a morphological analyzer/generator. Technology based on finite-state machines (Beesley and Karttunen, 2002) have been the most successful practical implementation of this approach.
- The machine learning approach: from a corpus of the language, sometimes using additional information about paradigms, a program learns a segmentation of word-forms in morphemes. Goldsmith (2001), for example, has proposed a popular method based on this idea.

While the first approach often produces better results and is considered the standard method of building morphological tools, the second approach is sometimes used when development time is at a premium or when experts and linguists are not available for such in-depth work.

An long-standing open question is how to combine both approaches. In our work with the Basque language, a morphological description is available for the standard language, but we want to learn to analyze variants and dialectal forms as well. The hope is that this second part—dealing with dialectal variant forms—could be automatically learned from a corpus given that we have tools to handle the standard language. This is an interesting problem because a good solution to this problem could be applied to many other tasks as well: to improve access to digital libraries (containing diachronic and dialectal variants) or to improve treatment of informal registers such as SMS messages and blogs, etc.

In this paper we assume that a small standard/variant parallel corpus is available (if not, it is possible prepare one) and we propose a method based on finite-state phonology to learn from the information of the corpus and translate a given word of the dialect to its standard-form equivalent. The variant we use for experiments is Lapurdian, a dialect of Basque spoken in the Lapurdi (fr. Labourd) region in the Basque Country.

Because Basque is an agglutinative, highly inflected language, we believe some of the results can be extrapolated to many other languages as well.

One of the motivations for the current work is that there are a large number of NLP tools available and in development for standard Basque (also called Batua): a morphological analyzer, a POS tagger, a dependency analyzer, an MT engine, among others. However, these tools do not work well with the different dialects of Basque and there is a desire to explore the possibility of reusing all or some of these for handling dialect-form input as well.

Here is a brief contrastive example of the kinds of differences found in the dialect (a, Lapurdian) and standard Basque (b) parallel corpus:

(a) Ez gero uste izan **nexkatxa guziek** tu egiten **dautatela**

(b) Ez gero uste izan **neskatxa guztiek** tu egiten **didatela**

As is clear, the differences are minor overall, but even such small discrepancies cause great problems in the potential reuse of current tools designed for the standard forms only.

We have experimented with an approach that attempts to improve on a simple baseline of learning word-pairs in the dialect and the standard. In our approach we have used the *lexdiff* command proposed by Almeida et al. (2010) in their related work on contrasting Brazilian Portuguese and the Portuguese in Portugal. We use *lexdiff* to extract information about the changes between the dialect and the standard, and then induce rules to apply these changes, given input forms in the dialect.

The remainder of the paper is organized as follows. The characteristics of the corpus available to us are described in section 2. In section 3 we describe the steps and variations of the methods we have applied. Section 4 and 5 present the parameters used in the evaluation and the experimental results. Finally, we discuss the results and present possibilities for potential future work in section 6.

2 The corpus

The corpus used in this research have been created as part of "TSABL" project (*Towards a Syntactic Atlas of the Basque Language*, web site: *http://www.iker.cnrs.fr/-tsabl-towards-a-syntactic-atlas-of-.html?lang=fr)*. Two groups are collaborating in this project, the IXA group at the University of the Basque Country and the IKER group in Baiona (fr. Bayonne). The main objective of the IKER group is to analyze and process dialects of Basque language and they have developed an application to analyze syntactic changes between dialects and the standard. This application works with examples they have collected. The researchers of the IKER project have provided us with examples of the Lapurdian dialect and their corresponding forms in standard Basque. Our parallel corpus then consists of running text in two variants: complete sentences of the Lapurdian dialect and equivalent sentences corresponding to standard Basque.

The characteristics of the corpus are presented in Table 1. Our corpus contains 2,117 sentences and 12,150 words for each variant (3,600 different words more or less). So, it can be considered like a parallel corpus of Lapurdian dialect and standard Basque.

In order to provide data for our learning algorithms and also to test their performance, we have divided the corpus into two parts: 80% of the corpus is used for the learning task (1,694 sentences) and the remaining 20% (409 sentences) for evaluation of the learning process.

	Full corpus	**80% part.**	**20% part.**
Sentences	2,117	1,694	423
Words	12,150	9,734	2,417
Unique words	3,610	3,108	1,243
Pairs of diff.	2,169	1,732	437
Unique pairs	1,078	908	301

Table 1. Characteristics of the parallel corpus used for experiments. Last two rows present the number of words that have different spellings in the dialect and in the standard.

3 Methods

We have used different methods to produce an application that will give us the equivalent standard word corresponding to an input word in the dialect.

To extract information from the corpus we use the *lexdiff* command, developed by Almeida et al. (2010). This program comes as part of a toolkit to aid in the automatic adjustment of orthography written in different Portuguese orthographies (Brazilian/European). Despite its focus on Portuguese, the toolkit is designed to be somewhat language-independent and relies produces a set of orthographic adjustment rules that are learned by comparing different corpora. The rule induction component (*lexdiff*) is what we have used in our research.

The overall process consists of three steps:

1. Apply *lexdiff* to the parallel corpus to obtain information about correspondences between the variant and the standard. With this pre-processing we can obtain a list of equivalent words or a list of equivalent n-grams with their frequency in the corpus.
2. Use previous information to "learn" phonological rules. There are many options to learn and we have to experiment with them to see how they modify the results.
3. Constrain the output of the rules learned by *lexdiff* to words in the Basque standard morphology.

3.1 The baseline

The baseline of our experiments is a simple method, based on a dictionary of equivalent words with the list of correspondences between words extracted from the 80% of the corpus with *lexdiff* command. This list of correspondences contains all aligned words in the variant vs. standard corpus, be they identical or not. For example, the output `112 eman = eman` indicates that the correspondence is between the same form and appears 112 times in the corpus while the entry `61 emaiten => ematen` indicates that the correspondence is between different forms and appears 61 times in the corpus.

In other words, the baseline approach is simply to memorize all the word pairs seen between the dialectal and standard forms, and subsequently use this knowledge in later conversion tasks.

3.2 Method 1

The second approximation is to infer phonological rules from the equivalences between words obtained with *lexdiff* and compile these rules into finite-state transducers (we use the freely available *foma* toolkit for this (Hulden, 2009)).

The *lexdiff* program tries to identify sequences of changes from seen word pairs and outputs string correspondences such as, for example: `76 ait => at ; 39 dautz => diz`, indicating that `ait` has changed into `at` 76 times in the corpus, etc.

With such information about word pairs we generate a variety of so-called replacement rules which can subsequently be compiled into finite transducers with the *foma* application. Even though the *lexdiff* program provides a direct string-to-string change as a rule, there are several ways to encapsulate its output as replacement rules and finite transducers, yielding variant approaches such as the following:

- We can restrict the rules by frequency and require that a certain type of change be seen at least n times in order to apply that rule. For example, if we set this threshold to 3, we will only apply a string-to-string changing rule that has been seen three times or more.
- We limit the number of rules that can be applied to the same word. Sometimes the *lexdiff* application divides the change between a pair of words into two separate rules. For example the word-word correspondence `agerkuntza => agerpena` is expressed by two rules: `rkun => rpen` and `ntza => na`. Now, given these two rules, we have to be able to apply both to produce the correct total change `agerkuntza => agerpena`. By limiting the number of rules that can apply to a single input word we can avoid creating many spurious outputs, but also at the same time we may sacrifice some ability to produce the desired output forms.
- We can also control the application mode of the rules: whether they be sequential or parallel. The rules in *foma* can be applied in parallel or sequentially and the results are different depending on the mode of application. The previous example can serve to illustrate the difference between two modes. If the previous two rules are applied in parallel, the form obtained from `agerkuntza` will not be correct since the *n* overlaps with the two rules. That is, when applying rules simultaneously

in parallel, the input characters for two rules may not overlap. However, if these two rules applied in sequence (the order in this example is irrelevant), the output will be the correct: we first change `rkun => rpen` and later `ntza => na`. We have not a priori chosen to use parallel or sequential rules and have decided to evaluate both approaches.

– We can also compact the rules output by *lexdiff* by eliminating redundancies and constructing context-sensitive rules. For example: given a rule such as `rkun => rpen`, we can convert this into a context-sensitive rule that only changes `ku` into `pe` when flanked by `r` and `n` to the left and right, respectively. This has a bearing on the previous point and will allow more rewritings within a single word in parallel replacement mode since there are less characters overlapping.

4 Evaluation

We have measured the quality of different approaches by the usual parameters of precision, recall and the harmonic combination of them, the F1-score. We have analyzed how the different options in the approaches affect the results of these three parameters. Given that we extract quite a large number of rules and that each input word generates a very large number of candidates if we use all the rules extracted, it is possible to produce a high recall on the conversion of unknown dialect words to the standard form. However, the downside is that this naturally leads to low precision as well, which we try to control by introducing a number of filters to remove some of the candidates output by the rules. More specifically we use two filters: (1) an obligatory filter which removes all candidate words that are not found in the standard Basque (by using an existing standard Basque morphological analyzer), and (2) using an optional filter which, given several candidates in the standard Basque, picks the most frequently occurring one.

5 Results

As mentioned, the learning process has been done using the 80% of the corpus, leaving 20% of the corpus for evaluation of the abovementioned approaches. In the evaluation, we have only tested those words in the dialect that *differ* from words in the standard (which are in the minority). In total, in the evaluation part, we have tested 301 words.

The results for the baseline—i.e. simple memorization of word-word correspondences—are (in %): P = 95.62, R = 43.52 and F1 = 59.82. As expected, the precision of the baseline is high: when the method gives an answer it is usually the correct one. But the recall of the baseline is naturally low: slightly less than half of the words in the evaluation corpus have been encountered before.

In the following, we give the results of a number of experiments using method 1.

5.1 Results depending on a frequency threshold

Varying the frequency threshold (see 3.2), we have tested with values of 1, 2, and 3. The values are in table 2. The results clearly show that the more information we extract

(frequency 1), the better results we obtain for recall while at the same time the precision suffers. The F-score doesn't vary very much and it maintains similar values throughout.

If we compare the results presented in Table 3 with the results of the baseline, it is obvious that the baseline is 9-10 points better with respect to the F-score: the precision of the baseline is very high compared to the precision of our first approach; on the other hand, the recall of the baseline is worse, but not significantly. The problem with this approach is one which we have mentioned before: the rules produce more than one answer for any given word and the consequence is that the precision suffers, even though only those output words are retained that correspond to actual standard Basque. With the frequency filter in place, the results improve somewhat[1]. The filtered results are given in table 3: with this addition, we can improve on the baseline, but not significantly.

	P	**R**	**F**
Baseline	95.62	43.52	59.82
Freq. 1	38.95	66.78	49.20
Freq. 2	46.99	57.14	51.57
Freq. 3	49.39	53.82	51.51

Table 2. Values obtained for Precision, Recall and F-scores by changing the minimum frequency of the correspondences to construct rules for *foma*. The rest of the options are the same in all three experiments: only one rule is applied in a word, and without context.

	P	**R**	**F**
Baseline	95.62	43.52	59.82
Freq. 1	70.28	58.13	63.64
Freq. 2	70.18	53.16	60.49
Freq. 3	71.76	51.50	59.96

Table 3. Values obtained for Precision, Recall and F-score by changing the threshold frequency of correspondences and applying a post-filter.

5.2 Results depending on other options

We have also varied the maximum number of possible rule applications within a single word by limiting it to 1 and 2 as well as applying the rules in parallel or sequentially, and

[1] These frequencies were obtained from a corpus of a Basque newspaper

compacting the rules to provide more context-sensitivity. We shall here limit ourselves to present the best results of all these options in terms of the F-score in table 4.

In general, we may note that applying more than one rule has a negative effect on the precision and does not help the recall very much either. Applying the post-filter—choosing the most frequent candidate—yields a limited improvement: mildly better precision but also slightly worse recall, and the F-score does not improve significantly. The parallel or sequential application of the rules (when they are more than one) doesn't change the results very much and if we analyze the F-score, it seems better to apply the rules in parallel. Finally, compacting the rules and producing context-sensitive ones is clearly the best option.

In all cases the F-score improves if the frequency filter is applied; sometimes significantly and sometimes only slightly. All the results of the table 4 which lists the best performing ones come from experiments where the frequency filter was applied.

	P	**R**	**F**
Baseline	95.62	43.52	59.82
EXP. 1	72.20	57.81	64.21
EXP. 2	72.13	58.47	64.59
EXP. 3	75.10	60.13	66.79

Table 4. EXP. 1: frequency 2; 2 rules applied; in parallel; without context. **EXP. 2**: frequency 1; 1 rule applied; with context. **EXP. 3**: frequency 2; 2 rules applied; in parallel; with context.

6 Conclusions and future work

We have presented a number of experiments to solve a very concrete task: given a word in the Lapurdian dialect of Basque, produce the equivalent standard Basque word. The approach has been based on the idea of extracting string-to-string changing rules from a parallel corpus of the two dialects, and to apply these rules to unseen words. We have been able to improve on the results of a naive baseline using a method which infers phonological rules of the information extracted from the corpus and applies them using finite state technology.

Although the results obtained in the experiments are encouraging, it seems necessary to improve upon them if we want to use the application for real tools. In particular, the overgeneration of the learned rules remains a problem and leads to low precision. We are working on other algorithms for string-to-string pattern induction based *Inductive Logic Learning*-type algorithms with the goal to limit this overgeneralization and still produce high recall. During the current work, however, we have accumulated a small but valuable training and test corpus which may serve as a future workbench for evaluation of phonological rule induction algorithms.

Bibliography

Almeida, J. J., Santos, A., and Simoes, A. (2010). Bigorna–a toolkit for orthography migration challenges. In *Seventh International Conference on Language Resources and Evaluation (LREC2010), Valletta, Malta*.

Beesley, K. R. and Karttunen, L. (2002). Finite-state morphology: Xerox tools and techniques. *Studies in Natural Language Processing. Cambridge University Press*.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

Hulden, M. (2009). Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32, Athens, Greece. Association for Computational Linguistics.