Clasificación de Distintos Conjuntos de Datos Utilizados en Evaluación de Métodos de Extracción de Conocimiento Creados para la Web

Juan M. Rodríguez[1,2], Hernán D. Merlino[2], Patricia Pesado[1], Ramón García-Martínez[2]

[1] Programa de Doctorado en Ciencias Informáticas. Facultad de Informática. Universidad Nacional de La Plata. Argentina.
[2] Grupo de Investigación en Sistemas de Información. Departamento de Desarrollo Productivo y Tecnológico. Universidad Nacional de Lanús. Argentina.
jmrodriguez1982@gmail.com, hmerlino@gmail.com, ppesado@lidi.info.unlp.edu.ar, rgm1960@yahoo.com

Resumen. En varios artículos se han utilizado distintos textos de prueba, como datos de entrada para medir el desempeño de los métodos de extracción de relaciones semánticas para la Web (OIE): ReVerb y ClausIE. Sin embargo estos textos nunca han sido analizados para entender si ellos guardan o no similitudes o para saber si existe entre ellos un lenguaje común o pertenecen a un mismo dominio. Es la intención de este trabajo analizar dichos textos utilizando distintos algoritmos de clasificación. Y comprender si se pueden agrupar de una forma coherente, de tal suerte que a priori uno pueda identificar que textos son los que trabajan mejor con ClausIE y cuales con ReVerb.

Palabras Clave. Extracción de conocimiento, extracción de relaciones semánticas, métodos de extracción auto-supervisados, *open information extraction*, procesamiento de lenguaje natural, clasificación de textos, Bayes Naive, SMO, J48.

1 Introducción

En [Rodríguez et al., 2015] fue realizada una investigación documental sobre distintos métodos de extracción de relaciones semánticas para la Web (*Open Information Extraction*, en inglés según la definición dada en [Banko et al., 2007]). En dicha investigación se relevaron nueve métodos de extracción de relaciones semánticas utilizando comparaciones y experimentos publicados por distintos autores. El resultado del trabajo de [Rodríguez et al., 2015] es el trazado de una línea evolutiva de estos distintos métodos en función del tiempo con la conclusión de que el método de extracción de relaciones semánticas para la Web más preciso (ver fórmula 1) es ClausIE (propuesto en [Del Corro & Gemulla, 2013]), seguido por OLLIE (propuesto en [Schmitz, 2012]), y en tercer lugar por ReVerb (propuesto en [Fader et al., 2011]). Esta conclusión estuvo basada principalmente en los resultados de los experimentos publicados en los artículos: [Del Corro & Gemulla, 2013; Schmitz et al., 2012].

En [Rodríguez et al., 2016] se realizó un experimento para verificar el supuesto anterior sobre la precisión de los distintos métodos. Para ello se utilizó un conjunto de

datos de entrada creado con 55 cables de noticias elegidos de forma aleatoria de la base de datos Reuters-21578 [Lewis, 1997]. El resultado de dicho experimento se resume en la Tabla 1.

Tabla 1. Precisión, Exhaustividad (*recall*) y Medida-F1

Método	Precisión	Exhaustividad	Medida-F1
ClausIE	0.513	**0.503**	**0.508**
ReVerb	**0.671**	0.355	0.464
OLLIE	0.488	0.420	0.451

Contrariamente a lo esperado el método más preciso resultó ser ReVerb, para este conjunto de datos de entrada. El experimento incluyó además la medición de la Exhaustividad (*recall* en inglés) y una medida que combina ambas (Medida-F1). Las fórmulas de estas tres medidas de rendimiento se dan a continuación:

$$\text{Precisión} = \frac{\text{cantidad de piezas de conocimiento extraídas correctamente}}{\text{cantidad\_de\_piezas\_de\_conocimiento\_extraidas}} \quad (1)$$

$$\text{Exhaustividad} = \frac{\text{cantidad de piezas de conocimiento extraídas correctamente}}{\text{cantidad\_de\_piezas\_de\_conocimiento\_totales\_en\_el\_documento}} \quad (2)$$

$$F1 = \frac{2 \cdot \text{precisión} \cdot \text{exhaustividad}}{\text{precisión} + \text{exhaustividad}} \quad (3)$$

En donde una "pieza de conocimiento" refiere de forma genérica a una relación semántica, ya que en términos más generales una relación semántica es una representación estructurada (manipulable en procesos de razonamiento automático) del conocimiento embebido en una fuente de datos, en principio no estructurada, como lo es el lenguaje natural [García-Martínez & Britos, 2004; Gómez et al., 1997].

En la fórmula 2, para calcular la cantidad total de piezas de conocimiento (extraídas y no extraídas), se realizó una extracción manual de las relaciones semánticas halladas en cada cable de noticias y se sumaron además las extracciones correctas realizadas por el algoritmo que no habían sido descubiertas durante la extracción manual.

1.1 Hipótesis

Los 55 cables de noticias evaluados permiten afirmar que Reverb es el método más preciso para extraer relaciones semánticas de la base Reuters-21578, con un nivel de confianza del 86%, considerando un margen de error del 10% [Rodríguez et al., 2016]. Sin embargo en los experimentos realizados en [Del Corro & Gemulla, 2013], en donde se utilizaron tres conjuntos de datos diferentes: 200 oraciones extraídas aleatoriamente del *New York Times collection* [Sandhaus E., 2008], 200 oraciones extraídas aleatoriamente de páginas de Wikipedia y 500 oraciones tomadas del servicio *Yahoo's random link* (este es el conjunto de datos utilizados originalmente para probar ReVerb en [Fader et al., 2011]) se obtuvo como resultado que ClausIE es un método más preciso que ReVerb.

A partir estos resultados, incompatibles entre sí, proponemos cómo hipótesis de investigación que el conjunto de datos de entrada, es decir el dominio al cual pertenece un texto escrito en lenguaje natural, es relevante a la hora de extraer las relaciones semánticas embebidas en él. Para probar esta hipótesis será necesario encontrar una forma de catalogar o clasificar los textos de entrada a partir de una o varias de sus características.

2 Experimentación

La demostración de la hipótesis propuesta, fue abordada como un problema de clasificación textos. Para ello se utilizaron los siguientes algoritmos de clasificación: Bayes Naïv, Bayes Naïv Multinomial, SMO y J48. Los cuatro algoritmos son parte de Weka [Hall et al., 2009], la herramienta de investigación y desarrollo utilizada en el presente trabajo.

2.1 Bayes Naïve y Bayes Naïve Multinomial

Bayes Naïve es uno de los modelos más simples y parte del supuesto de que todos los atributos de un documento de entrada (por ejemplo unigramas) son independientes entre sí en el contexto de una clase, esto es llamado "el supuesto de *Naïve Bayes*" [Mccallumzy et al., 1998]. A pesar de que este supuesto es en verdad falso, en la mayoría de las tareas del mundo real *Naïve Bayes* realiza buenas clasificaciones [Mccallumzy et al., 1998].

Este algoritmo de clasificación busca identificar a la clase que maximice el resultado de la multiplicación entre la probabilidad de una clase dada y las probabilidades individuales de las palabras dada dicha clase, matemáticamente:

$$C_{map} = \operatorname{argmax} P(c_i) \prod_{x \in X} P(w|c_i) \quad (4)$$

Las probabilidades de las distintas clases (c) junto con las probabilidades de cada una de las palabras (w) de pertenecer a una clase son estimadas en el conjunto de entrenamiento.

Una de las razones por las cuales fue seleccionado *Naïve Bayes* como algoritmo de clasificación, es por el trabajo de [Banko & Brill, 2001], en donde se utilizaron tres clasificadores distintos: (a) *Winnow* [Golding & Roth, 1999], (b) *Perceptron* [NG et al., 1997] y (c) *Naïve Bayes*, se observó que estos tienden a converger para grandes volúmenes de datos. Otro de los motivos es que *Naïve Bayes* es un algoritmo rápido, incluso con grandes cantidades de datos [Lowd & Domingos, 2005].

Por su parte la implementación de *Bayes Naïve Multinomial* que viene incluida en la herramienta WEKA es una mejora respecto al algoritmo tradicional de *Bayes Naïve* que consiste en lo siguiente:

- No considera de igual forma la ocurrencia de una palabra como su ausencia. Usa la probabilidad de la misma.
- Tiene en cuenta las múltiples repeticiones de una palaba en un documento
- No trata a todas las palabras de la misma forma, discrimina entre palabra más frecuentes y menos frecuentes, etc.

Ambos algoritmos fueron testeados contra un mismo conjunto de documentos de prueba extraídos de Reuters-21578 en [Hall et al., 2009] y el resultado fue que *Bayes Naïve Multinomial* obtuvo una precisión de 0.91 contra una precisión de 0.8 que obtuvo el algoritmo original *Bayes Naïve*.

2.2 SMO

El segundo método escogido para realizar la clasificación de textos y poder comparar resultados fue una implementación de *support vector machines* (SVMs) [Joachims, 1998] llamada *Sequential Minimal Optimization* (SMO): la cual consiste en una mejora en el algoritmo de entrenamiento de SVMs, de forma tal que este llega a ser 1200 veces más rápido para SVMs lineales y 15 veces más rápido para SVMs no lineales [Platt, 1998].

Las SVMs como método de clasificación son muy populares y ampliamente utilizadas [Kolesov et al., 2014] debido a su éxito, no solo para clasificar textos sino también para diversos problemas de clasificación. Sang-Bum Kim en [Kim et al., 2006] menciona que los clasificadores basados en complejos métodos de aprendizaje como los SVMs pertenecen al *state-of-the-art*.

Sequential Minimal Optimization (SMO) es un algoritmo simple que puede resolver rápidamente los grandes problemas de programación cuadrática (QP) de SVM sin utilizar una matriz de almacenamiento y sin utilizar pasos QP de optimización numérica. SMO descompone al problema QP general en sub-problemas QP, utilizando el teorema de Osuna para asegurar la convergencia [Platt, 1998].

2.3 J48

J48 es una implementación de código abierto del algoritmo C4.5 incluida en la herramienta Weka. C4.5 a su vez es un programa que crea arboles de decisión basados en un conjunto de datos de entrada previamente etiquetados. El algoritmo fue desarrollado por Ross Quinlan. Los arboles de decisión generados por el algoritmo C4.5 pueden ser utilizados para clasificación y suelen ser llamados clasificadores estadísticos [Gholap, 2012]. Este es un algoritmo es utilizado ampliamente para resolver problemas de aprendizaje automático [Witten et al., 1999].

El algoritmo trabaja partiendo la lista inicial de ejemplos etiquetados según aquellos atributos que más eficazmente dividen el conjunto inicial, utilizando para ello el concepto de entropía de información [Quinlan, 2014]. Aplicando esta técnica de forma recursiva por cada sub lista el algoritmo va construyendo un árbol.

2.4 Conjunto de datos de entrada

Para constituir el conjunto de datos de entrada se tomaron la totalidad de los datos utilizados para evaluar a los clasificadores y se construyó un primer gran conjunto de datos:

- Subconjunto de 55 cables de noticias la base Reuters-21578
- 200 oraciones extraídas aleatoriamente del *New York Times collection*
- 200 oraciones extraídas aleatoriamente de páginas de Wikipedia

- 500 oraciones tomadas del servicio *Yahoo's random link*

El conjunto anterior se utilizó para crear dos conjuntos: el conjunto "ReVerb" y el conjunto "ClausIE". El conjunto "ReVerb" se conformó con todos aquellos textos del conjunto anterior, tal que, al ser utilizados como entrada de los métodos ReVerb y ClausIE, ReVerb extrajo relaciones semánticas con una precisión al menos 50% superior a ClausIE. A su vez el conjunto "ClausIE" quedó conformado con todos aquellos textos en los cuales ClausIE logró una precisión al menos 50% superior a ReVerb en la extracción de relaciones semánticas.

El conjunto "ClausIE" quedó constituido por 124 casos y el conjunto "ReVerb" por 58 casos. Cómo es posible observar los casos en los cuales ClausIE es más preciso que ReVerb son mayoría. Hay que tener en cuenta que la mayoría de los casos son los mismos que fueron utilizados en el trabajo de [Del Corro & Gemulla, 2013].

Además el texto original en idioma inglés fue convertido palabra por palabra a sus categorías gramaticales (*PosTags*) y también se lo convirtió en etiquetas de IOB [Ramshaw, 1995], según la técnica de *text-chunkin. L*a técnica de *text-chunking*, consiste en dividir frases en segmentos de texto que no se superponen, en base a un análisis superficial. En [Abney, 1991] se propuso este método como un precursor útil y simple de implementar para detectar principalmente frases nominales y verbales [Ramshaw, 1995]. Para convertir una porción de texto en sus categorías gramaticales o bien para obtener las etiquetas de IOB correspondientes al *text-chunking* se utilizó el mismo programa ReVerb, solo que se lo modificó para que convierta el texto de la forma mencionada utilizando las mismas librerías que utiliza para extraer relaciones semánticas.

Para ilustrar lo anterior supongamos el siguiente texto en inglés como ejemplo:

She has done so with depth and confidence.

Convertido a categorías gramaticales (*pos-tags*) quedó como:

- PRP VBZ VBN RB IN NN CC NN .

Convertido a etiquetas IOB de al *text-chunking* quedó como:

- B-NP B-VP I-VP B-ADVP B-PP B-NP I-NP I-NP O

Además, tanto el texto en lenguaje natural, como las oraciones convertidas a categorías gramaticales y a etiquetas IOB de *text-chunking* fue por un lado clasificado tal cual, es decir como unigramas, pero fue convertido también a bigramas y trigramas. Siguiendo con el ejemplo anterior, la oración:

She has done so with depth and confidence.

Fue clasificada, según las siguientes maneras de agrupar palabras:

1. **Unigramas**: (*She, has, done, so, with, depth, and, confidence*)
2. **Bigramas**: (*<start>-She, She-has, has-done, done-so, so-with, with-depth, depth-and, and-confidence, confidence-<end>*)
3. **Trigramas**: (*<start>-She-has, She-has-done, has-done-so, done-so-with, so-with-depth, with-depth-and, depth-and-confidence, and-confidence-<end>*)

Este mismo tratamiento se le dio a la oración como categorías gramaticales y etiquetas IOB de *text-chunking*. Cada uno de los 4 métodos de clasificación fue entrenado 9 veces, una vez por cada conjunto de características: unigramas, bigramas y trigramas para el texto en lenguaje natural, luego unigramas, bigramas y trigramas para su versión en categorías gramaticales y nuevamente unigramas, bigramas y trigramas sobre las etiquetas IOB de *text-chunking*. De esta forma se intentó encontrar que características en un texto son más relevantes para que un método de extracción de relaciones semánticas actúe mejor que otro.

3 Resultados

A continuación se muestran los resultados en tres tablas distintas, en la primera tabla se muestran los resultados para el texto en lenguaje natural, en la segunda tabla los resultados para el texto convertido a sus respectivas categorías gramaticales y por último el texto convertido a etiquetas IOB de *text-chunking*. A su vez cada tabla muestra, para cada uno de los algoritmos de clasificación de texto seleccionados, un porcentaje de las instancias clasificadas correctamente en cada uno de los conjuntos de características de un texto. En todos los casos se utilizó el 66% de los datos para entrenar el algoritmo y el resto para validar el mismo.

Tabla 2. Texto en ingles

Clasificador	Unigramas	Bigramas	Trigramas
Bayes Naïve	**77.4194**	67.7419	70.9677
Bayes Naïve Multinomial	70.9677	51.6129	32.2581
SMO	64.5161	70.9677	69.3548
J48	53.2258	70.9677	69.3548

Tabla 3. Categorias gramaticales

Clasificador	Unigramas	Bigramas	Trigramas
Bayes Naïve	56.45	62.9	53.23
Bayes Naïve Multinomial	64.51	67.74	54.84
SMO	67.74	59.68	59.68
J48	54.84	**72.58**	58.06

Tabla 4. Etiquetas IOB de *text-chunking*

Clasificador	Unigramas	Bigramas	Trigramas
Bayes Naïve	62.90	61.29	59.68
Bayes Naïve Multinomial	**69.35**	61.29	66.13
SMO	67.74	58.06	59.68
J48	**69.35**	67.74	59.68

4 Conclusiones

De la tabla 4 se deduce que convertir un texto a etiquetas de *text-chunking*, técnica que utiliza ReVerb para realizar extracciones semánticas, no da buenos indicios acerca de si el texto será bien interpretado por ReVerb o no, tampoco por ClausIE. El porcentaje de 69.35 es muy bajo si tenemos en cuenta que solo se trabajó con dos conjuntos: ReVerb y ClausIE (ver punto 2.4). Menos de 20 puntos por encima de una clasificación completamente aleatoria. El valor más alto se registró en el conjunto de unigramas. Los bigramas y trigramas, es decir las secuencias de etiquetas de *text-chunking* aportan menos información que los simples unigramas.

La tabla 3 muestra un valor un poco más alto para el porcentaje de clasificaciones positivas. Las categorías gramaticales tomadas como bigramas, logran ser clasificadas con un porcentaje de aciertos de: 72.58%. La secuencia de categorías gramaticales (*PosTags*) son un indicio, aunque muy leve, acerca de qué método (y por ello de qué técnica) de extracción de relaciones semántica será más adecuada para un texto dado

De la tabla 2 se desprende que es posible identificar, con una precisión mayor al 77%, si un texto será mejor analizado por ClausIE o bien ReVerb. Es decir que las relaciones semánticas presentes en un texto dado, podrán ser extraídas de forma más precisa con uno u otro método y eso solo depende de los unigramas presentes. Dado que los unigramas coinciden con las palabras, (la *tokenizacion* utilizada para crear los unigramas es la misma que utilizan ambos métodos para separar las palabras en una oración dada), es posible afirmar en el 77% de los casos, que si determinadas palabras están presentes en un texto entonces dicho texto será mejor analizado ReVerb (o bien por ClausIE según sea el caso).

A partir de lo anterior se puede concluir que:

- El dominio al cual pertenece un texto será crucial para saber cómo reaccionará cualquiera de los dos métodos de extracción de relaciones semánticas analizados.
- La secuencia de categorías gramaticales tiene una ligera influencia en el proceder de ambos métodos.
- Las secuencias de etiquetas IOB de *text-chunking* no dan mucha información acerca de cómo el texto será interpretado por cualquiera de los métodos.

5 Futuras líneas de investigación

A partir de las conclusiones extraídas en el punto 4, queda pendiente el trabajo de aplicar un algoritmos de clasificación de textos cómo un paso previo a la extracción de relaciones semánticas. Según el resultado de este algoritmo debería ser utilizado ClausIE o ReVerb. Si los supuestos en el punto 4 son correctos se espera una mejora notable en la calidad de las piezas de conocimiento extraídas en comparación con la que obtendrían los métodos trabajando por separado.

Financiamiento

Las investigaciones que se reportan en este artículo han sido financiadas parcialmente por los Proyectos de Investigación 33B177 y 33A205 de la Secretaria de Ciencia y Técnica de la Universidad Nacional de Lanús (Argentina).

Referencias

Abney, S. P. (1991). Parsing by chunks (pp. 257-278). Springer Netherlands.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007, January). Open information extraction for the web. In IJCAI (Vol. 7, pp. 2670-2676).

Banko, Michele; Brill, Eric. Scaling to very very large corpora for natural language disambiguation (2001). En Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2001. p. 26-33.

Del Corro, L., & Gemulla, R. (2013, May). ClausIE: clause-based open information extraction. In Proceedings of the 22nd international conference on World Wide Web (pp. 355-366). International World Wide Web Conferences Steering Committee.

Evan Sandhaus. The New York Times Annotated Corpus, 2008.

Fader, A., Soderland, S., & Etzioni, O. (2011, July). Identifying relations for open information extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1535-1545). Association for Computational Linguistics.

García-Martínez, R. & Britos, P. V. (2004). Ingeniería de sistemas expertos. Nueva Librería. ISBN 987-1104-15

Gholap, J. (2012). Performance tuning of J48 Algorithm for prediction of soil fertility. arXiv preprint arXiv:1208.3943.

Golding, Andrew R.; Roth, Dan. A winnow-based approach to context-sensitive spelling correction. Machine learning, 1999, vol. 34, no 1-3, p. 107-130.

Gómez, A., Juristo, N., Montes, C., & Pazos, J. (1997). Ingeniería del conocimiento. Editorial Centro de Estudios Ramón Areces. ISBN 84-8004-269-9.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann P., Witten, I.H. (2009). The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features (pp. 137-142). Springer Berlin Heidelberg.

Kim, Sang-Bum, et al. Some effective techniques for naive bayes text classification. Knowledge and Data Engineering, IEEE Transactions on, 2006, vol. 18, no 11, p. 1457-1466.

Kolesov, Anton, et al. On Multilabel Classification Methods of Incompletely Labeled Biomedical Text Data. Computational and Mathematical Methods in Medicine, 2014, vol. 2014.

Lewis, D. (1997). Reuters-21578 text categorization test collection, http://goo.gl/NrOfu, página vigente al 10/09/2016

Lowd, Daniel; Domingos, Pedro. Naïve Bayes models for probability estimation. En Proceedings of the 22nd international conference on Machine learning. ACM, 2005. p. 529-536.

Mccallumzy, Andrew; Nigamy, Kamal. A comparison of event models for Naïve bayes text classification. En AAAI-98 workshop on learning for text categorization. 1998. p. 41-48.

NG, Hwee Tou; GOH, Wei Boon; LOW, Kok Leong. Feature selection, perceptron learning, and a usability case study for text categorization. En ACM SIGIR Forum. ACM, 1997. p. 67-73.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines

Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

Ramshaw, L. A., & Marcus, M. P. (1995). Text chunking using transformation-based learning. arXiv preprint cmp-lg/9505040. Vigente al 10/09/2016.

Rodríguez, J. M., Merlino, H., García-Martínez, R. (2015). Revisión Sistemática Comparativa de Evolución de Métodos de Extracción de Conocimiento para la Web. XXI Congreso Argentino de Ciencias de la Computación (CACIC 2015). Buenos Aires, Argentina.

Rodríguez, J. M., Merlino, H. D., Pesado, P., & García-Martínez, R. (2016, August). Performance Evaluation of Knowledge Extraction Methods. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 16-22). Springer International Publishing.

Schmitz, M., Bart, R., Soderland, S., & Etzioni, O. (2012, July). Open language learning for information extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 523-534). Association for Computational Linguistics.

Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.