Procesamiento del Lenguaje Natural, Revista nº 59, septiembre de 2017, pp. 129-132 recibido 29-03-2017 revisado 12-05-2017 aceptado 25-05-2017

Constructor automático de modelos de dominios sin corpus preexistente

Automatic constructor of domain models without pre-existing corpus

Edwin A. Puertas Del Castillo, Jorge A. Alvarado Valencia, Alexandra Pomares Quimbaya
Pontificia Universidad Javeriana
Carrera 7 No. 40 – 62, Bogotá D.C., Colombia
{edwin.puertas, jorge.alvarado, pomares} @javeriana.edu.co

Resumen: En este proyecto se presenta un constructor automático de modelos de dominios de conocimientos de forma automática sin corpus preexistente para describir semánticamente un contexto. El constructor está basado en técnicas y métodos para la construcción de corpus a partir de fuentes digitales, mediante el desarrollo de librerías de software que automaticen las fases del sistema propuesto. Este proyecto se encuentra en fases de pruebas conceptuales y desarrollo de componentes.
Palabras clave: Construcción de dominios, dominio del conocimiento, lingüística computacional, comprensión de lenguaje natural.

Abstract: This project is about an automatic builder of domain models without pre-existing corpus. The constructor is based on techniques and methods for the construction of corpus from data extracted from digital media, through the use and development of software libraries that automate the phases of the process of building domains. This project is in phases of conceptual testing and component development
Keywords: Construction of domains, knowledge domain, computational linguistics, natural language comprehension.

1 Introducción

La naturaleza no estructurada de los textos hace necesario contar con modelos de dominio que favorezcan la precisión y consistencia en el procesamiento de este tipo de datos a un bajo costo (Villayandre Llamazares, 2008). Poseer el dominio de conocimiento asociado a un texto específico es difícil debido a la diversidad y origen de las fuentes de información (Schreiber, 2000). Este problema es aún más complejo si no se tiene un corpus preexistente. En consecuencia, existe la necesidad de construir un sistema que automatice la construcción de modelos de dominios específicos de conocimiento sin la necesidad de contar con un corpus preexistente, utilizando fuentes de información como enciclopedias en línea, páginas web, blogs y *Rich Site Summary* (RSS) (Board, 2007).

Viendo esta necesidad el Centro de Excelencia y Apropiación en Big Data y Data Analytics (CAOBA) (Alianza CAOBA, 2017) creó el proyecto titulado: *Constructor automático de modelos de dominio sin corpus preexistente*, el cual tiene el propósito de avanzar en el área de la Lingüística Computacional (LC), la Minería de Textos (MT) y la Ingeniería de Software (IS), además de enfrentar y dar soluciones a nuevos retos que plantea el uso de la lengua en medios digitales.

En el campo de la LC se identifican y se utilizan técnicas y métodos para la construcción de corpus a partir de datos extraídos de medios digitales. Adicionalmente, en MT, el enfoque propuesto es generar herramientas que automaticen la extracción de información de medios digitales basados principalmente en extracción de texto de manera eficiente y confiable. Finalmente, en IS, el propósito es facilitar la integración con otros componentes de software y futuros proyectos utilizando estándares y tecnologías orientados a la web (W3C, 2017), además de lenguajes de

ISSN 1135-5948 © 2017 Sociedad Española para el Procesamiento del Lenguaje Natural

programación multiparadigma como Python (Python, 2017).

Este artículo está organizado de la siguiente manera: se establece el objetivo principal del proyecto, seguido por la metodología empleada, la descripción del sistema, los avances desarrollados hasta la fecha, y finalmente, los resultados esperados.

2 Objetivo

El objetivo de este proyecto es desarrollar un sistema que permita construir modelos de dominios de conocimiento de forma automática sin corpus preexistente para describir semánticamente un contexto. Igualmente, se espera profundizar en métodos y técnicas en áreas de LC, MT e IS para fortalecer las líneas de investigación del centro de excelencia.

3 Metodología

El desarrollo de este proyecto se fundamenta en la técnica *Design Science Research in Information Systems* desarrollada por (Vaishnavi y Kuechler, 2004), la cual consiste en el diseño de una secuencia de actividades por parte de un experto que produce un artefacto innovador y útil para un problema en particular. El artefacto debe ser evaluado con el fin de asegurar su utilidad para el problema especificado y debe contribuir de forma novedosa a la investigación; además, debe resolver un problema que aún no ha sido resuelto o proporcionar una solución más eficaz. A continuación, se describen detalladamente las fases que forman parte del proceso de construcción de modelos de dominio sin corpus preexistente, representado en la Figura 1.

- Búsqueda y recuperación de información: en esta fase se identifican artículos en Wikipedia y páginas Web relacionadas con un dominio en particular mediante el uso de librerías públicas (API's). Según los autores (Arnold y Rahm, 2015) la obtención de información de los artículos en Wikipedia en un tema en particular se realiza mediante el cálculo de las regiones de dominio en la cual se identifican los artículos adyacentes al artículo inicial. Para el caso del constructor de modelos de dominio el artículo inicial es el documento semilla del cual se quiere extraer el dominio. Igualmente, para las páginas Web, los autores (Shi, Liu, Shen, Yuan, y Huang, 2015) proponen el análisis y la extracción de textos, mediante la detección adyacente de todos los conjuntos de registros similares del árbol Document Object Model (DOM) (Nicol, Wood, Champion, y Byrne, 2001).
- Análisis de la extracción: aquí se determina la calidad de la extracción mediante el uso de métricas como *F-measure* (Powers, 2011), precisión y exhaustividad (Zhu, 2004), utilizando la colección de artículos y páginas Web identificados en la búsqueda de la fase anterior, con la finalidad de establecer la relevancia de la información extraída.
- Normalización: en esta fase se realiza el proceso de preparación de información utilizando reconocimiento de caracteres especiales, textos de otros idiomas y *stopwords* (Leskovec, Rajaraman, y Ullman, 2014).
- Generación de reglas para el dominio: En esta fase se definen reglas de asociación para una mejor precisión y exhaustividad en la detección de dominios mediante la segmentación, el análisis morfológico, el reconocimiento de entidades nombradas y el etiquetado.
- Detección de dominios: se comparan los textos extraídos con corpus general de referencia a definir, por ejemplo, el Corpus de Referencia del Español Actual (CREA) (RAE, 2010). Para extraer términos candidatos a pertenecer al dominio se emplean las métricas *C-Value* (Tsai, Lu, y Yen, 2012) y Similitud Coseno (Sidorov, Gelbukh, Gómez-Adorno, y Pinto, 2014).
- Generación de dominio: en esta fase se combinan los términos extraídos del dominio con las reglas generadas para el mismo. Todo ello es almacenado en una base de datos con la finalidad de crear repositorios de dominios de conocimiento.
- Validación dominios: en esta fase, mediante una interfaz web, se comprueba la relevancia de los textos extraídos por parte de un experto que verifica y valida las palabras y su relación con el dominio, incluyendo las relaciones de jerarquía.

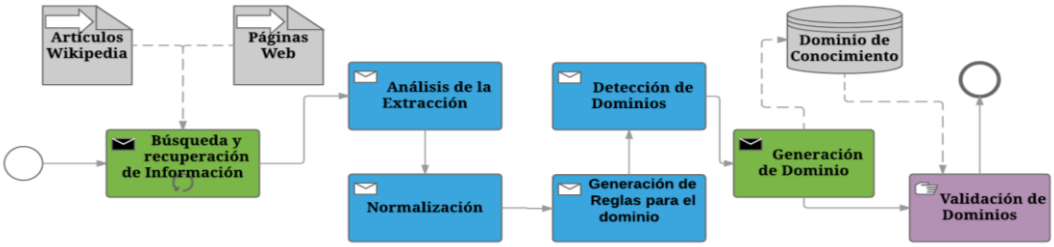

Figura 1: Proceso de construcción de dominios

4 Descripción del sistema

Basado en las técnicas anteriormente mencionadas y utilizando las mejores prácticas de ingeniería de software, el diseño del sistema se realiza en *Unified Modeling Language* (UML) (Uml, 2004); la metodología de desarrollo utilizada es *Agile Unified Process* (AUP) (Edeki, 2013) y el desarrollo de la aplicación se está implementando mediante un enfoque orientado a objetos en *Python 3.6* (Phillips, 2015), utilizando las siguientes librerías: *Natural Language Toolkit* (NLTK) (Bird, 2006), *Wikipedia*, *Google*, *html2text*, *Scrapy*, *bs4 y urllib*. La interfaz gráfica se implementará mediante metodologías ágiles (Ratcliffe y McNeill, 2011). Además, se utilizan las siguientes tecnologías: *HyperText Markup Language (HTML 5)* (Hickson y Hyatt, 2008), *JavaScript*, *jQuery*, entre otras. Por último, para facilitar la interoperabilidad y la integración con otras aplicaciones se diseña un servicio Web REST (Battle y Benson, 2008), mediante un enfoque orientado a servicio (Erl, 2005).

5 Avances

En la primera fase, se ha ejecutado la extracción de artículos en Wikipedia mediante el método de fronteras de domino propuesto por los autores (Arnold y Rahm, 2015) y mediante el método de rutas gráficas de las categorías de Wikipedia propuesto por los autores (Vivaldi y Rodríguez, 2001). Adicionalmente, se ha realizado la extracción de textos de páginas web utilizando el enfoque *Automatic data Record Mining – AutoRM* propuestos por (Shi et al., 2015). En este módulo se ha logrado obtener la identificación de aproximadamente un 90% de los artículos y páginas web correspondiente a un dominio en particular, utilizando librerías como: *Wikipedia API for Python* (WikipediaAPI, 2014), *Screen-scraping library* (Richardson, 2013), *Turn HTML into equivalent Markdown-structured text* (html2text, 2016) y *Python bindings to the Google search engine* (Google, 2016).

En la segunda fase, se realizaron pruebas de precisión y exhaustividad, utilizando la colección de artículos de Wikipedia identificados en la primera fase, lo cual obtuvo una precisión del 90% y una exhaustividad del 10%.

Para la tercera fase, se ha trabajado en la eliminación de textos en idiomas diferentes al español, normalización de textos utilizando *stopwords*, y eliminación de textos irrelevantes al contexto mediante la utilización de expresiones regulares. Los resultados previos en este módulo han sido la experimentación en un contexto en particular con 10 artículos en Wikipedia (de los cuales un 80% se han normalizado). Con respecto a las páginas Web identificadas se han presentado inconvenientes, debido a patrones de textos que no se habían contemplado, por ejemplo: URL, nombres de archivos, enlaces, texto en otros idiomas, entre otros. En esta fase se ha utilizado la librería NLTK y su corpus en español para realizar *Lematización*, *Stemming* y Análisis morfológico.

6 Resultados esperados

Al finalizar este proyecto se espera dar cumplimiento al objetivo propuesto y mejorar las capacidades investigativas en procesamiento de lenguaje natural, minería de textos y lingüística computacional en CAOBA. Además, se espera obtener un constructor automático de modelos de dominios y un servicio web, que se pueda utilizar en creación de nuevos corpus, clasificadores, análisis de sentimientos y análisis de personalidades.

Agradecimientos

Los desarrollos presentados en este proyecto se llevaron a cabo dentro de la construcción de capacidades de investigación del Centro de Excelencia y Apropiación en Big Data y Data Analytics (CAOBA), liderado por la Pontificia Universidad Javeriana, financiada por el Ministerio de Tecnologías de la Información y Telecomunicaciones de la República de Colombia (MinTIC) (Alianza CAOBA, 2017).

Bibliografía

Alianza CAOBA, 2017. Centro de Excelencia big data y Data Analytics Colombia, tic. (n.d.). Retrieved from http://alianzacaoba.co/

Arnold, P., y E. Rahm. 2015. Automatic extraction of semantic relations from wikipedia. *International Journal on Artificial Intelligence Tools, 24*(2), 1540010.

Battle, R., y E. Benson. 2008. Bridging the semantic web and web 2.0 with representational state transfer (REST). *Web Semantics: Science, Services and Agents on the World Wide Web, 6*(1), 61-69.

Bird, S. NLTK: The natural language toolkit. *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, 69-72.

Board, A. R. 2007. RSS 2.0 specification.

Edeki, C. 2013. Agile unified process. *International Journal of Computer Science, 1*(3)

Erl, T. 2005. *Service-oriented architecture: Concepts, technology, and design* Pearson Education India.

RAE, R. A. 2010. Corpus de referencia del español actual. *Accesible on Line at Http://Corpus.Rae.Es/Creanet.Html*,

Google, 1. 9. 3. 2016. *Python bindings to the google search engine*. Retrieved from https://pypi.python.org/pypi/google/1.9.3

Hickson, I., y D. Hyatt. 2008. No title. *Html 5: W3c Working Draft*,

html2text, 2. 9. 1. 2016. *Turn HTML into equivalent markdown-structured text.*. Retrieved from https://github.com/Alir3z4/html2text/

Leskovec, J., A. Rajaraman, y J. D. Ullman. 2014. *Mining of massive datasets* Cambridge University Press.

Nicol, G., L. Wood, M. Champion, y S. Byrne. 2004. Document object model (DOM) level 3 core specification. *W3C Working Draft, 13*, 1-146.

Phillips, D. 2015. *Python 3 object-oriented programming* Packt Publishing Ltd.

Powers, D. M. 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation.

Python, 3. 6. 2017. Python software foundation. Retrieved from https://www.python.org/

Ratcliffe, L., y M. McNeill. 2011. *Agile experience design: A digital designer's guide to agile, lean, and continuous* New Riders.

Richardson, L. 2013. Beautiful soup. *Crummy: The Site*,

Schreiber, G. 2000. *Knowledge engineering and management: The CommonKADS methodology* MIT press.

Shi, S., C. Liu, Y. Shen, C. Yuan, y Y. Huang. 2015. AutoRM: An effective approach for automatic web data record mining. *Knowledge-Based Systems, 89*, 314-331. doi: 2048/10.1016/j.knosys.2015.07.012

Sidorov, G., A. Gelbukh, H. Gómez-Adorno, y D. Pinto. 2014. Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación Y Sistemas, 18*(3), 491-504.

Tsai, C., Y. Lu, y D. C. Yen. 2012. Determinants of intangible assets value: The data mining approach. *Knowledge-Based Systems, 31*, 67-77.

Uml, O. 2004. 2.0 superstructure specification. *OMG, Needham, 21-187*

Vaishnavi, V., y W. Kuechler. 2004. Design science in information systems research. *MIS Q*, 28, 75-105.

Villayandre Llamazares, M. 2008. Lingüística con corpus (I). *Estudios Humanísticos. Filología*, 30, 329-349.

W3C, 2. 2017. World wide web consortium. Retrieved from https://www.w3.org/

WikipediaAPI, 1. 4. 2014. *Wikipedia API for python.* Retrieved from https://github.com/richardasaurus/wiki-api

Zhu, M. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, 2*, 1-30.