Procesamiento del Lenguaje Natural, Revista nº 58, marzo de 2017, pp. 163-166 recibido 01-12-2016 revisado 11-01-2017 aceptado 06-02-2017

Detección de reutilización de código fuente monolingüe y translingüe

Crosslingual and monolingual source code re-use detection

Enrique Flores Sàez
Universitat Politècnica de València
Camí de Vera, s/n, Valencia
enflosae@gmail.com

Resumen: Tesis doctoral en Informática realizada por Enrique Flores Sàez en la Universitat Politècnica de València bajo la dirección de la Dra. Lidia Moreno Boronat y del Dr. Paolo Rosso. El acto de defensa tuvo lugar el lunes 30 de Mayo de 2016 ante el tribunal formado por los doctores Fidel Cacheda (Universidad de A Coruña), Juan Manuel Torres (Universidad de Avignon Murcia) y Greg Kondrak (Universidad de Alberta). Obtuvo mención internacional y calificación de Sobresaliente.
Palabras clave: Detección de reutilización en códigos fuente, detección de reutilización monolingüe, detección de reutilización translingüe, detección de plagio

Abstract: PhD Thesis in Computer Science written by Enrique Flores Sàez at the Universitat Politècnica de València under the supervision of PhD. Lidia Moreno Boronat and PhD. Paolo Rosso. The author was examined on 30th May 2016 by a committee formed by the PhD Fidel Cacheda (University of A Coruña), Juan Manuel Torres (University of Avignon) and Greg Kondrak (University of Alberta). The thesis obtained the grade of Excellent and received the international mention.
Keywords: Source code re-use detection, monolingual re-use detection, crosslingual re-use detection, plagiarism detection

1 Introducción

La detección automática de reutilización en códigos fuente consiste en determinar si un (fragmento de) código ha sido creado considerando otro como fuente. El plagio y las ramificaciones en proyectos software son dos ejemplos de tipos de reutilización en códigos fuente. Con la llegada de la Web y los medios electrónicos ha crecido enormemente la facilidad de acceso a códigos fuente para ser leídos, copiados o modificados. Esto supone una gran tentación para programadores que, con propósitos de reducir costes (temporales o económicos), deciden utilizar códigos fuente previamente depurados y probados. Este fenómeno ha causado que expertos en lenguajes de programación estudien el problema.

La gran cantidad de recursos disponibles en la Web hace imposible un análisis manual de códigos fuente sospechosos de haber sido reutilizados. Por ello, existe una necesidad urgente de desarrollar herramientas automáticas capaces de detectar con precisión los casos de reutilización. Basándose en técnicas del procesamiento del lenguaje natural y recuperación de información, las herramientas de detección automáticas de reutilización son capaces de realizar multitud de comparaciones de códigos fuente de forma eficiente.

Uno de los principales objetivos de esta tesis es abordar el problema de la reutilización de códigos fuente escritos en un lenguaje de programación como si se tratase de un texto escrito en un lenguaje natural. Por este motivo, en este trabajo de investigación se han estudiado los principales modelos aplicados a reutilización de textos en lenguaje natural, se han implementado y adaptado para aplicarse sobre códigos fuente. En esta tesis se describen los modelos implementados para la detección de reutilización en código fuente. Los modelos presentados en esta tesis siguen el siguiente esquema: Inicialmente, una colección de códigos fuente es tratada mediante un preproceso concreto. Tras este preproceso se obtiene un conjunto de características que representan a cada código fuente, y finalmente, se aplica una medida de similitud obteniéndose como resultado un listado de pares de códigos fuente junto al valor de similitud entre cada par.

ISSN 1135-5948

En esta tesis proponemos un conjunto de modelos que pueden aplicarse indistintamente a nivel monolingüe o translingüe. Es decir, se pueden comparar dos códigos que están escritos en el mismo, o en distinto, lenguaje de programación. Inicialmente, hemos evaluado los modelos tanto en un escenario académico como en un escenario de detección a gran escala. Finalmente, nuestras mejores propuestas se han comparado con otras propuestas del estado de la cuestión dentro de un mismo marco de evaluación.

Estas pruebas de nuestros modelos se han realizado mediante millones de comparaciones tanto a nivel monolingüe como translingüe empleando diversas técnicas que fueron efectivas al aplicarlas sobre textos escritos en lenguaje natural. Con los recursos utilizados en esta tesis hemos configurado dos escenarios (monolingües y translingües) de evaluación que son un referente para que actuales y futuros trabajos de investigación puedan ajustar y comparar sus propuestas.

2 Estructura de la tesis

- Estado de la cuestión: En este capítulo se resume como se ha abordado la detección de reutilización en el área de la investigación. Primeramente se repasa la detección de reutilización en textos y seguidamente sobre códigos fuente tanto a nivel monolingüe como translingüe.
- Recursos: Este capítulo describe los corpus recopilados, procesados y etiquetados manualmente en este trabajo de tesis.
- Modelos propuestos: Este capítulo propone diferentes modelos para resolver la problemática de la detección automática de reutilización en códigos fuente. Estos modelos están orientados a considerar los códigos fuente como si de un texto se tratara.
- Experimentación: Este capítulo está dividido en dos partes. La primera parte describe la experimentación realizada a nivel monolingüe, para detectar la reutilización entre códigos fuente escritos en un mismo lenguaje de programación. La segunda parte representa la contribución más novedosa de este trabajo de investigación, la detección de reutilización translingüe.
- Evaluación en la competición (CL-) SOCO: En el marco de esta tesis se han organizado dos competiciones de detección de reutilización en código fuente, una monolingüe y la otra translingüe. En ambos casos se estudian los resultados conseguidos por los participantes respeto a los resultados de mayor rendimiento desarrollados en esta tesis.
- Conclusiones y trabajos futuros: Este capítulo resume las principales aportaciones de este trabajo de investigación. También se enuncian las posibles líneas a seguir en trabajos futuros.

3 Principales contribuciones

La principal dificultad para la investigación en detección automática de reutilización en códigos fuente es la escasez de colecciones de códigos fuente que estén etiquetadas con casos de reutilización. De este modo, hemos propuesto y adaptado un conjunto de modelos que mostraron un buen rendimiento en la detección de reutilización y similitud en textos. Los resultados obtenidos por estos modelos han mostrado ser válidos y más eficaces que las demás aproximaciones del estado de la cuestión.

Durante esta tesis se han recopilado una serie de recursos que han ayudado a la investigación de causas y tipos de reutilización. Un ejemplo de ello es el corpus A&T++ con casos de reutilización monolingüe en C y Java, y que a posteriori hemos enriquecido con casos de reutilización translingüe. Hemos etiquetado manualmente los corpus monolingües. En el corpus A&T++, en primer lugar, hemos comparado con la herramienta JPlag siendo los resultados de ambos modelos con desempeño superior a ésta. Finalmente, comparamos todos los modelos propuestos. También añadimos a la comparación un ensamble de modelos utilizando clasificadores conocidos. El ensamble de los modelos, muestra una ligera mejoría en rendimiento. Por otra parte, se han identificado las modificaciones más frecuentes para evadir su detección. Ha resultado importante conocer y diferenciar los mecanismos/modificaciones que se realizan tanto a nivel monolingüe como a nivel translingüe (Flores et al., 2011; Flores et al., 2014; Flores, Moreno, y Rosso, 2016; Flores et al., 2012; Flores et al., 2011).

Se han analizado los cd́igos fuente en busca de casos de reutilización en la primera fase

de la competición Google Code Jam. El objetivo es descubrir si es un entorno propicio para la reutilización, si la reutilización es muy abundante y de que forma se realiza (Flores et al., 2015). El corpus Google Code Jam supone el mayor corpus disponible de códigos fuente con casos de reutilización conocidos. Esto supone una mayor dificultad para descartar los casos de reutilización. Solo seleccionamos los tres lenguajes más populares, C/C++, Java y Python, por lo que existe posibilidad de ampliación. Se han etiquetado manualmente 360 pares de códigos fuente muy similares.

En la experimentación realizada con la colección Google Code Jam hemos realizado más de 34 millones de comparaciones. Comprobamos que es posible aplicar y detectar casos de reutilización en un corpus a gran escala con nuestra propuesta. Además, comparamos con la herramienta JPlag donde observamos que, en líneas generales, se observa un desempeño similar en tareas de mayor cantidad de líneas de código fuente, mientras que en los códigos cortos detectamos más casos de reutilización.

Por otra parte, a nivel translingüe hemos recorrido una línea experimental similar a la monolingüe donde primeramente comprobamos que es importante considerar el código fuente completo para obtener el mejor rendimiento. Se han realizado estudios sobre otro tipo de escenario como plantea el repositorio de códigos fuente Rosettacode.org. Sobre este repositorio hemos estudiado el desempeño de los modelos propuestos recuperando códigos fuente relacionados y también traducidos (Flores et al., 2014; Flores et al., 2015).

Rosettacode.org consiste en un "banco" de códigos fuente con implementaciones de algoritmos conocidos. Hemos recopilado los códigos fuente de C, Java y Python. Tras un etiquetado manual se encontraron casos de reutilización translingüe en este corpus que además hemos enriquecido mediante herramientas de traducción automática.

La experimentación realizada con el corpus Rosettacode.org de gran tamaño comparando los modelos propuestos en dos escenarios de recuperación de información supuso más de cinco millones de comparaciones de códigos fuente por cada modelo de un total de nueve modelos. En general, los modelos muestran un mejor rendimiento en la recuperación de códigos fuente paralelos que en los comparables. Otro experimento que hemos realizado es simular un escenario de reutilización translingüe. Para ello consideramos los pares de códigos fuente paralelos como casos de reutilización y los comparables como no reutilizados. Comparando los modelos en este escenario de reutilización simulado, los modelos basados en características léxicas muestran mejores resultados. Este comportamiento sigue el comportamiento observado en las comparaciones a nivel monolingüe.

Como trabajo paralelo a esta investigación, y con la intención de crear un marco de evaluación común para otros trabajos de investigación, hemos organizado dos competiciones de detección de reutilización automática en códigos fuente. En estas competiciones se han ofrecido tanto recursos para entrenar como para evaluar a disposición de los participantes para el desarrollo y mejora de sus sistemas propuestos. Estos marcos de evaluación han sido propuestos dentro del laboratorio PAN para la detección de reutilización en código fuente (SOCO) y para la detección de reutilización translingüe en código fuente (CL-SOCO) (Flores et al., 2014; Flores et al., 2015; Flores et al., 2015).

Finalmente, hemos realizado una comparación de los modelos de mejor rendimiento propuestos en esta tesis con los sistemas propuestos en las competiciones aprovechando los marcos de evaluación que se facilitan en SOCO y CL-SOCO. Hemos entrenado los modelos utilizando los corpus de entrenamiento de la competición mediante validación cruzada. Comparando los resultados de SOCO con los que se han obtenido con los dos modelos propuestos, hemos podido apreciar que ambos modelos obtienen mejores resultados en términos de la medida F_1. Del mismo modo, en la comparativa translingüe con los sistemas presentados en CL-SOCO, el modelo SoCo-LSA obtiene los mejores resultados siendo estos significativamente mejores, mientras que el modelo SoCo-NG obtiene resultados similares a los mejores de la competición.

4 Conclusiones y trabajos futuros

El objetivo principal de esta tesis ha sido desarrollar modelos eficaces para la detección automática de reutilización en códigos fuente. Para ello, hemos tenido en cuenta distintos escenarios donde es posible que suceda la reutilización de códigos fuente. La re-

utilización/colaboración entre compañeros en el ámbito académico es tan importante como la reutilización en recursos externos como Webs, respositorios, etc. Se han realizado experimentos tanto a nivel monolingüe como translingüe para comprobar la validez de los modelos propuestos en situaciones distintas. Así pues, el trabajo de investigación se estructuró de la siguiente manera: *a)* recopilación de colecciones que contienen casos de reutilización monolingüe como translingüe; *b)* desarrollo y adaptación de modelos de detección de similitud en textos para códigos fuente; *c)* modificación de estos modelos para escenarios translingües; *d)* Estudio de las modificaciones para evitar la detección; *e)* análisis de la detección tanto a nivel académico como a gran escala; *f)* comparación del rendimiento de los modelos propuestos con distintos ensambles de estos modelos; *g)* comparación de los modelos en escenarios de recuperación translingüe; *h)* simulación de un escenario de reutilización translingüe a gran escala; *i)* contribución a la creación de marcos de evaluación para la detección de reutilización en códigos fuente; y *j)* comparación de los modelos desarrollados con propuestas del estado de la cuestión tanto a nivel monolingüe como translingüe.

Identificamos como trabajos futuros las siguientes líneas de investigación: Identificación del origen de la reutilización; detección intrínseca de reutilización; y detección de fragmentos reutilizados.

Bibliografía

Flores, E., A. Barrón-Cedeño, P. Rosso, y L. Moreno. 2011. Detecting Source Code Reuse across Programming Languages. *Póster en Conferencia Sociedad Española para el Procesamiento del Lenguaje Natural.*

Flores, E., P. Rosso, E. Villatoro-Tello, L. Moreno, R. Alcover, y V. Chirivella. 2015. PAN@FIRE: Overview of CL-SOCO track on the Detection of Cross-Language SOurce COde Re-use. En *7th International Workshop of the Forum for Information Retrieval Evaluation*, Gandhinagar, India.

Flores, E., A. Barrón-Cedeño, L. Moreno, y P. Rosso. 2014. Cross-language source code re-use detection. En *3a Conferencia Española de Recuperación de Información*, páginas 145–156, A Coruña, España.

Flores, E., A. Barrón-Cedeño, L. Moreno, y P. Rosso. 2015. Uncovering source code reuse in large-scale academic environments. *Computer Applications in Engineering Education*, 23(3):383–390.

Flores, E., A. Barrón-Cedeño, P. Rosso, y L. Moreno. 2011. Towards the detection of cross-language source code reuse. En *Natural Language Processing and Information Systems*, volumen 6716 de *Lecture Notes in Computer Science*. Springer, páginas 250–253.

Flores, E., A. Barrón-Cedeño, P. Rosso, y L. Moreno. 2012. DeSoCoRe: Detecting Source Code Re-use across programming languages. En *Conference of the North American Chapter of the ACM: HLT*, páginas 1–4, Montreal, Canadá. Association for Computational Linguistics.

Flores, E., M. Ibarra-Romero, L. Moreno, G. Sidorov, y P. Rosso. 2014. Modelos de recuperación de información basados en n-gramas aplicados a la reutilización de código fuente. En *3a Conferencia Española de Recuperación de Información*, páginas 185–188, A Coruña, España.

Flores, E., L. Moreno, y P. Rosso. 2016. Detecting source code re-use with ensemble models. En *4a Conferencia Española en Recuperación de Información*, Granada, España.

Flores, E., A. B.-C. no, L. Moreno, y P. Rosso. 2015. Cross-Language Source Code Re-Use Detection Using Latent Semantic Analysis. *Journal of Universal Computer Science*, 21(13):1708–1725.

Flores, E., P. Rosso, L. Moreno, y E. Villatoro-Tello. 2014. PAN@FIRE: Overview of SOCO track on the detection of SOurce COde re-use. En *6th Forum for Information Retrieval Evaluation.*

Flores, E., P. Rosso, L. Moreno, y E. Villatoro-Tello. 2015. On the detection of source code re-use. En *Forum for Information Retrieval Evaluation*, páginas 21–30.