

АГЕНТИ ЗА ПРОВЕРКА ЗА ПРЕПИСВАНЕ*

Никола Вълчанов¹, Асен Рахнев², Николай Павлов³, Олга Рахнева⁴

^{1, 2, 3} Факултет по математика и информатика, Пловдивски университет, България

⁴ Университет по хранителни технологии, Пловдив, България

¹ nvalchanov@uni-plovdiv.bg

AGENTS FOR PREVENTING COPYING

Nikola Valchanov¹, Asen Rahnev², Nikolay Pavlov³, Olga Rahneva⁴

^{1, 2, 3} Faculty of Mathematics and Informatics, Plovdiv University, Bulgaria

⁴ University of Food Technologies, Plovdiv, Bulgaria

¹ nvalchanov@uni-plovdiv.bg

Abstract. This work describes the development of software agents that look for similarities in papers submitted by students within the dynamic website development class. These similarities are identified by specific code attributes that guarantee with high accuracy that the paper contains copied content. The agents are integrated in the e-testing process within the “Tourism” masters program in Plovdiv university “Paisii Hilenradski”, branch Smolyan. They are used for verification of papers on “dynamic website development with specialized tools” (<http://www.tur-mag.kodar.net>).

Key words: software agents, copy prevention, system integration, design templates

Информационните технологии играят важна роля в съвременното обучение. Те се използват при подготовката на лекционни материали, в обучителния процес, при провеждане на изпити и др. Чрез тях обучаващите автоматизират времеемки дейности като подготовка на изпитни варианти, оценяване и много други процеси, свързани с обучението. По този начин с помощта на информационните технологии, обучаващите пестят време и усилия и имат възможност да се концентрират върху по-важни въпроси.

Една от възможностите, които информационните технологии предоставят, е автоматизирано сравняване на писмени работи на обучаеми. Чрез добре познати алгоритми за директно сравняване на низове или търсене на разстояния между низове може да бъде установено дали между две писмени работи има сходства, които будят подозрение. Така информационните технологии могат да бъдат прилагани и за ограничаване на преписването.

Като част от курса „Компютърни информационни системи в туризма” в рамките на магистърска програма „Мениджмънт в туризма” във филиала на Пловдивски университет „Паисий Хилендарски” в гр. Смолян, е разработен специализиран сайт (Рахнев, 2010), чрез който студентите се обучават в изграждане на динамични уеб-сайтове със специализирани инструменти. Всеки студент получава профил и набор от

* Това изследване е частично финансирано от проект НИ11-ФМИ004 към НПД на Пловдивски университет „Паисий Хилендарски”

инструменти за изграждане на динамичен уеб-сайт. В периода на обучение, студентите трябва да изготвят и предадат под формата на проект динамичен уеб-сайт, изграден с предоставените им инструменти.

Настоящата работа проследява създаването на софтуерни агенти, които търсят сходства в съдържанието на предадените проекти. Обърнато е внимание на техническите специфики, свързани с интеграцията на агентите в специализирания сайт с употребата им за проверка за преписване.

Актуално състояние

Реализации на системи за идентифициране на плагиатство се използват в практиката от години. Основно тези системи се делят на две групи – проверяващи текст на естествен език и сравняващи програмен код (Ангелова, 2012).

Системите, проверяващи текст на естествен език, предоставят възможности за сравняване на набор от документи и идентифициране на локално и глобално дублиране на текст. Често този тип системи предоставят възможност за добавяне на шаблон, като той бива игнориран при проверката на текстовете. Повечето представители на този тип системи предоставят два режима на проверка – в локални ресурси и в търсещи машини в Интернет. При проверка в локални ресурси потребителя предоставя набор от контролни документи. Така системата търси съвпадения между текста и някой от контролните документи. При втория режим на проверка вместо контролни документи, системите използват търсещи машини в Интернет (Google, Bing и др.). Представители на този тип системи са:

- Chimpsky – поддържа форматите PDF, DOC, DOCX, WPD, RTF и ODT. Предоставя възможност за качване на шаблон и работа с търсещи машини в Интернет (Palmer);
- CitePlag – специализира в търсене, базирано на цитати в академични документи (Meuschke, 2012);
- PlagScan – поддържа работа с търсещи машини в Интернет. Предоставя приложен програмен интерфейс за интеграция във външни системи (PlagScan);
- Plagiarisma.Net – поддържа работа с търсещи машини в Интернет (Plagiarisma.Net);
- Turnitin – предоставя три продукта – проверка за оригиналност (OriginalityCheck), оценяване на работата (GradeMark) и такъв за оценка от други обучаеми (PeerMark) (Turnitin).

При системите сравняващи програмен код се взимат предвид специфики на използвания програмен език. Те извличат структурата на програмата и сравняват отделни нейни елементи. Представители на този тип системи са:

- Sherlock – успешно може да бъде прилагана за търсене в стандартен текст и в код. Системата проверява всички файлове в дадена папка на принципа всеки с всеки (Sherlok);
- PMD – може да бъде прилагана за търсене в Java, JSP, C, C++, Fortan и PHP код (PMD);
- MOSS – може да бъде прилагана за търсене в C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, ML, Haskell, Lisp, Scheme, Pascal, Modula2, Ada, Perl, TCL, Matlab, VHDL, Verilog, Spice, MIPS assembly, a8086 assembly,

a8086 assembly, MIPS assembly и HCL2 код. Системата поддържа задаване на шаблони, които да бъдат игнорирани при проверката (MOSS);

- JPlag – може да бъде прилагана за търсене в Java, C#, C, C++, Scheme и естествени езици (JPlag).

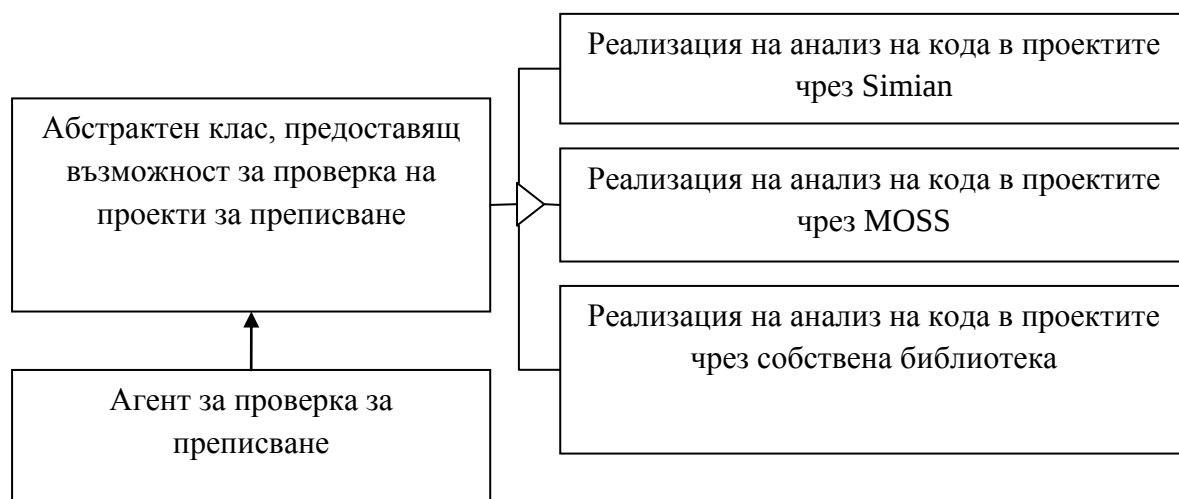
Подобни на решенията за идентифициране на плагиатство в програмен код са инструментите за търсене на дублиращ се код в софтуерни системи. При тях се търсят повтарящи се секции, като отново се взимат предвид синтаксиса и специфичната конструкция на съответния език за програмиране.

Системата Simian (Simian) е представител на този тип приложения. Тя може да бъде използвана за идентифициране на дублиран код във системи, разработени на Java, C#, C, C++, COBOL, Ruby, JSP, ASP, HTML, XML, Visual Basic и Groovy. Инструментът работи добре и при сравняване на обикновен текст.

Реализация на агенти за проверка за преписване

Направеният анализ на съвременни решения за идентифициране на плагиатство открии инструмента Simian като най-подходящ за настоящите цели на агентите. Съдържанието на проектите, които студентите предават се съхранява под формата на HTML код. Именно поради това анализиращият инструмент е избран така, че да дава възможно най-добра производителност при работа с именно този дескриптивен език.

Агентът за търсене на дублиране на код е разработен чрез платформата .NET и се изпълнява посредством JADE Sharp Connector в JADE контейнер (JadeSharp). Той капсулира в себе си вътрешната логика по сравняване на материали като за конкретната реализация се използва Simian. В конфигурацията на агента се съхранява лимита допустими дублирани редове между съдържанието на две страници от два различни проекта. По този начин агентът може да достигне до заключение дали два документа имат приемлива степен на сходство или въпросът трябва да бъде отнесен до контролен орган, който да извърши щателна проверка.



Фигура 1. Схема на структурата на агент за проверка за преписване

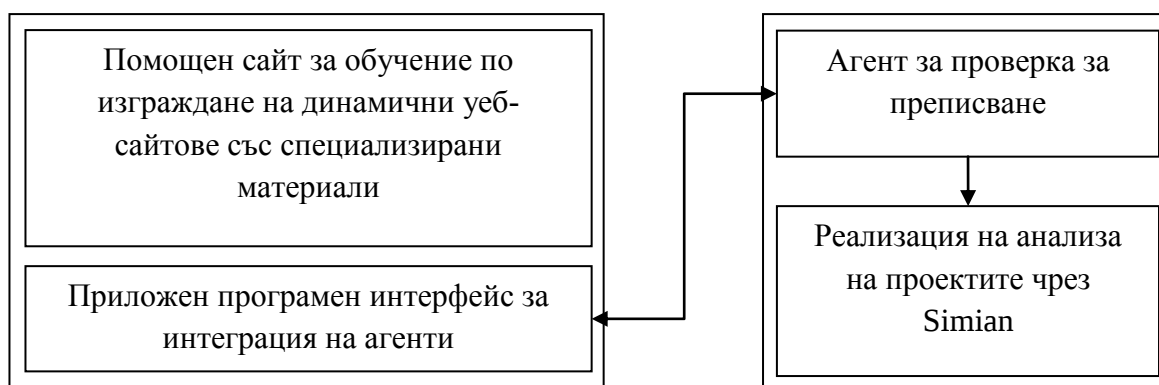
Структурата на агента е реализирана чрез шаблона за дизайн „доставчик“. По този начин конкретната реализация на библиотеката за проверка за преписване се скрива зад абстрактен клас, предоставящ функционалността на агента. При заявка за сравняване на писмени работи агентът прочита от конфигурацията си данни за библиотеката, с която трябва да реализира сравнението. След като библиотеката е

известна агентът я зарежда, създава неин представител, реализиращ абстрактния клас и го използва при проверката на проекти за преписване.

Интеграция на реализирания агент в система за електронно изпитване

В рамките на системата е създаден набор от уеб услуги, които предоставят достъп до функционалността, свързана с оценяването на проектите. По този начин е предоставен приложен програмен интерфейс към агентите, подпомагащи процеса на оценяване. Чрез него агентите за проверка за преписване получават достъп до системната функционалност и могат да се включат активно в работата на системата.

Агентът за проверка за преписване следи проектите, които постъпват в системата като всеки нов се проверява за дублиран код с предадените до момента. Сравняването на две студентски работи се извършва като всяка страница на единия проект се сравнява с всяка една от страниците на другия. При намиране на страници, надхвърлящи лимита допустими дублирани редове агентът изпраща електронно писмо до проверяващият орган и създава запис в специализирана таблица в базата от данни. По този начин се цели известяване на оценяващия преподавател за установена висока вероятност за плагиатстван материал.



Фигура 2. Схема на интеграцията на агент за проверка за преписване в помощния сайт

При вход в системата, оценяващият преподавател вижда рисковите проекти и обръща съответното внимание на посочените от агента секции в документите.

Заклучение

В света на информационните технологии все по-често могат да бъдат намерени нови разработки по проблеми, свързани с идентифицирането на плагиатство. Актуалното качество на тези приложения дава добри резултати при прилагането им в практиката. Именно затова те биват използвани в образователния процес, биват интегрирани в системи за електронно обучение, а в следствие от тях разработените алгоритми се прилагат и за изграждане на инструменти за търсене на дублиране на код.

Не винаги е възможна интеграцията на система за идентифициране на плагиатство. В настоящата разработка е демонстриран именно такъв случай. Като алтернативна мярка е използван инструмент за идентифициране на дублиране на код, който обслужва по-адекватно нуждите на агента за проверка за преписване. Разработените агенти се прилагат успешно при електронно изпитване на студенти от магистърската специалност „Туризм“ към ПУ „Паисий Хилендарски“, филиал Смолян, при разработка на динамични уеб-сайтове със специализирани инструменти (<http://www.tur-mag.kodar.net>).

Литература

Рахнев, А., Стоева, М., Арнаудова, В., Радев, Р. Учебен динамичен уеб сайт за неспециалисти. // *Доклади на юбилейна международна конференция „Синергетика и рефлексия в обучението по математика SREM 2010”*, 10-12 септември, 2010, Бачиново, с. 453-460.

Ангелова, Н. Софтуерът за плагиатство в обучението. // *V-та Национална конференция „Образованието в информационното общество”*, 2012, с. 283-291.

Palmer, M. <http://chimpsky.uwaterloo.ca>. // *University of Waterloo*.

Meuschke, N., Gipp, B., Breitingner C. CitePlag: A Citation-based Plagiarism Detection System Prototype. // *5th International Plagiarism Conference '12*, July 16–18, 2012, Newcastle upon Tyne, UK.

PlagScan. <<http://www.plagscan.com/>>

Plagiarisma.NET. <<http://plagiarisma.net/>>

Turnitin. <<http://turnitin.com/>>

Sherlock. <<http://sydney.edu.au/engineering/it/~scilect/sherlock/>>

PDM. <<http://pmd.sourceforge.net/>>

MOSS. <<http://theory.stanford.edu/~aiken/moss/>>

JPlag. <<https://www.ipd.kit.edu/jplag/>>

Simian. <<http://www.harukizaemon.com/simian/>>

JadeSharp. http://jade.tilab.com/doc/tutorials/JADE_SHARP_Guide.pdf

