Textual and Source Code Plagiarism in Academic Environment: a Serbian perspective

Invited presentation

Marko Mišić (marko.misic@etf.bg.ac.rs)
Assistant Professor

University of Belgrade
School of Electrical Engineering
Department of Computer Engineering and Informatics
Serbia

Plagiarism detection conference 2022

Background

- Assistant Professor at University of Belgrade (UB)
  - School of Electrical Engineering (SEE),
    Department of Computer Engineering and Informatics
- Teaching several freshman-year massively-enrolled courses
  - Programming, algorithms and data structures with 200-700 students
  - Tackling plagiarism detection problems for more than 10 years
- PhD and research in plagiarism detection
  - Improving source code plagiarism detection
  - Developing software tools and methodology
  - Numerous papers in journals and conferences
- Chairman and member of the disciplinary committee at UB-SEE (4 years)
  - Cases of student academic dishonesty, disciplinary hearings
  - 30-50 cases annually

Content

- Introduction
- Context and motivation
- Textual plagiarism detection
- Serbian language linguistic features
- Text comparison considerations
- Document repositories in Serbian
- Current efforts and status in Serbia
- Source code plagiarism detection
- Similarity detection systems
- Plagiarism investigation
- Conclusion

Introduction

- Academic integrity has increasingly become an important topic in the academic community in recent years
- Several notable cases of plagiarism among highly-positioned individuals in Europe
  - Karl-Theodor zu Guttenberg (Minister of Defence of Germany, PhD thesis, 2011)
  - Pal Schmitt (Hungarian President, PhD thesis, 2012)
  - Victor Ponta (Romanian Prime Minister, PhD thesis, 2012)
  - Ursula von der Leyen (Minister of Defence of Germany, PhD thesis, alleged, 2016)
  - Xavier Bettel (Luxembourg Prime Minister, MSc thesis, alleged, 2021)
- Serbia is not an exception to that problem
  - Siniša Mali (Mayor of Belgrade, Minister of Finance, PhD thesis, alleged, 2013-2021)
- A much wider problem revealed at the student level

Context and motivation (1)

- Plagiarism definition(s):
  - "Presenting someone else's ideas or work, in whole or in part, without proper author or source attribution/crediting"
  - "The act of illegally appropriating someone else's spiritual creations and presenting them as one's own"
  - Serious academic misconduct and breach of academic honesty!
- Various acts, regulations, and codes of honour to regulate the matter
  - Both for professors/researchers, and students
  - Princeton University, USA - Constitution of the Honor System
  - MIT, USA - Academic Integrity Handbook for students
  - University of Belgrade - Rulebook on disciplinary responsibility of students

Context and motivation (2)

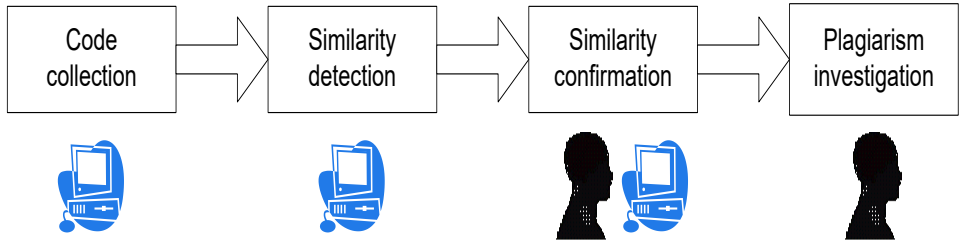
- Students are not well-informed about plagiarism
  - Definition, allowed practices, honour codes
- Different views regarding following practices:
  - (Un)allowed collaboration patterns and teamwork
  - Text, image and source code reuse
  - Autoplagiarism
- Surveys in the open literatures state that:
  - More than 30% of students admitted plagiarism once during their studies
  - More than 60% admitted that they have given their work to the others
  - 5-10% of students plagiarize their solutions

Context and motivation (3)

- Anonymous survey at UB-SEE revealed that students have generally softer stance towards plagiarism:
  - One out of ten students considers plagiarism tolerable practice
  - More than 40% of students sent their work to the others
  - 80% of students think that it is allowed to send the work to other party and that the sole responsibility is on the side that uses someone else's work
  - 7% admits that they submitted someone else's work as their own
- Numerous students are aware of dishonest practices:
  - Purchasing for papers, thesis, projects of homework assignments
  - Passing exams using electronic devices
    - Cell phones, smart watches, miniature cameras or ear plugs
  - Passing exams instead of a someone else

Context and motivation (4)

- Textual and source code plagiarism represent the most frequent cases of academic misconduct
  - Thesis work, projects, homework solutions, various reports
- Obvious need to check submitted documents for plagiarism
  - Software tools are used for similarity detection to prevent such inappropriate behaviour
  - TurnitIn/iThenticate, Antiplagiat/Advacheck, etc., for text
  - JPlag (Karlsruhe University), Moss (Stanford University), etc., for source code
  - Numerous non-profit, academic efforts
- Document similarity ≠ document plagiarism!
  - Positive and negative causes of similarity
  - Thorough manual inspection of suspicious cases



Textual plagiarism detection

- Numerous methods to hide plagiarism
  - Lack of citations or improper citations
  - Simple or mosaic paraphrasing, rewording, word reordering, and similar
  - Metaphors
  - Foreign language translations
- Documents for comparison
  - Institutional/professor local repositories
  - Index databases and repositories, internet documents
- Software tools are mostly adapted to English language
  - Do not take into account linguistic features of other languages
  - Typically yielding lower similarity scores for other languages

Serbian language linguistic features (1)

- Serbian language is one of the standardized varieties of Bosnian-Croatian-Serbian common south Slavic language
  - Spoken in Serbia, Croatia, Bosnia and Herzegovina, and Montenegro
  - Differences in used scripts (alphabets), some dialectic details, and accentuation
- Official script in Serbia is Cyrillic, but Latin script is also widely used
  - Serbian is practically the only European standard language whose speakers are fully functionally digraphic
  - The standard recognizes the usage of both scripts
- The language orthography is built around phonemic principle "one letter for one voice"
  - However, there are unofficial orthographies in internet documents

Serbian language linguistic features (2)

- Bosnian-Croatian-Serbian has two standardized word pronunciations and spellings: Ekavian, and Ijekavian
  - Ekavian is widely used in Serbia, while Ijekavian is officially used in Croatia, Bosnia and Herzegovina, and Montenegro
  - The differences are based on the iotification of old Slavic letter yat (ѣ in Cyrillic or ě in Latin) in some words

Cyrillic	Latin	Alternative Latin (unofficial usage)	Cyrillic	Latin	Alternative Latin (unofficial usage)
А а	A a		Н н	N n	
Б б	B b		Њ њ	Nj nj	
В в	V v		О о	O o	
Г г	G g		П п	P p	
Д д	D d		Р р	R r	
Ђ ђ	Đ đ	Dj dj	С с	S s	
Е е	E e		Т т	T t	
Ж ж	Ž ž	Z z	Ћ ћ	Ć ć	C c
З з	Z z		У у	U u	
И и	I i		Ф ф	F f	
Ј ј	J j		Х х	H h	
К к	K k		Ц ц	C c	
Л л	L l		Ч ч	Č č	C c
Љ љ	Lj lj		Џ џ	Dž dž	Dz dz
М м	M m		Ш ш	Š š	S s

Text comparison considerations

- The easiest way to hide plagiarism is to change the used script
- A need for transliteration in plagiarism detection is obvious
  - Comparing documents in one canonical form
- To improve plagiarism detection results, several notes should be taken into consideration:
  - Texts from both Cyrillic and Latin corpora should be considered
  - Latin script should be used for comparison, as text can be borrowed from documents written in Croatian, Bosnian, and partially in Montenegrin
  - Alternative Latin orthography should be considered for internet sources

Document repositories in Serbian

- There are several repositories of scientific documents written in Serbian that are open-access:
  - Serbian national repository of PhD thesis from 2017 onwards https://nardus.mpn.gov.rs/
  - University of Belgrade, university library "Svetozar Marković" repositories list and early access to PhD thesis https://uvidok.rcub.bg.ac.rs/
  - Singidunum University https://singipedia.singidunum.ac.rs/diplomski-radovi
  - University of Novi Sad, Faculty of Philosophy http://remaster.ff.uns.ac.rs/
- Most institutions do not have open access to their BSc and MSc thesis repositories
  - Concerns related to plagiarism and academic dishonesty

Current efforts and status in Serbia

- Archiving and checking for plagiarism of PhD thesis is mandatory
  - Law on higher education of Republic of Serbia (2014 and 2018)
  - NaRDuS system for archiving of thesis and their reports
  - TurnitIn/iThenticate tools used for checking
  - Procedures and regulations were improved on several occasions
- Serious problems with financing of plagiarism checking
  - Delaying thesis defenses of candidates
  - Problems with public procurements and lack of funds
- Sporadic efforts at other (social) schools/faculties:
  - Seminary work, BSc and MSc thesis checking is mandatory at the School of Economy with Ephorus plagiarism checker
    - Allowed similarity of 25% for seminary and BSc, 10% for MSc, and 5% for PhD works
  - School of Political Sciences has its own regulations

Source code plagiarism detection (1)

- Computing education is a demanding activity that involves practical training
  - Programming assignments, & projects, laboratory work
  - Important for gaining programming competences
  - Significant problem at IT schools, but also in industry
- Source code plagiarism definition
  - "Source code plagiarism is any intentional or unintentional source code submission and reuse which fails to adequately acknowledge the other's work" (Cosma, Joy, 2008.)
- Context
  - Academic environment – plagiarism detection
    - Comparison of numerous small-scale software solutions
    - Active attempts to hide plagiarism
  - Industry – software clone detection
    - Several larger software solutions
    - Intellectual property & patent rights

Source code plagiarism detection (2)

- Source code plagiarism detection differs from textual plagiarism detection in several aspects:
  - Source code has a clear structure
  - Programming languages are formal languages
  - Abstract representation can be defined more easily
- Students use different transformations and modifications to hide plagiarism in the source code
  - While keeping the original functionality of the program
  - Lexical changes
    - Renaming of identifiers, addition or deletion of comments, changes in formatting and output…
  - Structural changes
    - Reordering of expressions, statements or code blocks, loop transformations, addition of superfluous code, function inlining or vice-versa, changes in scoping…

Source code plagiarism detection (3)

- Source code similarity detection tools use different preprocessing techniques to eliminate the effects of lexical changes and structural changes
  - Tokenization, abstract representation
- Structure-oriented comparison based on string matching is the most popular approach for similarity detection
  - Source codes are converted to a sequence of tokens
  - Token sequences are compared using comparison algorithms
  - Several techniques & algorithms
    - *String matching, parse trees, program dependency graphs*
  - GST, *Karp Rabin, Winnowing* algorithms for string matching
    - Computationally viable in academic context of massive courses

Similarity detection systems (1)

- Numerous source code similarity detection systems reported in the open literature
  - Most of them are developed by academic community
- Several key features of such systems:
  - Supported programming languages (frontends)
  - Extendibility
  - Detection algorithms used
  - Presentation of results
  - User interface
  - Security
  - Exclusion of template code and small files
  - Comparison with history or external resources

Similarity detection systems (2)

- Measure of Software Similarity (Moss) from Stanford University
  - http://theory.stanford.edu/~aiken/moss/
  - Web-based system, command-line user interface
  - Support for more than 23 different languages
    - C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, Haskell, Lisp, assembly…
  - Winnowing algorithm, based on k-grams and fingerprinting
- JPlag from Karlsruhe Institute of Technology
  - https://github.com/jplag/JPlag
  - Stand-alone system, CLI & GUI
  - Open-source
  - Support for more than 10 different languages
    - Java, C#, C/C++, Python 3, Go, Rust, Kotlin, Swift, Scala…
  - RKR-GST string matching
- Similar presentation of results in HTML
- Moss is more robust to changes, but JPlag is more precise

Drawbacks of existing systems

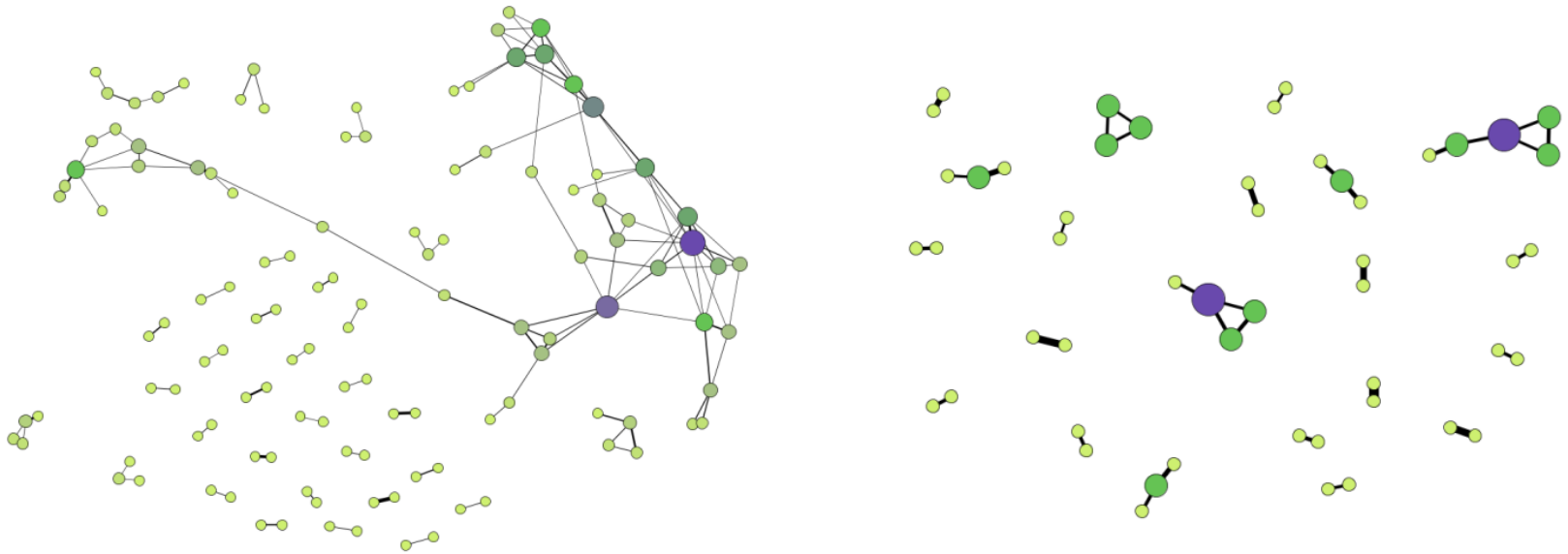
- Mostly focused on the similarity detection stage in plagiarism detection
- Counter-intuitive user interfaces
- Presentation of results is limited to set of HTML pages
  - Lacking meaningful visualization
- Collaboration analysis, grouping and clustering of similar assignment is rarely supported
- Processing time and scalability for massive courses in academic environment are not negligible

Source code plagiarism detection in practice

- UB-SEE has common freshman year and two large IT-related study programs
  - Common freshman year (~720 students per year)
  - Software Engineering (~200 students per year)
  - Computer Engineering and Informatics (~120 students per year)
- Plagiarism is most commonly found in programming courses
  - Programming in Python and C, object-oriented programming courses
    - Smaller, but rather frequent homework assignments
  - Operating systems, compilers, web application programming courses
    - Larger programming projects

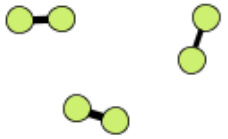
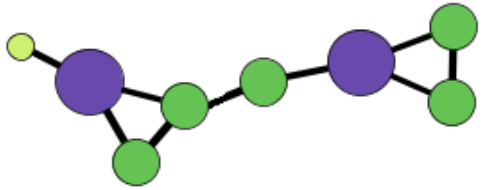
Plagiarism investigation (1)

- Presentation and visualization of results in the form of a graph (network)
  - Undirected, weighted network
  - Social network approach
  - Filtering with threshold is important to improve detection



Plagiarism investigation (2)

- Social network analysis methods can be used to characterize plagiarism network
  - Degree centrality, betweenness centrality, eigenvector centrality are valuable in collaboration analysis
  - Community detection algorithms
  - Discovering collaboration patterns

PAIR	STAR	GROUP	MIX
			
Degree centrality equals to 1, betweenness centrality equals to 0	High degree, betweenness, and beta centrality	Low degree, and betweenness centrality, high clustering coefficient	Low degree centrality, high betweenness centrality

Conclusion

- Plagiarism is a serious threat to the regularity of examination process
  - Both in textual documents and source code
  - Different aspects of fighting this malpractice
- Serbia is not exception to the rest of the world
  - Tools adaptation needed for south Slavic languages
  - Plagiarism in programming assignments is present
- The importance of software tools and their future development
  - Improving visualization and presentation of results
  - Integrating more contextual information about students
  - Using machine learning and AI techniques to improve similarity confirmation and plagiarism investigation might be the future
    - Decision systems

References

- Mišić M., Šuštran Ž., Protić J.,
  "A Comparison of Software Tools for Plagiarism Detection in Programming Assignments",
  International Journal of Engineering Education, Vol. 32, No. 2, pp. 738-748, 2016., ISSN: 0949-149X, IF 2015: 0.559
  http://www.ijee.ie/latestissues/Vol32-2A/13_ijee3202ns.pdf
- Mišić M., Protić J., Tomašević M.,
  "Improving Source Code Plagiarism Detection: Lessons Learned",
  invited paper, 25th Telecommunications forum TELFOR 2017, Belgrade, Novembar 2017., pp. 856-864, ISSN/ISBN: 978-1-5386-3072-3
  https://ieeexplore.ieee.org/abstract/document/8249481

Thanks!

Questions?

Textual and Source Code Plagiarism in Academic Environment:
a Serbian perspective

Marko Mišić (marko.misic@etf.bg.ac.rs)
Assistant Professor

University of Belgrade
School of Electrical Engineering
Department of Computer Engineering and Informatics
Serbia

Plagiarism detection conference 2022