

# VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

---

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program: Aplikovaná informatika

Obor: Informační systémy a technologie

## **Možnosti automatické detekce plagiátů**

Diplomová práce

Michal Hauzírek

Vedoucí práce: Ing. Luboš Pavlíček

Recenzent: Ing. Jan Mach

květen 2007

## ZADÁNÍ DIPLOMOVÉ PRÁCE

**Jméno** : Hauzírek Michal

**Obor:** : informační systémy a technologie

Vedoucí katedry Vám ve smyslu nařízení vlády o státních závěrečných zkouškách a státních rigorózních zkouškách určuje tuto diplomovou práci:

**Téma** : Možnosti automatické detekce plagiátů

**Osnova:**

1. Úvod do problematiky
2. Algoritmy pro detekci plagiátů
3. Srovnání dostupných řešení
4. Návrh systému detekce plagiátů
5. Závěry

**Seznam literatury :**

1. Clough, Paul: Plagiarism in Natural and Programming Languages: An Overview of Current Tools and Technologies. Research Memoranda, CS-00-05, Department of Computer Science, University of Sheffield, UK, 2000.  
<http://ir.shef.ac.uk/cloughie/papers/plagiarism2000.pdf>.
2. Zeidman, Bob: Tools and algorithms for finding plagiarism in source code. Dr.Dobb's Journal 6/2004.

**Vedoucí diplomové práce: Ing. Luboš Pavlíček**

**Datum zadání diplomové práce: 04/05/2006**

.....  
Vedoucí katedry

.....  
Děkan

V Praze, dne

## **Poděkování**

Rád bych tímto poděkoval vedoucímu své práce Ing. Luboši Pavlíčkovi  
za podporu při její tvorbě a mnohé cenné podněty.

## **Prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 1. 5. 2007

Michal Hauzírek

## **Abstrakt**

Tato diplomová práce se zabývá možnostmi automatické detekce plagiátů. Cílem je zejména poskytnout na jednom místě relevantní informace těm, kteří uvažují o použití případně vývoji nástroje pro detekci plagiátů. V teoretické části zejména zobecňuji, na příkladech demonstruji a uvádím do souvislostí, případně doplňuji konkrétní informace získané z různých zdrojů zabývajících se touto problematikou. Dále pak tyto informace aplikuju, testuji a rozvíjím.

V první části stručně uvádím čtenáře do problematiky plagiátorství a současného stavu zejména ve vzdělávacích institucích a do obecných možností a omezení automatické detekce. Dále vysvětluji rozdíly mezi různými typy nástrojů pro detekci plagiátů a možnostmi jejich využití. V následující části se zabývám základními metrikami používanými nejčastěji v různých nástrojích jako rozhraní mezi funkčností a uživatelem a jejich vzájemnými vztahy. Představuji i několik zajímavých a inspirativních alternativních způsobů detekce.

Velmi podrobně a po fázích popisují princip nejpoužívanějšího způsobu detekce plagiátů v přirozeném jazyce a dále některé další přístupy včetně těch užívaných pro detekci v programovacích jazycích. Zabývám se také teoretickými možnostmi porovnání různých nástrojů.

Velký prostor je v praktické části věnován popisu, srovnání a testům vybraných dostupných řešení pro detekci plagiátů a to jak komerčních, tak poskytovaných zdarma. Největší důraz je přitom kladen na jejich fungování v českém jazykovém prostředí. Druhým praktickým přínosem práce je koncept návrhu systému pro detekci plagiátů a jeho začlenění do komplexního programu prevence plagiátorství.

## **Abstract**

This diploma thesis concerns with possibilities of automatic detection of plagiarism. Its main aim is to provide relevant information for those who consider using or developing plagiarism detection tool. In the theoretical part, I mainly generalise particular information acquired from various relevant sources, present them in relations, and demonstrate by examples. In the next part, I apply, test, and further develop that information.

In the first part, I introduce the reader to the issue of plagiarism and contemporary situation mainly in educational institutions and to the general possibilities and limitations of automatic plagiarism detection. Further, I explain the differences between various types of tools and possible ways of using them. In the next part, I deal with basic metrics used often in such tools as an interface between the tool's internals and the user. Otherwise I also present a few alternative and inspiring ways of detection.

I describe in detail and by phases the principle of the most used way of detecting plagiarism in natural language. I describe some other ways to detect including some used for programming languages too. I also consider theoretical possibilities of comparing particular tools.

Considerable volume of the practical part is used for description, comparison, and tests of some chosen available solutions for plagiarism detection. I have chosen both commercial tools as well as those available free of charge. The emphasis is very much on testing its functionality in Czech language environment. Second practical contribution of this thesis is my concept of plagiarism detection system and its integration into a complex plagiarism prevention programme.

## Obsah

Úvod.....	10
1 Plagiátorství a plagiáty.....	12
1.1 Plagiátorství a jeho formy.....	12
1.2 Plagiáty.....	13
1.3 Stručná historie plagiátorství a boje s ním.....	14
1.3.1 První zmínky o automatizované detekci.....	14
1.3.2 Rozvoj plagiátorství.....	14
1.3.3 Současná situace ve světě a u nás.....	15
1.4 Příčiny a prevence plagiátorství.....	16
1.5 Limity automatické detekce plagiátů.....	16
2 Klasifikace nástrojů pro detekci.....	17
2.1 Tradiční typologie.....	17
2.1.1 Atribute counting.....	17
2.1.2 Structure metric.....	18
2.2 Nová typologie.....	18
2.2.1 Typ dokumentů.....	18
2.2.2 Jazyk a formát dokumentů.....	19
2.2.3 Způsob detekce plagiátů.....	19
2.2.4 Dostupnost nástroje.....	21
3 Alternativní přístupy k detekci plagiátů.....	24
3.1 Neviditelné značkování.....	24
3.2 Editor neumožňující plagiátorství.....	26
3.3 Doplňování textu.....	28
3.4 Shrnutí.....	29
4 Metriky používané v nástrojích pro detekci.....	30
4.1 Úvod k části o metrikách v rámci nástrojů.....	31
4.2 Symetrická metrika podobnosti.....	32
4.3 Druhá varianta podobnosti.....	33
4.4 Asymetrická metrika obsahu.....	36
4.5 Symerizované nesymetrické metriky.....	38
4.6 Shrnutí.....	39
5 Detekce plagiátorství volného textu.....	41
5.1 Specifika přirozeného jazyka.....	41
5.2 Porovnání reprezentací dokumentu.....	42
5.3 Aplikace pravidel předzpracování.....	43

---

5.4 Volba základních jednotek.....	44
5.4.1 Volba délky jednotky.....	45
5.5 Volba hashovací funkce.....	49
5.6 Volba jednotek pro reprezentaci.....	50
5.6.1 Počet jednotek v reprezentaci.....	50
5.6.2 Volba konkrétních jednotek.....	52
5.6.3 Vylepšení strategií.....	53
5.6.4 Shrnutí volby jednotek pro reprezentaci.....	54
5.7 Porovnávání.....	55
5.8 Převod dokumentů do textového formátu.....	55
5.9 Přímé porovnání obsahu dokumentů a jiné přístupy.....	56
5.9.1 Vyhledávání v řetězcích.....	57
5.9.2 Další práce s jednotlivými prvky.....	58
5.9.3 Nejdelší společná část.....	59
5.9.4 Komprese.....	59
5.10 Snížení počtu porovnání.....	60
5.10.1 Metadata u dokumentů.....	61
5.10.2 Nové a staré dokumenty.....	61
5.11 Shrnutí.....	63
6 Specifika detekce plagiátorství u zdrojových kódů.....	64
6.1 Specifika zdrojových kódů programů.....	64
6.2 Zpracování zdrojových kódů.....	66
6.3 Univerzální zpracování zdrojových kódů.....	67
6.4 Podobné problémy.....	68
7 Kritéria testování nástrojů pro detekci plagiátů.....	69
7.1 Metriky nástrojů na detekci.....	69
7.1.1 Základní měření výkonnosti nástrojů.....	69
7.1.2 Srovnávání výkonnosti různých nástrojů.....	72
7.1.3 Bezpečnost nástrojů.....	73
7.1.4 Několik poznámek ke kapacitě a době odezvy.....	74
7.2 Sestavení korpusu pro testování plagiátů.....	74
7.2.1 Intrakorpální testy.....	75
7.2.2 Extrakorpální testy.....	76
7.2.3 Další dílčí testy.....	76
7.3 Shrnutí.....	77
8 Dostupná řešení.....	78

---

8.1 Výběr nástrojů.....	78
8.2 Komerční nástroje a služby.....	80
8.2.1 CatchItFirst.....	80
8.2.2 Glatt Plagiarism Services.....	82
8.2.3 CopyCatch.....	83
8.2.4 Eve 2.....	84
8.2.5 Plagiarism Finder.....	86
8.2.6 MyDropBox.....	88
8.2.7 TurnItIn.....	92
8.3 Volně dostupné nástroje a služby.....	95
8.3.1 WCopyFind.....	95
8.3.2 Pl@giarism.....	97
8.3.3 Ferret.....	99
8.3.4 CISE tools.....	102
8.3.5 JPlag.....	104
8.3.6 Sherlock.....	106
8.4 Shrnutí a doporučení.....	108
9 Návrh systému detekce plagiátů.....	112
9.1 Předpoklady návrhu.....	112
9.1.1 Instituce.....	112
9.1.2 Integrace do programu prevence plagiátů.....	112
9.2 Požadavky na nástroj ODPUST.....	114
9.3 Koncept návrhu systému ODPUST.....	115
9.3.1 Jádro systému.....	115
9.3.2 Databáze dokumentů a obecně datové základna.....	116
9.3.3 Akvizice dokumentů.....	117
9.3.4 Hlavní funkce jednotlivých modulů.....	118
9.4 Shrnutí.....	120
Závěr.....	121
Rejstříky vložených tabulek a obrázků.....	122
Přehled literatury a použitých zdrojů.....	124
Terminologický slovník.....	130

## Úvod

Plagiátorství je závažným problémem nejen ve vzdělávacích a vědeckých institucích a má velmi dlouhou a bohatou historii. V poslední době se však stává celosvětově velmi rozšířeným a mnozí se proti tomu snaží bojovat. Souvisí to zejména s rozvojem informačních a komunikačních technologií v minulých letech a s jejich stále větším pronikáním do každodenního života. Velká většina nových textových dokumentů je vytvářena v elektronické podobě, obrovské množství jiných dokumentů často na stejná téma je snadno dostupné prostřednictvím Internetu. Tato situace činí potenciální plagiátorství velmi snadným a jeho odhalení klasickými přístupy náročné. K tomu se přidává i nízké povědomí zejména mladších generací o informační etice a souvisejících otázkách. Důsledkem je růst případů plagiátorství (at' již odhalených nebo přiznávaných v anonymních výzku-mech) a to, že mnohé instituce tuto situaci považují za velmi závažnou a podnikají kroky k její nápravě. V zahraniční, zejména na západ od nás, jsou viditelné kroky podnikány již minimálně pět a více let. V ČR se zdá, že také místní instituce si začínají právě v této době uvědomovat závažnost situace a přijímat příslušná opatření.

Jednou z možností řešení je zapojení informační a komunikační technologie, která plagiátorství nahrává, do boje proti němu. Historicky je tato úvaha stará již několik desítek let, ale právě v posledních několika letech vlivem výše zmíňovaných faktorů prožívá velký nástup. Tématem této práce jsou právě možnosti automatické detekce plagiátů. Zaměřuji se zde jednak na všeobecné teoretické základy a historické konotace, ale také na popis a srovnání nejčastějších přístupů a otestování a porovnání některých dostupných řešení.

Na vymezeném prostoru se samozřejmě nemohu věnovat na jedné straně veškerým netechnickým (např. pedagogickým, etickým, právním atd.) souvislostem problematiky plagiátorství a na druhé straně technologickým detailům implementace konkrétních nástrojů. Takto široký záběr by jen obtížně hledal svého čtenáře, který by byl schopen využít detailní informace z různých oborů. Zvolil jsem proto střední cestu a zaměřil svou práci zejména do pomyslného těžiště problematiky automatické detekce plagiátů k obecné charakteristice nejčastěji používaných technologií a přístupů.

Nezabývám se detailně různými příčinami a sociálními souvislostmi plagiátorství a v první kapitole pouze velmi stručně uvádím do problematiky. Nepředkládám také detailní popisy a analýzy konkrétních algoritmů, které mohou být v nástrojích pro detekci plagiátů použity. Zaměřuji se programově na obecná východiska různých způsobů a metod detekce. Popisuji obecná schémata a zobecňuji poznatky získané z různých konkrétních zdrojů. Strukturu a celý přístup práce podřízuji jejímu účelu a předpokládaným čtenářům, kteří by ji mohli chtít využít. Těmi by měli být lidé, kteří v blízké budoucnosti vážně uvažují o nasazení některého řešení založeného na automatické detekci plagiátů zejména ve vzdělávací instituci. Může jít jak o výběr existujícího typového softwaru, tak také vlastní implementaci případně kombinaci obojího. Zde poskytnuté informace by měly sloužit k poměrně kvalitnímu zorientování se ve věcné problematice a možných standardních způsobech jejího řešení.

Tomu zcela odpovídají i stanovené cíle práce. Prvním je seznámit čtenáře s různými obecnými typy nástrojů pro detekci plagiátů a jejich specifiky včetně základních typů používaných metrik. Dalším z cílů je popsat a zobecnit nejčastější principy fungování detekce plagiátů při práci zejména s volným textem v přirozeném jazyce.

Kromě uspořádání, zobecnění a případného doplnění existujících dostupných informací bude pro českého čtenáře se zájmem o možnosti automatické detekce plagiátů jistě přínosné také plánované srovnání vybraných existujících nástrojů a služeb a to zejména (ale nejen) z hlediska schopnosti práce v českém prostředí.

Na základě získaných informací a zkušeností bych se dále také rád pokusil o návrh konceptu komplexního systému pro automatizovanou detekci plagiátů v místním prostředí. Cílem není detailní návrh ani implementace, ale základní návrh funkcionality a návaznosti systému na své okolí. Budu se přitom snažit toto okolí pojmut komplexně a začlenit náš navrhovaný systém do širšího kontextu.

Protože se tato problematika a zejména subjekty na příslušném trhu vyvíjí poměrně dynamicky, připouštím a zároveň upozorňuji, že některé zde popisované postupy či nástroje mohou být v blízké době nahrazeny či zrušeny. I proto jsme se snažil příliš nezabýdat detailů konkrétních nástrojů, ale raději použít metodu syntézy zobecněných poznatků. Rovněž upozorňuji, že jsem vyčázel zejména z dostupných otevřených zdrojů (cílem rozhodně nebylo pátrat po existujících ale neveřejných implementacích různých nástrojů) a je tak možné, že jsem zde nepostihl veškeré aspekty problematiky, které v nich nejsou popsány a mohou tvořit součást know-how některých subjektů.

Přestože jsem jediným autorem tohoto textu, dovolím si jeho v podstatnou část psát v tzv. autorském plurálu (tj. první osobě množného čísla), který je v českém prostředí u podobných textů poměrně obvyklý.

# 1 Plagiátorství a plagiáty

*If you steal from one author, it's plagiarism; if you steal from many, it's research.*

Wilson Mizner, americký dramatik (1876–1933)<sup>1</sup>

V této kapitole uvádíme do problematiky plagiátů a plagiátorství. Velmi stručně nastíníme základní problematiku plagiátorství, definujeme pojem plagiátu a budeme diskutovat vhodnost jeho obecné definice z hlediska využitelnosti pro automatizovanou detekci. Velmi stručně také popíšeme historický vývoj automatizované detekce, a její obecné limity.

## 1.1 Plagiátorství a jeho formy

Definic plagiátorství existuje celá řada a liší se jak svou šíří záběru, tak oblastí pro kterou jsou určeny. Většinou mají ale společnou základní myšlenku: plagiátorství se dopouští ten, kdo vydává cizí dílo, myšlenku, nebo jiný výtvar za vlastní<sup>2</sup>.

Plagiátorství a jiné případy podvádění a nečestného jednání se často překrývají a splývají. Například [Collins2005] uvádí jako příklady různých forem plagiátorství následující činnosti.

- Zkopírovat podstatné části práce z jiných zdrojů aniž jsou tyto označeny.
- Doslovně zkopírovat blok cizího textu a neoznačit ho vhodnou formou (například kurzívou, uvozovkami) a případně ani neuvést zdroj.
- Použít práce jiného studenta bez jeho vědomí a uvedení původního zdroje<sup>3</sup>
- Koupit práci od někoho jiného nebo ji nechat napsat někoho jiného a vydávat ji za svou.
- Přeložit cizího textu z jiného jazyka a jeho vydávání za vlastní tvorbu (např. opět neuvedením zdroje)
- Uvádět kolektivní práci jako svou vlastní aniž je uvedeno, kdo se na ní dále podílel.

Tyto prohřešky jsou různě závažné a některé z nich se mohou projevit v různé míře. V praxi pak bývá třeba rozlišovat úmyslné plagiátorství od neznalosti a neschopnosti korektní práce se zdroji. Je důležité si také uvědomit, že ani automatizovaný nástroj není a nemůže být na základě pouhého textu schopen odhalit ani všechny výše uvedené příklady. I velmi dobrý nástroj odhalí nanejvýš tři, velmi výjimečně čtyři první uváděné příklady a to ještě za velmi specifických podmínek.

---

1 Stejný nebo velmi obdobný citát bývá ale připisován také americkému právníkovi jménem John Burke, který žil přibližně ve stejné době (1859–1937). Dokonce i nad citáty o plagiátech se tedy vznáší podivný stín pochybností ohledně plagiátorství.

2 Někdy bývá jako plagiátorství označováno i napodobování tedy naopak snaha vydávat vlastní neoriginální výtvar za originál. Tak je tomu například u uměleckých děl případně značkových výrobků nebo cenin. Plagiátorstvím v tomto smyslu se zde nezabýváme. Raději bychom ho označili termínem falzifikátorství nebo padělání.

3 Ani s jeho vědomím či dokonce povolením se nestává takové jednání čestným, pokud autor tuto práci např. před svým vyučujícím vydává za své dílo.

Plagiátorství se týká různých oblastí lidského konání. My se v tomto dokumentu zaměříme zejména na oblast vyšších stupňů vzdělávání a akademické sféry. Dále v textu popisované metody jsou však většinou velmi obecné a s jistými výhradami snadno aplikovatelné i do jiných oborů.

## 1.2 Plagiáty

S pojmem plagiátorství velmi úzce souvisí pojem plagiát. Je zřejmé, že plagiát je produktem plagiátorství. Jde tedy o dílo (at' už text v přirozeném jazyce, zdrojový kód programu, výtvarné dílo nebo třeba audiovizuální počin), které vychází z, nebo při jehož vzniku bylo použito děl cizích, a které je přitom vydáváno za výtvor originální, původní.

Z hlediska detekce plagiátů má ovšem takováto definice jednu naprosto zásadní vadu. Odvolává se totiž na proces vzniku díla. Ten při detekci ale nemáme pod kontrolou a nejsme zpravidla schopni ho spolehlivě reprodukovat. Situace, kdy tuto znalost máme, je velmi výjimečná. Proto budeme pro naše účely plagiáty dle prvního odstavce, tedy ty, o nichž jistě víme, že vznikly procesem plagiátorství<sup>4</sup>, nazývat skutečnými plagiáty.

Některé důmyslné přístupy k odhalování plagiátů, které pracují již na úrovni procesu vzniku dokumentu, budou popsány dále ve zvláštní kapitole 3. Jsou sice teoreticky poměrně přesné v odhalování skutečných plagiátů, ale mají některé nevýhody, které jejich využití značně limitují. Většina metod detekce se tak zaměřuje na odhalování plagiátů na základě děl samotných respektive zpracování výsledných dokumentů. V těchto případech, kdy nemáme možnost dohlížet přímo na vznik díla a na to, zda při něm nedochází k plagiátorství, si musíme pomoci jinak.

Tyto metody tedy nemohou nikdy naprosto přesně oddělit původní díla od skutečných plagiátů (dle definice v prvním odstavci). Místo toho pracují spíše s různými metrikami, které vyjadřují, jak moc je dané dílo podezřelé z toho, že je plagiátem. Tuto metriku ale nelze obecně chápat ani jako pravděpodobnost, že je daný dokument plagiátem. Blíže se různým typům metrik venujeme v kapitole 4. Hodnoty metrik jsou určovány zejména na základě obsahu různých podezřelých znaků v dokumentech. Názor na to, co je a co není podezřelým znakem přitom není v detailech obecně sdílen. Jsou určovány spíše heuristicky, přibližně, na základě zkušeností. Předpokládáme přitom většinou, že různé takové podezřelé znaky se ve větší míře objeví v díle, které je plagiátem.

Klasickým podezřelým znakem je shoda obsahu dvou dokumentů. Ale zda to má být doslovná shoda na úrovni odstavce nebo věty, nebo shoda významu kupříkladu u dvou zdrojových kódů (a co je to taková shoda významu), nebo cosi mezi tím například přibližná shoda s využitím synonym, to již není tak jednoznačné. Každý jednotlivý nástroj (případně jeho konfiguraci) tak vlastně můžeme chápat jako jakousi individuální praktickou heuristickou definici plagiátu.

Níže uvádíme několik málo potenciálních podezřelých znaků. Více jich je možné najít například v [Haris2004] a [Clough2000], odkud jsme je i částečně převzali. Všechny tyto znaky mohou (ale nemusejí) ukazovat na podezřelé dokumenty nebo jejich skupiny a mohou nám tedy pomoci odhalit plagiáty. Zdaleka se ale nejedná o kompletní výčet a žádný z těchto znaků samostatně ani ve skupině s ostatními nemůže být jasným a nezvratným důkazem o případném plagiátorství. S většinou těchto znaků také standardní nástroje pro detekci nepracují.

---

4 Třeba proto, že jsme je tak sami tak vytvořili nebo nějakým způsobem odhalili.

- Použití neobvyklých slov (nebo slov, která autor v textu nevysvětluje či je pochybnost o tom, že jim vůbec rozumí)
- Stejné pravopisné chyby v různých dokumentech
- Stejná struktura a členění textu (např. do kapitol) v různých dokumentech
- Změny formátování v různých částech textu
- Velmi podobná distribuce (počet a frekvence) slov v různých dokumentech
- Změny v použité slovní zásobě a stylu<sup>5</sup> v různých částech textu
- Dlouhé shodné pasáže v různých dokumentech

V praxi se, kvůli relativně snadnějšímu počítačovému zpracování, nejčastěji používá zejména poslední uváděný znak. Některé (zejména intrinsic – viz dále) nástroje mohou pracovat i se dvěma předposledními.

## 1.3 Stručná historie plagiátorství a boje s ním

### 1.3.1 První zmínky o automatizované detekci

Ruční vyhledávání plagiátů (byť podpořené například klasickým počítačovým vyhledáváním) je přeci jen velmi časově náročné. Již poměrně dříve se tak vcelku logicky objevily úvahy o jeho automatizaci. První zmínky tak pochází již ze 70. let dvacátého století, tedy dříve před masovým rozšířením osobních počítačů. Tehdy v digitální podobě samozřejmě nebylo tolik dokumentů s přirozeným textem. V tomto případě šlo zejména o zdrojové kódy v různých tehdy používaných programovacích jazycích. A plagiátorství tehdy trápilo hlavně akademické pracovníky.

První známé přístupy k automatizaci detekce plagiátů se objevily právě v hlavách vyučujících programování ([Ottenstein1976], [Donaldson1981]). Šlo o přístupy odlišné od těch dnešních, což vycházelo z tehdy dostupné výpočetní kapacity. Ale právě vyučující a zejména ti, kteří se specializovali na programování byli i v následujících letech často u zrodu nových metod pro detekci plagiátů. Jak uvidíme v dalších kapitolách tohoto textu, jejich role je v tomto oboru velmi významná, a to nejen pro oblast detekce ve zdrojových kódech.

### 1.3.2 Rozvoj plagiátorství

Rozvoj nástrojů pro automatizovanou detekci plagiátů ve volném textu psaném přirozeným jazykem byl ale mnohem pomalejší. To je dáné několika faktory. Jednak strojové zpracování přirozeného textu je mnohem náročnější úkol, než zpracování zdrojových kódů, které jsou již ze své podstaty určeny ke zpracování počítači. Navíc většina nástrojů pro odhalování plagiátů pracuje s porovnáváním dokumentů nebo jejich reprezentací. K tomu je potřeba mít k dispozici případně zdrojové dokumenty v elektronické podobě. To dlouho nebylo běžné a donedávna nebylo standardní ani odevzdávání dokumentů elektronickou formou.

S masivním rozvojem osobních počítačů v domácnostech i školách nižších stupňů zejména v devadesátých letech a od poloviny devadesátých let také silným rozvojem Internetu, dochází stále více

5 Mimo jiné například střídání 1. osoby jednotného čísla (já), 1. osoby množného čísla (my), případně 2. osoby množného čísla (vy).

k digitalizaci psaných dokumentů a zároveň k tomu, že velké množství jich je veřejně a snadno dostupných.

Také technologický náskok mladší generace studentů oproti vyučujícím dále vede k tomu, že případné plagiátorství z bohatých zdrojů Internetu je prakticky velmi obtížně detekovatelné. Navíc digitální podoba dokumentů umožňuje velmi snadno kopírovat dlouhé pasáže textu. Na Internetu jsou navíc zveřejňovány kompletní práce různých kvalit pro standardní frekventovaná školní téma-ta<sup>6</sup> a to buď veřejně volně ke stažení, nebo i za poplatek<sup>7</sup>.

S živelným rozšířením Internetu do domácností v posledním desetiletí a způsobem jeho využívání souvisí také možný pocit jeho uživatelů, že co je na Internetu, je volně dostupné a použitelné pro všechny a zdarma. To je podpořeno i technologicky prakticky bezproblémovým a hojně využívaným digitálním šířením produktů jako je hudba, filmy a software<sup>8</sup>.

### 1.3.3 Současná situace ve světě a u nás

To všechno přispělo k tomu, že situace ohledně plagiátorství zejména na univerzitách začala být po-važována za velmi neutěšenou a dlouhodobě neudržitelnou. Na mnoha západních univerzitách již zhruba kolem roku 2000 začaly vznikat různé komise, skupiny a projekty pro boj s plagiátorstvím<sup>9</sup>. Za komerční nástroje a služby, které mají k dokonalosti rozhodně daleko, byly a jsou utráceny mnohdy nemalé částky (viz např. tabulku 13 na straně 94 a celou kapitolu 8.2).

Vývoj různých metod a nástrojů pro automatizovanou detekci plagiátů volného textu se tak od této doby rozvíjí velmi dynamicky. První články na tato téma se objevují právě v polovině devadesátych let ([Brin1995])<sup>10</sup>. Následuje již zmiňovaný poměrně rychlý vývoj často v režii právě zejména univerzitních pracovníků, nezřídka vyučujících a programátorů. Tento vývoj, zdá se, zatím rozhodně není u konce, i když z dostupných dokumentů z nedávné doby to vypadá, že některé započaté projekty se v posledních letech již nevyvíjejí a mohla by začínat fáze konsolidace.

Objektivní statistiky míry plagiátorství neexistují, případně jednotlivé instituce, pokud je mají, se jimi příliš nechlubí<sup>11</sup>. Existují některé publikované zahraniční studie (například [Culwin2001]) provedené nejčastěji na základě anonymních dotazníků. Přesto se zdá, že zejména vzdělávací instituce si ten to problém stále více uvědomují a i ve zdejším prostředí se začínají objevovat podobné tendenze jako na západě před několika lety. Zdá se tak, že i u nás se bude stále více rozširovat praxe automatizované detekce plagiátů a místní instituce budou tvořit poměrně silnou potenciální po-ptávku po vhodných nástrojích.

---

6 Různé čtenářské deníky, slohové práce na frekventovaná téma jako „Jaro“ nebo „Co jsem dělal o prázdninách“.

7 V americkém prostředí jsou takové služby nazývané Paper Mills.

8 A to ať už v legální nebo často nelegální podobě.

9 Například Plagiarism Detection Research Group na University of Hertfordshire, Plagiarism detection and prevention group při Centre for Interactive Systems Engineering na London South Bank University a mnohé další.

10 Hlavním autorem této téma zakladatelské práce ze Standfordské univerzity je ten Sergey Brin, který na stejně univerzitě o pár let později s kolegou Larry Pagem pracoval na projektu internetového vyhledávače a ještě o něco později založili společnost Google Inc.

11 Z čehož by bylo možné dovodit, že pokud existují, tak tato čísla nejsou příliš pozitivní.

## 1.4 Příčiny a prevence plagiátorství

Není naším cílem se v tomto textu detailně zabývat možnými příčinami plagiátorství a důvody, které k němu některé jedince vedou. K této problematice odkazujeme například na [Haris2004], česky kupříkladu na [Mares2005]. Některí autoři se domnívají, že sklony k plagiátorství ve školství jsou podmíněné také kulturními rozdíly. K této problematice viz například [Hayes2005].

Pro celkový kontext i z hlediska použití nástrojů pro automatickou detekci je však důležité vnímat problém plagiátorství komplexně. Jako každý společenský problém, jej není možné řešit pouze represí. Musí být řešen na různých úrovních a různými přístupy tak, aby bylo pokryto pokud možno celé široké spektrum možných příčin. Některé mohou být eliminovány příslušným vzděláváním, další úpravami v zadávání úkolů, a snižováním příležitostí k plagiátorství, jiné jasným stanovením pravidel a trestů a případnou detekcí.

Pouze propojením všech těchto prvků je možné efektivně bojovat proti plagiátorství a to tak, aniž by to bylo zúčastněnými vnímáno jako primární nedůvěra k nim. Proto je nutné vnímat jakýkoliv nástroj pro detekci plagiátů pouze jako jednu z částí programu jejich prevence. Trochu blíže se o příkladu podobného programu zmiňujeme v kontextu návrhu systému v kapitole 9.1.2.

## 1.5 Limity automatické detekce plagiátů

Je zřejmé, že ale žádný způsob detekce plagiátů nemůže být považován za dokonalý. Různé nástroje se od sebe liší většinou tím, jak rychle jsou schopny dané dokumenty zpracovat, jaké odhalí podezřelé znaky, případně po jak velkých transformacích jsou ještě schopny označit text jako podobný, aniž by přitom prudce rostla pravděpodobnost falešných poplachů.

Ke všem automatizovaným nástrojům je nutno přistupovat tak, že ve skutečnosti neodhalují plagiáty, ale pouze některé podezřelé znaky. Upozorňujeme tedy (a rozhodně ne naposledy v této práci), že není možné se stoprocentně spoléhat pouze na takovéto nástroje. Není možné považovat dokument, který je nástrojem označen, automaticky za plagiát. Není možné takto neoznačený dokument mít za jistě originální. Vždy je třeba ke všem přistupovat individuálně a hodnotit je ve všech souvislostech, které nástroj nemůže brát v úvahu.

Žádný v současnosti existující nástroj například není schopen odhalit plagiát, který vznikl překladem dokumentu z cizího jazyka, byť by ten byl doslovný. Stejně tak některé nástroje z podstaty svého fungování nemohou být schopny odhalit plagiát ze zdroje, který jim není znám.

I proto je vhodné, aby i uživatel věděl, jaké možné kategorie nástrojů pro automatické odhalování plagiátů existují, a co od jednotlivých typů může očekávat. Stejně tak je dobré si ujasnit co, případně proč, prostě některé typy nedokáží. V této základní orientaci by měla alespoň částečně pomoci následující kapitola tohoto textu.

## 2 Klasifikace nástrojů pro detekci

*Crude classifications and false generalizations are the curse of organized life.*

*George Bernard Shaw, irský kritik a spisovatel (1856–1950)*

Aby bylo možné alespoň do určité míry objektivně porovnat různé nástroje pro automatickou detekci plagiátů, je potřeba je nějak jednotně klasifikovat. Různé nástroje používají odlišné přístupy a nezřídka fungují na zcela odlišných principech. Stejně tak různé třídy nástrojů umožňují i z hlediska jejich rutinního použití pouze některé operace. Není proto možné nebrat na to při jejich srovnání, popisu a koneckonců i výběru zřetel. V této kapitole se tedy pokusíme o jednotnou a všeobecnou klasifikaci nástrojů pro detekci plagiátů.

Po prostudování dostupných zdrojů musíme spolu s Thomasem Lancasterem a Fintanem Culwinem [Lancaster2005] konstatovat, že v oblasti typologie nástrojů pro detekci plagiátů doposud nebylo publikováno mnoho teoretických prací. Zdá se dokonce, že touto oblastí se podrobně zabývali pouze zmiňovaní dva autoři, a proto vyjdeme z jejich klasifikace, kterou pro naše potřeby v některých detailech upravíme. Předtím, než se seznámíme s touto typologií, představíme si tradiční historický pohled na klasifikaci nástrojů.

### 2.1 Tradiční typologie

Jak autoři pojmenovávají, z hlediska historického vývoje (který trochu přibližuje kapitola 1.3), se jakési rozdelení těchto nástrojů do dvou skupin objevilo pouze v jejich rané fázi. Jde o nástroje založené na počítání atributů (attribute counting systems) a novější, založené na analýze struktury (structure metric systems). Oba dva tyto přístupy se však uplatňovaly pouze a jedině pro detekci plagiátů ve zdrojových kódech.

#### 2.1.1 Atribut counting

První – starší z nich byl založen na sledování možné podobnosti zdrojových kódů na základě metrik založených zejména na pouhém počtu různých prvků či typů obratů v programu. Šlo zejména o různé metriky složitosti algoritmů.

Často bývá uváděn příklad takového použití Halsteadovy metriky<sup>12</sup>. Zde jsou jako základní proměnné brány počet jedinečných operátorů, počet jedinečných operandů, celkový počet operátorů a celkový počet operandů. Tyto hodnoty bývají využívány k výpočtům různých poměrových ukazatelů<sup>13</sup> složitosti programu. Dobové dokumenty ([Ottenstein1976], [Donaldson1981]) ukazují, že tato čtveřice samotná a její modifikace, přesněji shoda jejich hodnot u různých dokumentů, byly svého

12 Blíže k této metrice a jejímu původnímu i současném použití viz např.  
[http://www.verifysoft.com/en\\_halstead\\_metrics.html](http://www.verifysoft.com/en_halstead_metrics.html)

13 Mimo jiné například délka programu, velikost slovníku, obsah programu, úroveň programu a další.

času využívány také pro odhalení plagiátorství<sup>14</sup>. Dokonce je zmiňována i úvaha o použití podobného přístupu na volný text v anglickém jazyce.

### 2.1.2 Structure metric

Modernější (a výpočetně náročnější) přístupy odhalování plagiátů ve zdrojových kódech programů již nebyly založeny na prostém porovnání základních statistik dokumentů, ale zaměřily se na jejich celkovou strukturu. To umožňuje detailnější analýzu a porovnání činností, kterou mají programy vykonávat. Tyto metody jsou obecně důkladnější a tedy přesnější. Využívány jsou i v některých současných nástrojích.

Jak však upozorňují Lancaster a Culwin, neexistuje žádná přesná definice těchto dvou kategorií. Některé systémy obě možnosti různě kombinují a samotní jejich autoři řadí podobné systémy někdy do jedné a jindy do druhé kategorie. V předchozím textu jsme uvedli, že tyto původní klasifikace zahrnovaly pouze nástroje pro detekci v rámci zdrojových kódů. Zařazování systémů pro práci s volným textem do těchto kategorií by ještě více rozostřilo a zamlžilo jejich hranice a definice. Navíc některé přístupy by nebylo možné zařadit ani do jedné ze skupin a tak bychom nakonec zřejmě skončili u jedné velké skupiny nástrojů pro detekci plagiátů.

Skutečnost, že tato klasifikace vznikla zejména v komunitě programátorů a navíc v době, kdy se k výpočetní technice blíže dostali zejména oni má i další důsledek. Klasifikace je založená pouze na tom, jak systémy uvnitř fungují. Nebere v potaz „čistě uživatelské“ vlastnosti.

Z těchto důvodů ve shodě s oběma autory přistoupíme na klasifikaci novou, která bude pro dnešní dobu a její potřeby vhodnější a užitečnější.

## 2.2 Nová typologie

Vzhledem k různorodosti nástrojů a požadavků na ně, není možné ani rozumné klasifikovat je podle jediného (třeba složeného) kritéria. Výsledkem by bylo buď několik vnitřně nepříliš homogenních skupin zastoupených velkým počtem nástrojů (jako u zmiňované klasické typologie), nebo velké množství skupin zastoupených pouze jedním či dvěma nástroji. Jako ideální se tak jeví klasifikace nástrojů podle různých kritérií, která charakterizují daný produkt v dané oblasti. Připomínáme, že vycházíme z typologie pánnů Lancastera a Culwina, kterou pro své potřeby upravíme a rozšíříme.

### 2.2.1 Typ dokumentů

Velmi důležitou charakteristikou je samozřejmě typ dokumentu, který je schopen systém zpracovat. Základní členění by mohlo být například na *texty v přirozeném jazyce, zdrojové kódy a ostatní binární formáty*.

Mezi zdrojové kódy by bylo možné typově zařadit také některé další textové formáty, které nejsou přímo zdrojovým textem programu. Jde zejména o takové silně strukturované formáty pro popis objektů, kde je význam klíčových slov a struktura dokumentu důležitější než jeho (volný) textový ob-

---

<sup>14</sup> Zajímavé je také z dnešního pohledu lehce úsměvné konstatování autora [Ottenstein1976], že tato metoda je i levná, neboť program pro výpočet metriky již byl vyvinut pro jiné účely (s náklady 300 amerických dolarů) a jedno jeho spuštění pro stořádkový kód studenta přijde na pouhých pět centů.

sah. Příkladem by mohly být některé formáty na bázi XML pro popis vektorové grafiky (SVG) nebo popis datové struktury (XML Schema). Podobně by mohlo jít o některé konfigurační soubory.

Zásadní odlišnosti v typu a strukturovanosti dokumentů většinou vyžadují odlišný způsob zpracování. Dokonce i univerzální nástroje, které podporují oba základní typy dokumentů, mívají algoritmy případně alespoň parametry pro jejich zpracování rozdílné.

### **2.2.2 Jazyk a formát dokumentů**

Jakousi podmnožinou typu dokumentu je jeho formát či jazyk. U zdrojových kódů je otázka formátu většinou bezpředmětná, protože bývají ve formě běžných textových souborů. Mnohem důležitější je pro ně ovšem programovací (nebo značkovací nebo jiný popisný) jazyk. Pokud má nástroj těžit z výhod znalostí struktury konkrétních jazyků, musí být pro každý odlišný jazyk implementován vlastní modul.

Otázka jazyka je relevantní také pro dokumenty s přirozeným textem. Některé pokročilé metody mohou pracovat s nějakou variantou optimalizace, která vyžaduje specializovaný slovník. Například by bylo možné nahrazovat ohebná slova jejich standardizovaným tvarem. Jiný způsob by třeba mohl pracovat se synonymy. Obdobně některé vlastnosti konkrétního přirozeného jazyka mohou být využity ke zlepšení detekce v tomto jazyce, ale nepomohou (nebo naopak uškodí) výsledkům v případě jiného jazyka. Často se například pracuje s běžnými (bezvýznamovými) slovy, která se při porovnání vynechávají (např. členy, spojkami). I u méně sofistikovaných nástrojů, pracujících pouze s textem bez podobných optimalizací, je důležitá i tak obyčejná vlastnost (která ale někdy rozhoduje o použitelnosti daného nástroje v našich podmínkách) jako je podpora české diakritiky.

Zajímavým problémem je i podpora volného textu v různých formátech. Úplným základem je čistý text bez formátování. Některé nástroje si interně dokáží poradit s textovými značkovacími formáty případně i s některými binárními formáty (DOC, PDF). Zdá se, že většinou se jedná o jejich (více či méně dokonalý) převod do čistého textu před samotným zpracováním. Jako potenciálně zajímavé by se z hlediska detekce ale mohlo také jevit, vzít v potaz strukturu textového dokumentu a metadatu obsažená v těchto formátech<sup>15</sup>.

### **2.2.3 Způsob detekce plagiátů**

Dle způsobu detekce plagiátů můžeme v nejobecnějším pohledu rozdělit nástroje na ty, které pracují primárně s vlastním obsahem dokumentu, a ty, které pracují jiným způsobem.

#### **Nástroje nepracující s obsahem dokumentů**

Ty druhé jmenované patří spíše mezi alternativní a nepředstavují aktuální trend. Jejich základní charakteristikou je to, že nejsou založeny na porovnání nebo analýze ať již části nebo celého vlastního obsahu dokumentu, ale na jiném principu. Pracují často naopak s daty, která jsou do dokumentu přidána právě za účelem odhalování plagiátů. Jejich výhodou je zejména přesnost detekce, pokud jsou dobře navrženy, a to, že za určitých okolností jsou schopny odlišit skutečné plagiáty od pouhých

---

15 Například nápadná podobnost formátování (které nebylo předem stanovenovo požadavcích) u různých i obsahově podobných dokumentů by mohla svědčit o podezřelém chování. Stejně tak nekonzistentní formátování textu v dokumentu může být znakem rozsáhlého kopírování z různých zdrojů. Naopak případně nepříliš rozsáhlé a správně označené citace z cizích zdrojů by mohly být z porovnání vyřazeny, nebo dokonce (pokud je identifikovaný zdroj k dispozici) ověřeny.

podezřelých dokumentů. K nevýhodám naopak patří nutnost přidávat metadata do dokumentů a s tím související nižší bezpečnost případně omezení komfortu uživatelů. Příklady několika takových nástrojů i s podrobnějším popisem jejich fungování uvádíme v kapitole 3.

Pokud jde o první skupinu nástrojů, které naopak pracují s obsahem dokumentů, případně ho nějak zpracovávají či porovnávají, můžeme zde dále rozlišovat i způsob porovnání. Tato charakteristika je obecná a neříká ještě nic o konkrétním použitém algoritmu. Je však velmi důležitá pro pochopení toho, co může uživatel od nástroje očekávat. Nástroj může vyhledávat či odhalovat plagiáty několika následujícími způsoby. Přitom výsledek (i pokud bychom předpokládali, že algoritmus porovnání je dokonalý) bude odlišný, protože se budou vůči sobě porovnávat jiné dokumenty.

Množinu dokumentů předloženou nástroji pro odhalování plagiátů označíme jako korpus (corpus).

### ***Intrakorpální nástroje***

Jako *intrakorplání* (intra-corpal) případně pracující v intrakorpálním režimu označíme nástroj<sup>16</sup>, jestliže porovnává pouze dokumenty v rámci korpusu. Takový nástroj není schopen odhalit plagiát vzešlý z jiných dokumentů, než těch, které mu byly v dané důvce poskytnuty (leda by tam bylo několik ze stejného externího zdroje). Tento způsob, kdy jsou si dokumenty rovny (tj. jsou při porovnání považovány jak za možný případný zdroj, tak za možný případný plagiát) také v případě nalezení podobné dvojice vylučuje rozlišení autora a plagiátora.

Implementačně je však takový přístup jednodušší, a pokud není korpus velmi rozsáhlý, umožňuje (při použití vhodného algoritmu) relativně rychle porovnat důkladně všechny dokumenty mezi sebou.

### ***Extrakorpální nástroje***

Jako *extrakorplání* (extra-corpal) případně pracující v extrakorpálním režimu označíme nástroj, jestliže dokumenty v rámci korpusu neporovnává mezi sebou, ale hledá jim podobné v nějaké externí databázi. Takovou databází může být elektronické skladiště prací z minulých let nebo třeba prostředí Internetu zpřístupňované pomocí vyhledávače. Výhodou takového řešení je samozřejmě větší množina zdrojů, vůči kterým lze porovnávat, a tím také větší schopnost odhalit (zejména u volných textů zřejmě častější) extrakorpální plagiáty. V čase se navíc databáze může rozrůstat o další dokumenty a úspěšné plagiátorství se tak stává výrazně obtížnějším. Navíc, pokud (nový) dokument v porovnávaném korpusu odpovídá jinému (starému) dokumentu v databázi, lze předpokládat, že novější dokument je plagiátem a jeho „autor“ plagiátorem<sup>17</sup>.

Nevýhody tohoto řešení plynou zejména z velkého počtu dat, která je třeba zpracovat. Aby byla zachována rozumná rychlosť vyhledávání a odpovídající doba odezvy, není možné porovnávat přímo testovaný dokument s každým v databázi. Je nezbytné použít některé méně přesné, ale výrazně rychlejší metody vyhledávání, které umožní redukovat počet dokumentů pro případné důkladnější porovnání. Tyto metody bývají nezřídka založeny na empirických pozorováních nebo i náhodě.

---

16 V literatuře se přídavná jména intrakorpální (intra-corpal) a extrakorpální (extra-corpal) spojují spíše s podstatným jménem plagiátorství (plagiarism). My ho zde užíváme spolu s podstatným jménem nástroj, případně režim (práce nástroje). Vycházíme totiž z toho, že o způsobu, kterým případný plagiátor pracoval, nic nevíme. Odhalená přítomnost dvou podobných dokumentů v korpusu ještě nemusí znamenat, že šlo o intrakorpální plagiátorství. Stejně tak může text pocházet z jiného zdroje mimo korpus, který oba „autoři“ využili.

17 A na tomto případném odhalení nic nemění ani teoretická možnost, že by mohl existovat ještě nějaký starší dokument, který v databázi není, a ze kterého původně čerpal i autor toho dokumentu, který v databázi je.

### ***Smíšené nástroje***

Za smíšené považujeme ty nástroje, které kombinují oba předchozí přístupy. Tedy umožňují jak porovnat dokumenty korpusu mezi sebou, tak je porovnat je s externí databází.

### ***Intrinsic nástroje***

Zvláštní skupinu tvoří nástroje, které se snaží odhalit plagiát nikoli porovnáním s jinými dokumenty, ale pouze na základě analýzy obsahu samotného dokumentu. Ty bývají označovány jako intrinsic (vnitřní). V takovém případě jsou v dokumentu hledány ty části, které se některou svou charakteristikou vymykají ostatním částem dokumentu. Zjištované charakteristiky bývají založeny například na relativní frekvenci výskytu konkrétních slov v různých částech dokumentu, případně na dalších stylometrických vlastnostech textu (viz např. [MeyerZE2006]).

Nespornou výhodou takového přístupu je možnost nalezení extrakorpálního plagiátu, přestože není dostupná žádná databáze. Z toho plyne i další výhoda, že není potřeba takovou databázi udržovat a spravovat. Výhodou je také rychlosť, kdy není potřeba vzájemně porovnávat velké množství dokumentů. Navíc je možné testovat dokument okamžitě, když je k dispozici a není třeba čekat, až bude vytvořen celý korpus.

Tímto způsobem je možné nalézt pouze takové plagiáty, kde se vyskytují části od různých autorů (či spíše psané různými styly). Pokud je značná část dokumentu tvořena cizím textem pocházejícím z jednoho zdroje (například plagiátor použije vybranou kapitolu z rozsáhlejšího díla, případně spojí její části minimem vlastního textu), detekce nebude úspěšná. Mezi nevýhody je nutné počítat také to, že ani v případě odhalení nedá takový nástroj zřejmý důkaz. Ostatní metody ukáží na podobný dokument, případně zvýrazní přímo odpovídající části. Zde je podkladem pro podezření „pouze“ metrika stylu textu.

Samozřejmě i tento způsob je možné kombinovat s ostatními. Některé nástroje například pomocí takového inspekce textu jednoho dokumentu určí „podezřelé“ části textu. Následně se je pak snaží porovnat s dokumenty nalezenými pomocí internetového vyhledávače. Zde popisovaná charakteristika však má být primárně zaměřená na uživatele aby mu poskytla informace o tom, čeho je nástroj schopen. Za intrinsic nástroje tak budeme považovat pouze ty, které žádné porovnávání mezi dokumenty neprovádějí.

Kromě základního způsobu porovnání je zajímavá také informace o použité metrice porovnání. Metrikám se podrobně věnuje kapitola 4.

### ***2.2.4 Dostupnost nástroje***

Na dostupnost nástroje můžeme pohlížet hned z několika pohledů.

#### ***Lokální a distribuované zpracování***

První charakteristikou dostupnosti může být umístění té části nástroje, která dokumenty zpracovává. Pokud bychom šli do detailů, mohli bychom rozlišovat různé varianty distribuovaného systému. My se zde však opět zaměříme zejména na uživatelské hledisko, postačí nám rozdělení na dvě základní kategorie. Zvláštním případem distribuovaného systému je potom nástroj dostupný přes veřejnou síť Internet.

U distribuovaných systémů je pro uživatele důležitý mimo jiné zejména způsob distribuce dokumentů ke zpracování. Může jít například o zaslání množiny dokumentů najednou. Při jiném způsobu distribuce jsou dokumenty nahrávány zvlášť samotnými studenty. Pokud se jedná o distribuovaný systém provozovaný třetí stranou a poskytovaný jako služba (viz dále), může být důležitým hlediskem také to, jak je dále zacházeno s poskytnutými dokumenty. Ty mohou být kupříkladu zařazeny do databáze a použity k pozdějšímu porovnávání. Některé instituce odmítaly využívat takové služby z důvodů obav o práva svých studentů k jejich dokumentům.

### **Nástroj a služba<sup>18</sup>**

Z hlediska dostupnosti můžeme rozlišovat také poskytnutí nástroje (například udělením licence k užívání softwaru) a poskytování služby odhalování plagiátů. Služba bude zřejmě většinou poskytována po Internetu jako distribuovaná. Distribuovaný systém ovšem může být také získán formou licence a provozován na vlastní infrastruktuře. Kritériem je tedy provozovatel. O službě budeme hovořit, pokud je nástroj samotný provozován třetí stranou<sup>19</sup> a uživateli je poskytován přístup k němu případně výsledky jeho práce.

### **Požadavek na připojení**

Z hlediska uživatele je také důležité, jestli nástroj pro svou práci potřebuje připojení k síti typicky k Internetu. Přirozeně lze očekávat, že všechny distribuované nástroje budou takové připojení vyžadovat. Tuto charakteristiku ale nelze zaměňovat s výše popisovanými (lokální/distribuované zpracování). I některé lokální nástroje vyžadují přístup k Internetu. Jde zejména o ty, které se snaží nalézt dokumenty podobné porovnávanému s pomocí ať už standardních nebo specializovaných vyhledávačů.

### **Licence nástroje**

Nástroje pro odhalování plagiátů vznikají a vznikaly v různých časech a v různých prostředích s různými cíli. To se samozřejmě muselo projevit také na jejich různé dostupnosti z hlediska licence a to jak uživatelské, tak k případným zdrojovým kódům.

Některé nástroje jsou komerční produkty dostupné standardními obchodními cestami, někdy je možné nástroj nebo službu vyzkoušet ve funkčně nebo časově omezené verzi. Některé nástroje, zejména vyvinuté pro osobní potřeby, nejsou veřejnosti dostupné vůbec. Jiné jsou naopak dostupné zcela bezplatně a to často i se zdrojovými kódy. S tím souvisí také dostupnost informací o použitých metodách.

### **Informace o použitých metodách**

Kromě samotných nástrojů je rozdílná také dostupnost informací o použitých algoritmech a řešeních. U komerčních produktů je většinou algoritmus detekce utajený a dostupné jsou pouze údržkovité informace o jeho fungování. Naopak fungování některých jiných nástrojů je poměrně dobře popsáno buď v článcích jejich autorů, nebo alespoň dostupným zdrojovým kódem.

---

18 Ve většině tohoto textu hovoříme o nástrojích. Děláme tak zejména tam, kde hovoříme o principech, možnostech a metodách automatického odhalování plagiátů. Služby zmiňujeme zejména tam, kde jde o využití těchto nástrojů k poskytování služeb. I za službou je tedy většinou jakýsi nástroj provozovaný tím, kdo službu poskytuje.

19 Teoreticky si lze představit i formu interní služby, kdy jsou provozovatel nástroje i uživateli součástí jediné instituce. I v tomto případě ale existuje (byť pouze organizační) oddělení uživatele a provozovatele.

### Aktuální existence

Již pánové Lancaster a Culwin ve svém dva roky starém výzkumu konstatovali, že situace v oblasti detekce plagiátů je poměrně nestabilní a některé ještě před několika lety v literatuře popisované nástroje jsou již nedostupné, nedosažitelné. Některé společnosti poměrně záhy ukončily své aktivity v tomto obooru, nebo je transformovaly do jiné podoby. Jindy zase byl vývoj nástroje součástí akademického projektu, který byl ukončen, případně prací studenta, který již školu opustil. Že tato situace trvá i nadále je se potvrdilo i nyní. Některé nástroje jimi označované jako současné, přestaly existovat a mnohé jiné se objevily.

Níže uvedená tabulka 1 shrnuje ještě jednou přehledně různá námi popisovaná kritéria nástrojů a jejich možné hodnoty.

*Tabulka 1: Přehled kriterií pro klasifikaci nástrojů pro detekci plagiátů z pohledu uživatele*

<b>Kritérium</b>		<b>Možné hodnoty kritéria</b>
<b>doplňující kritérium</b>		
<b>Podporované typy dokumentů</b>	obsah	volný text, zdrojové kódy, binární soubory
	jazyk	český jazyk, anglický jazyk, JAVA, C, ...
	formát	volný text, PDF, ODF, DOC, ...
<b>Způsoby porovnání/detekce</b>		intrakorpální, extrakorpální, intrinsic, smíšený, nepracující s obsahem dokumentů
<b>Dostupnost nástroje</b>	zpracování	lokální, distribuované
	provozovatel	nástroj, služba
	požadavek na připojení k síti	vyžaduje, nevyžaduje
	licence	komerční (platba jednorázová, za časové období, za použití), dostupný zdarma (volně, po registraci)
	informace o použitých metodách	utajené informace, částečné informace, úplné informace, zdrojový kód k dispozici
	aktuální existence	existující, neexistující

## 3 Alternativní přístupy k detekci plagiátů

*If you want to catch something, running after it isn't always the best way.*

*Lois McMaster Bujold, americká autorka sci-fi (1949)*

Ještě předtím, než se podíváme na některé přístupy k detekci plagiátů na základě analýzy obsahu dokumentů, popíšeme si několik velmi zajímavých alternativních přístupů. Některé z nich částečně vycházejí přímo z kontroly nad tím, jak dokumenty vznikají. Jak plyne z naší definice skutečného plagiátu z 1. kapitoly, je to jediný způsob, jak plagiát poměrně spolehlivě odhalit, případně vůbec zamezit jeho vzniku.

### 3.1 Neviditelné značkování

Týmy, ve kterých pracovali Charles Daly a Jane Horne publikovaly dva texty ([Daly2005], [Byrne2004]), kde popsali své zajímavé postupy při odhalování plagiátů. Šlo jim spíše o sociologické aspekty věci a sledovali, jak se plagiátorství v praxi vyvíjí v čase a zda se ho častěji dopouštějí muži nebo ženy. Rovněž sledovali, jak závisí získané známky na tom, jestli se daný student dopouští plagiátorství aktivně (tj. odevzdá plagiát) případně pasivně (tj. poskytne někomu svou práci). Zájemce o tyto statistiky odkazujeme na příslušné články, nás bude zajímat hlavně způsob detekce, který autoři použili.

Z popsaného vyplývá, že autoři požadovali poměrně detailní informace o subjektech svého zájmu (studentech) a jejich chování ve vztahu k plagiátům v relativně delším časovém období. V obou případech pracovali se studenty počítačových oborů v kurzech programování. V prvním případě šlo o studenty prvního semestru na Dublin City University, ve druhém případě o studenty druhého a třetího ročníku University of Natal<sup>20</sup> v Jihoafrické republice. Během semestru studenti odevzdávali několik (20–40) úkolů naprogramovaných zdrojových kódů.

Inspirace pochází ze systémů pro ochranu autorských práv. Ta bývá někdy řešena také na principu vodoznaku (watermark). Každá originální vydaná kopie je unikátně označena tak, aby pokud se objeví případně její nelegální kopie, bylo jasné, která z kopií vedla k jejich vzniku. Svázáním vodoznaku s konkrétním uživatelem/příjemcem tak lze dohledat toho, kdo svou kopii poskytl ke kopírování. Vodoznak může být zjevný (například plné jméno příjemce je odlišnou barvou na psáno na každé stránce diagonálně v pozadí pod textem), pak má i silný preventivní účinek. Může být ale také na první pohled neznatelný (identifikátor je do dokumentu zakódován drobnými změnami rozložení textu na stránce, šírkou mezer mezi řádky a odstavci a podobně.) Podobný princip byl použit také zde.

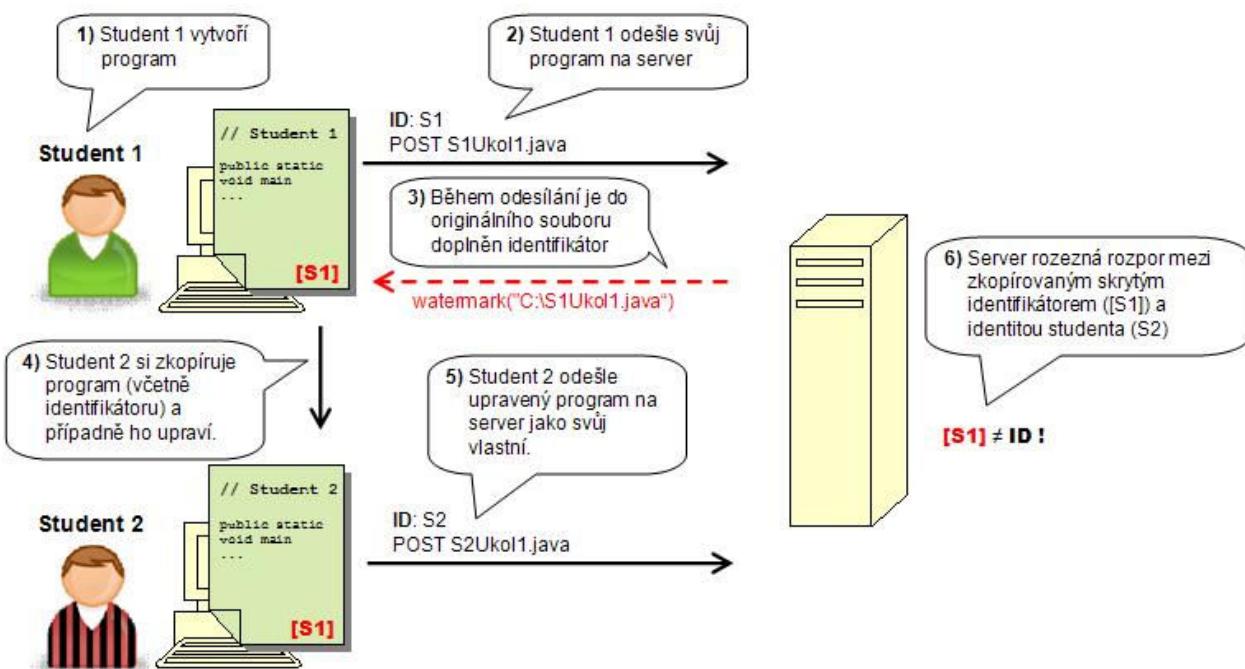
Vtip spočívá v tom, že součástí dokumentu byly také skryté identifikační údaje, které byly do souboru (utavené) přidány během procesu elektronického odevzdávání. Tyto údaje byly zároveň zapsány přímo do originálního souboru na studentově disku. Mezi ně samozřejmě patřil identifikátor studenta – student se před odesláním svého úkolu musel do systému přihlásit, čili jeho identifikace a autentizace byla zaručena. Dále k těmto údajům patřil také kontrolní součet pro ověření integrity údajů. Všechny byly do dokumentu zapsány jako binární číslo a to ve formě netisknutelných

---

<sup>20</sup> Nyní University of KwaZulu-Natal, která vznikla sloučením University of Natal a University of Durban-Westville.

(whitespace) znaků konkrétně mezery a tabelátoru. Konkrétní místo bylo voleno tak, aby byla malá pravděpodobnost, že tento kód bude náhodně změněn nebo smazán<sup>21</sup>.

Pokud student dále někomu poskytl svůj (již označený) dokument, zůstal v něm jeho identifikátor a při odevzdání souboru (byť i podstatně upraveného) někým jiným, byl identifikován jak plagiátor (odevzdávající), tak (podle identifikátoru) ten, který mu svůj původní dokument poskytl. Kvůli dlouhodobějšímu sledování nebyli studenti předem na tuto možnost upozorněni a ani během semestru nebyly případy odhaleného plagiátorství nijak řešeny.



Obrázek 1: Schéma fungování detekce plagiátů pomocí neviditelného značkování

Tento systém samozřejmě není dokonalý, a pokud je prozrazen, je triviální jej obejít. K dalším jeho nevýhodám, které zmiňují i autoři ([Daly2005]) patří to, že je schopen odhalit plagiáty pouze těch dokumentů, které někdo plagiátorovi elektronicky poskytl až poté, co je sám odevzdal (do té doby nejsou označeny). Odhalení se tak vyhne každý, kdo pracuje s kopíí, kterou udělal (at' již autor nebo kdokoliv jiný) před odevzdáním. Značka složená z mezer a tabelátorů může být také poměrně snadno studentem porušena a to jak vědomě, tak zcela nevědomky. Rovněž je potřeba, aby plagiátor kopíroval (případně přepracovával) celý původní dokument a ne pouze jeho část (která neobsahuje značku). Toto by se dalo řešit vkládáním značek na více míst (například do deklarace každé metody), ale ostatní problémy přetrvají.

I když má systém tolik nevýhod, je téměř až s podivem, jakých výsledků se s ním podařilo dosáhnout (a to opakovaně i když na různých univerzitách). Takovýto přístup má navíc oproti klasickým nástrojům pro detekci i několik jedinečných výhod (opět [Daly2005]):

21 V jednom případě šlo o konec deklarace metody main v jazyce Java, v druhém o samotný konec 'formulářového' souboru, který se odevzdával.

- umožňuje odhalit plagiátorství i ve velmi krátkých a jednoduchých zdrojových textech, se kterými pracují začátečníci
- dokáže zcela jednoznačně rozlišit mezi plagiátorem a tím, kdo mu poskytl svůj dokument
- plagiáty odhaluje téměř s naprostou jistotou, neupozorňuje pouze na podezřelé shody a nevyžaduje ruční prozkoumání
- odhaluje plagiáty okamžitě, jak jsou odevzdány, není potřeba čekat až se sejdou úkoly od všech studentů a vytvoří korpus pro porovnání
- je zcela nezávislý na programovacím jazyku použitém v dokumentech

Jeho zásadní nevýhodou ovšem je jeho nízká bezpečnost (ve smyslu schopnosti jej obejít v případě znalosti principu). Pokud se jej podaří dostatečně utajit, je možné ho používat i několik let za sebou (a identifikovat tak i toky úkolů mezi jednotlivými ročníky). To ale předpokládá nevyužití k represivním účelům. Lze uvažovat o kombinaci s jiným „klasickým“ systémem, který by byl používán pro detekci souběžně a mohl by být otevřen použit jako důkaz či argument v případném sporu.

Podobné srovnání provedli také autoři, když porovnávali výsledky takového systému s klasickým detektorem plagiátů ve zdrojových kódech MOSS. Oba systémy detekovaly 37 případů plagiátorství (přesněji 23 v něm zainteresovaných studentů). Ne všechny případy se ale překrývaly.

Nastíněný přístup se tak jeví jako velmi zajímavý doplněk klasickým nástrojům, jeho použití je však značně omezené. Případné nasazení jako primárního systému by mohlo přinést jeho uživatelům značné zklamání (případně falešný pocit, že plagiátorství se jich netýká). Základní předpoklady úspěšného fungování takového systému jsou následující:

- schopnost vložit identifikátor přímo do uživatelského dokumentu po jeho identifikaci (např. při odevzdání)
- utajení takového identifikátoru
- skutečnost, že v případě plagiátorství se s vysokou pravděpodobností stane součástí nového dokumentu také identifikátor

Tyto předpoklady jsou poměrně dobře splněny pro popisovaný příklad detekce ve spíše kratších zdrojových kódech. Pro detekci plagiátů ve volném textu a delších programech je překážkou zejména třetí bod.

### **3.2 Editor neumožňující plagiátorství**

Jistou variací na předchozí možnost, která navíc řeší problémy utajení identifikátoru a jeho přítomnosti od počátku, je praxe kterou popisují Vamplew a Dermoudy ([Vamplew2005]). Ti řeší problém plagiátorství v kurzech programování tak, že vyžadují od studentů, aby pro psaní kódu používali pouze jejich speciální editor s funkcemi, které maximálně omezují možnosti plagiátorství (Anti-Plagiarism Editor – APE).

Také v něm se ukládají do souborů metadata a to již od začátku psaní kódu. Autoři se nezmiňují, že by byly ukládány přímo údaje o uživatelích. Jako příklad uvádějí historii uložení souboru, takže

je zpětně možné zjistit, jak dlouho trvaly úpravy. Navíc pokud by si student otevřel a přepracoval dokument jiného studenta, historie ukládání obou dokumentů by se na začátku shodovala.

Na rozdíl od předchozího přístupu jsou ovšem tato data v souboru viditelná. Opět se jedná o nástroj využívaný při kurzech programování. Metadata jsou tedy uložena jako řádky s komentáři<sup>22</sup>. Proti změně jsou tyto údaje chráněny šifrováním a kontrolními součty (jak metadat, tak samotného obsahu dokumentu). Editor zřejmě odmítne otevřít dokument s pozměněnými (nesouhlasícími) údaji. Z textu neplyne, co se stane, pokud nejsou v otevíraném dokumentu údaje vůbec přítomny, ale předpokládáme, že otevření bude možné, ale při ukládání se tyto údaje do dokumentu nevloží. Protože jsou ale tyto údaje povinnou součástí odevzdávaných souborů a jsou kontrolovány, není možné systém snadno obejít. Nelze tak samozřejmě pracovat ani v jiném editoru.

Kromě tohoto zabezpečení se plagiátorství brání ještě jedním způsobem. Editor má omezenou funkcionality kopírování a vkládání (copy&paste) textu. Text lze poměrně volně kopírovat ven z editoru (podle slov autorů aby studenti mohli zkopirovat část kódu, ke které mají dotaz svému vyučujícímu). Vkládání dovnitř do dokumentů je ale omezeno. Autoři hovoří o jakési vlastní implementaci schránky, která by umožňovala kopírovat a vkládat pouze v rámci jednoho dokumentu a zamínila by vkládání z jiných dokumentů. Jedinou možností plagiátorství by tak bylo ruční přepisování kódu z jednoho okna do druhého<sup>23</sup>.

Elektronicky odevzdávané dokumenty jsou opět automaticky kontrolovány, zda obsahují správná metadata a po odevzdání všech úkolů jsou spuštěny testy možného plagiátorství právě na základě metadat (již zmiňovaná historie ukládání souborů její shoda v různých dokumentech ukazuje na to, že pocházejí ze stejného zdroje).

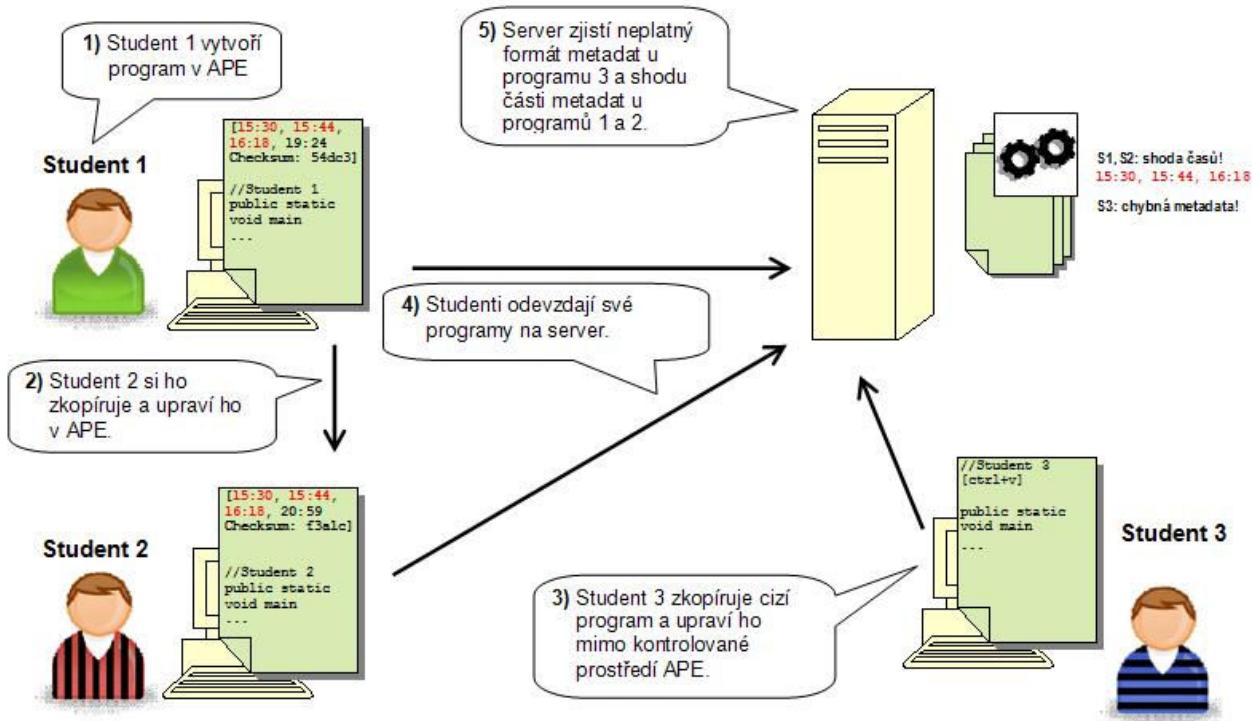
Tento přístup se nám zdá velmi restriktivní. Omezení výběru prostředí na jediný povinný nástroj nemusí být studenty vnímáno příliš pozitivně, přestože implementace zmiňovaného editoru má být založena na bezesporu kvalitním prostředí Eclipse. Ještě rozporuplněji lze vnímat absenci možnosti kopírování textu. Skutečné reakce studentů se nám nepodařilo dohledat, neboť zmiňovaný článek, je z období přípravy prvního nasazení tohoto opatření.

Takový způsob může být účinnou prevencí plagiátorství v hodinách programování ale za cenu poměrně velkých omezení studentů. Teoreticky si lze aplikaci podobného přístupu představit také pro běžné texty. Implementace by mohla spočívat například v integraci nějakého obdobného sledujícího a omezujícího rozšíření do textového editoru.

---

22 To dle autorů do jisté míry omezuje univerzálnost při použití různých programovacích jazyků, protože je potřeba určit, jak jsou v daném jazyce reprezentovány komentáře. Tato schopnost se ovšem od současného vývojového nástroje předpokládá (i třeba kvůli zvýrazňování syntaxe).

23 Což výrazně snižuje přínosy pro plagiátoru. Navíc je toto stále do jisté míry odhalitelné klasickými přístupy detekce.



Obrázek 2: Princip detekce a prevence plagiátorství v prostředí APE

### 3.3 Doplňování textu

Poslední z alternativních přístupů, který zde představíme, neprobíhá přímo při vzniku dokumentů, ale až po jejich dokončení či odevzdání. V tomto případě jde o nástroj pro práci s volným textem. Procesu se tu účastní přímo sám autor dokumentu a je testováno, jestli je jeho skutečným autorem.

Jde o komerční program Glatt Plagiarism Screening Program<sup>24</sup> [Glatt1999]. Jeho historie sahá údajně (dle informací na jeho webu) až do roku 1988<sup>25</sup>. Nástroj pracuje s předpokladem, že autor zná svůj styl, kterým píše a naopak nezná styl případného zdroje svého plagiátorství. Podle dostupných informací program pracuje tak, že odstraní z textu každé páté slovo<sup>26</sup> a nahradí ho mezery o standardní velikosti<sup>27</sup>. Je poté na studentovi, aby chybějící slova doplnil. Sledován je počet správně doplněných slov, chyby, čas a některé další blíže nespecifikované údaje. Na jejich základě je pak vypočítáno skóre udávající pravděpodobnost, že došlo k plagiátorství.

Nevýhodou takového systému je bezesporu nutnost přímé účasti studenta. Z tohoto důvodu také nelze počítat s tím, že by se takovéto testování dělo vzdáleně. Aby bylo účinné, je potřeba zajistit, aby student neměl během vyplňování či těsně před ním možnost sledovat kopii svého dokumentu (atž již vytisknou nebo v elektronické podobě). Lze si představit jeho aplikaci například tam, kde student odevzdává písemnou práci a dále je hodnocen na základě ústní části zkoušky. V tom případě by mohl přímo v kanceláři před zkoušejícím vyplnit podobný test.

24 Cena licence se pohybuje okolo 300 dolarů. V případě zakoupení licence i na druhý produkt společnosti je poskytována sleva na 500 dolarů za oba.

25 O poměrně dlouhé tradici této firmy v oblasti odhalování plagiátů svědčí i to, že vlastní poměrně lukrativní doménové jméno plagiarism.com, kterou si zaregistrovala již koncem února 1996. Nutno podotknout, že dle archivu webových stránek se tam umístěná prezentace od té doby prakticky nezměnila.

26 Zdá se, že toto nastavení je fixní a neměnné, čímž se mírně snižuje bezpečnost takového systému.

27 Zřejmě aby délka mezery nenapovídala, které slovo bylo odstraněno.

Z dnešního pohledu se v tomto případě jedná o poměrně zastaralý systém automatické detekce plagiátů. Organizační nároky odpovídají spíše době jeho vzniku, než současnému prostředí. Přesto však může být jít o poměrně účinné a relativně levné řešení, tam kde nejsou vysoké nároky na kapacitu a organizačně je zvládnutelný potřebný dohled. Alternativně lze také uvažovat o jeho využití v kombinaci s klasickými detektory pro prověření podezřelých případů.

### **3.4 Shrnutí**

Ukázali jsme si tři alternativní přístupy k boji s plagiátorstvím v akademickém prostředí. Dva z nich se zaměřují přímo na samotný proces vzniku dokumentů. Jednalo se v obou případech o nápady a postupy, které v nedávné době vymysleli a implementovali vyučující programování. I v současnosti se tedy potvrzuje, že právě tato skupina lidí přináší do boje s plagiátorstvím značnou část inovací, stejně jako tomu bylo i v historii (vzpomeňme vůbec první postupy pro odhalování plagiátů v sedmdesátých letech dvacátého století). Zatímco první způsob spoléhá zejména na své utavení, druhý způsob tvrdě vyžaduje podřízení se jasným pravidlům. Díky nim se zdá poměrně soběstačný a obejít jej je možné snad jen ručním opisováním kódu<sup>28</sup>. Přesto se jeví jako vhodné, oba případně kombinovat s klasickými detektory plagiátů, o kterých pojednávají následující kapitoly.

---

<sup>28</sup> Pomineme-li možnost prolomení šifrovaných metadat umožňující jejich libovolnou změnu.

## 4 Metriky používané v nástrojích pro detekci

*Measure what is measurable, and make measurable what is not so.*

*Galileo Galilei, italský fyzik a astronom (1564–1642)*

*You can't control what you can't measure.*

*Tom DeMarco, odborník na vývoj software a jeho řízení (1944)*

Druhá kapitola rozebrala typologii nástrojů zejména z pohledu uživatelského. Než přistoupíme k podrobnějšímu zkoumání vnitřního fungování některých typů nástrojů, považujeme za vhodné na chvíli se zastavit u jakéhosi mezistupně. Tím jsou právě metriky. Ty představují propojení mezi technologickými útrobami každého nástroje a jeho výstupem pro uživatele. I z uživatelského hlediska je proto vhodné vědět, jak se tato čísla počítají a co mohou v praxi znamenat.

Pokud chceme hovořit o metrikách v souvislosti s nástroji pro detekci plagiátů, musíme striktně rozlišovat dva přístupy. Ten první, zjevnější, představuje metriku jako důležitou součást konkrétního nástroje. V tomto pojetí slouží metrika přímo k vyhledávání plagiátů a vyjadřuje úroveň výskytu podezřelých znaků v dokumentech, případně přímo podobnost testovaných dokumentů. Právě takovému pojetí metrik se budeme obecně věnovat v této kapitole.

Druhý přístup zahrnuje metriky, které umožňují do určité míry vzájemně porovnávat různé nástroje pro detekci. Takové metriky neměří přímo podobnost dokumentů a samotné nástroje s nimi většinou nepracují. Jejich použitím ale můžeme porovnávat kvality různých nástrojů, případně různých nastavení téhož nástroje. Tento lehce abstraktnější přístup probereme až v kapitole 7.

Lancaster a Culwin ve své již zmiňované klasifikaci nástrojů zavádějí také klasifikaci podle použitých metrik. Proto také navrhují jakýsi nástin obecné klasifikace metrik<sup>29</sup>[Lancaster2005]. Pracují přitom se dvěma hledisky.

Prvním z nich je „rozměr“. Předpokládají, že metrika je definována nad množinou dokumentů a rozměr metriky odpovídá velikosti této množiny. Jinými slovy, rozměr metriky odpovídá tomu, s kolika dokumenty najednou metrika pracuje. Autoři tak rozlišují metriky singulární, párové a multidimenzionální. Specifickým případem je pak metrika korpální.

Singulární metriky, jsou ty, které pracují pouze nad jedním dokumentem. Při praktickém využití by mohlo jít o různé stylometrické ukazatele jako je počet a frekvence konkrétního slova, průměrná délka věty a podobně.

Protože nástroje často porovnávají dva dokumenty mezi sebou, patří mezi zřejmě nejčastěji používané metriky ty, označované jako párové. Pracují pochopitelně se dvěma dokumenty a často se používají k vyjádření míry jejich podobnosti. Může jít o délku nejdelšího společného řetězce, počet shodných slov, případně jejich poměr k délce dokumentu atd.

---

<sup>29</sup> Zabývají se přitom pouze metrikami „prvního typu“ tj. těmi, které jsou přímo součástí nástrojů a slouží k samotné detekci. To je v celku pochopitelné, neboť jejich cílem je v rámci klasifikace roztrádit nástroje podle používaných metrik, nikoliv je detailněji porovnávat.

Multidimenzionařní metriky jsou obecně takové, které měří vlastnosti více než dvou dokumentů zároveň. Praktickým využitím takového přístupu může být například vyhledávání shluků (clusterizace) podobných dokumentů, které umožní odhalit skupinky autorů, kteří od sebe navzájem opisovali (viz např. [Moussiades2005]). Zvláštním případem multidimenzionařní metriky je potom metrika korpální. Ta pracuje najednou s celou předloženou množinou dokumentů (k pojmu korpus viz kapitolu o typologii nástrojů, konkrétně část 2.2.3). Kromě srovnání různých korpusů je možné uvažovat o využití například ke zpřesnění detekce tím, že se automatizovaně určí společné části většiny dokumentů korpusu (například stejný úvod, hlavička zdrojového kódu, ...), které dále již nejsou považovány za podezřelý znak, pokud se při srovnání vyskytují v obou porovnávaných dokumentech.

V závěru autoři zobecňují svou klasifikaci podle rozměru tak, že singulární metriky mají rozměr jedna, tj. jsou jednodimenzionařní, párové metriky mají rozměr dva. Korpální metriky mají potom rozměr  $m$ , kde  $m$  je velikost korpusu.

Druhé hledisko, kterým se oba autoři ve své klasifikaci zabývají, je komplexita metriky. Podle něj rozlišují metriky na strukturální a povrchové (superficial). Jako strukturální metriky označují takové, pro jejichž určení je třeba znát strukturu zpracovávaného dokumentu. Z našeho pohledu se toto rozdělení jeví sice jako možné, ale velmi nepřesné. Ostatně samotní autoři dále uznávají, že je prakticky nemožné v některých případech jasně rozhodnout, o kterou metriku by se jednalo<sup>30</sup>. Tato diskuse se velmi nápadně podobá kritice tradiční typologie nástrojů (viz kapitolu 2.1) a sporům ohledně toho, co je ještě attribute counting a co již structure-metric přístup. Proto považujeme takovéto kritérium za podobně nevhodné, jako právě tuto historickou typologii. Pokud jde o klasifikaci podle metrik, z uživatelského hlediska by bylo užitečnější klasifikovat nástroje podle použití symetrické nebo asymetrické metriky respektive podle toho, jak nástroj pracuje s dokumenty různých délky (podrobně k této problematice viz dále).

V další části této kapitoly se podrobněji zaměříme na základní a nejpoužívanější párové metriky.

## 4.1 Úvod k části o metrikách v rámci nástrojů

Většina nástrojů<sup>31</sup> pro detekci plagiátů pracuje na bázi porovnávání dokumentů mezi sebou a hledání shody. V mnohých detailech se od sebe mohou jednotlivé přístupy odlišovat, například v tom, jak dokumenty předzpracovat, jestli pracovat s celými dokumenty nebo jenom s jejich nějakou zjednodušenou reprezentací a také v tom, co považovat za základní jednotku porovnání.

Ať již je jako základní jednotka zvoleno slovo, řádek<sup>32</sup>, věta, odstavec nebo třeba překrývající se triplety<sup>33</sup>, výsledkem je nakonec porovnání dvou dokumentů a číslo, které vyjadřuje jejich podobnost. V případě detekce plagiátů pak tuto podobnost dokumentů, či přesněji toto číslo vypočítané konkrétním algoritmem s danými parametry, interpretujeme jako míru podezřelosti, že jeden dokument je plagiátem druhého<sup>34</sup>. Toto zásadní číslo je (v případě porovnávání vždy dvou dokumentů mezi sebou) párovou metrikou, jak byla definována výše. Právě pro jejich časté použití si zde

30 Uváděn je příklad povrchové metriky ovšem počítané na předem upraveném dokumentu. Taková úprava (tokenizace, odstranění často užívaných bezvýznamových slov, nahrazení synonymy, ba i převod netisknutelných znaků na jiné) by se již dala považovat za úpravu se znalostí struktury dokumentu.

31 Samozřejmě se toto netýká intrinsic nástrojů a těch, které vůbec nepracují s obsahem dokumentů.

32 například ve zdrojovém kódu

33 trojice slov v pořadí z původního dokumentu, které se překrývají a vznikají, postupným posouváním se po dokumentu; podrobněji viz kapitolu 5.4.

34 Případně, že jsou oba plagiátem nějakého třetího.

ukážeme a okomentujeme obecně několik základních párových metrik. Budeme přitom postupovat od metrik symetrických, tedy takových, které dívají pro oba směry porovnání, respektive pro oba porovnávané dokumenty stejnou hodnotu, přes metriky asymetrické, které umožňují každý z dokumentů ohodnotit jinak, až k symetrizovaným asymetrickým metrikám, které spojují přednosti jedné hodnoty s výhodami asymetrických metrik. Podíváme se přitom na jejich základní vlastnosti, které jsou určující pro použití v nástrojích pro detekci plagiátů.

Protože, jak jsme již upozorňovali, různé nástroje používají různou základní jednotku porovnání a různé techniky předzpracování, budeme prozatím pro účely seznámení se základními metrikami pracovat s jakýmsi obecným nástrojem.

Pro vlastní potřebu si nadefinujeme  $V(X)$  jako množinu základních jednotek dokumentu  $X$ . Funkci  $V()$  můžeme v tomto případě chápat jako obecný preprocesor, který podle předem daných (a pro nás v tuto chvíli nepodstatných) pravidel zpracuje dokument  $X$  a vrátí množinu jeho základních jednotek. S těmito množinami pak budeme pracovat pomocí standardních množinových operací sjednocení a průnik. Podobně použijeme standardní značení pro počet prvků množiny.

## 4.2 Symetrická metrika podobnosti

Základní párová metrika při porovnávání dvou dokumentů je v literatuře nejčastěji nazývána podobnost (resemblance) a je aplikací Jaccardova koeficientu<sup>35</sup>. Výše popsaným značením ji lze zapsat následujícím způsobem.

$$res(A, B) = \frac{|V(A) \cap V(B)|}{|V(A) \cup V(B)|}$$

Takovouto metriku využívá například nástroje Ferret ([Lyon2004]) či PDetect ([Moussiades2005]), pracuje s ní ([Malkin2005]) a velmi detailně se jí věnuje rovněž ([Broder1997]). K základním vlastnostem této metriky patří to, že  $res(A, B) = res(B, A)$ , čili je symetrická. Její hodnota je číslo mezi nulou a jedničkou. Nula je to v případě, kdy dokumenty nesdílí žádnou společnou část (přesněji základní jednotku), jedna potom pokud oba dokumenty obsahují pouze tytéž jednotky. Samozřejmě tedy  $res(A, A) = 1$ . V ostatních případech je konkrétní velikost závislá na velikosti průniku (shody obou dokumentů), ale také na velikosti obou dokumentů.

Ukažme si to na příkladu. Po zpracování funkcí  $V()$  přistupujeme k dokumentům  $A$  a  $B$  jako k množinám základních jednotek. Ty jsme zde označili pro přehlednost malými písmeny. Funkce  $V()$  mohla například v dokumentech každému unikátnímu slovu přiřadit písmeno<sup>36</sup>. Stejně tak ale mohou písmena v závislosti na konkrétním zpracování (reprezentovaném zde obecnou funkcí  $V()$ ) odpovídat třeba větám, odstavcům nebo jiným složitějším jednotkám. Na principu výpočtu metriky se přitom prakticky nic nezmění. Dokumenty tedy můžeme chápat jako následující množiny. Prozatím pro zjednodušení předpokládáme, že v jednom dokumentu není obsažena konkrétní základní jednotka vícekrát<sup>37</sup>.

35 Jaccardův koeficient podobnosti je jedna ze standardně používaných měr podobnosti dvou souborů znaků ve statistice.

36 Toto je zjednodušený názorný příklad. Samozřejmě by v běžném dokumentu jednotlivá písmena jako identifikátor jedinečných slov početně nestačila. V praxi se pro to nejčastěji užívá hashovacích funkcí aplikovaných na obsah základní jednotky.

37 V případě slov jako základních jednotek to není příliš reálný předpoklad, v případě vět nebo umělých uskupení slov

$$V(A) = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n\}$$

$$V(B) = \{x, b, c, d, y, j\}$$

Jejich průnik a sjednocení jsou potom následující.

$$V(A) \cap V(B) = \{b, c, d, j\}$$

$$V(A) \cup V(B) = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, x, y\}$$

Odpovídající číselná vyjádření velikostí množin jsou pak samozřejmě takováto.

$$|V(A)| = 14$$

$$|V(B)| = 6$$

$$|V(A) \cap V(B)| = 4$$

$$|V(A) \cup V(B)| = 16$$

Z toho je zřejmé, že pro náš příklad je míra podobnosti dokumentů 25 %.

$$res(A, B) = \frac{|V(A) \cap V(B)|}{|V(A) \cup V(B)|} = \frac{4}{16} = 0,25$$

Protože jde o symetrickou podobnost, je stejná pro oba dokumenty. Tedy také  $res(B, A) = 0,25$ .

### 4.3 Druhá varianta podobnosti

V literatuře se objevuje také jiná varianta podobnosti (např. Monostori ji v [Monostori2002] uvádí jako „symetrickou podobnost“<sup>38</sup>). Ta se od té první liší ve jmenovateli. Namísto počtu prvků sjednocení je tam použito součtu počtu prvků obou množin<sup>39</sup>. Snad je to dáné tím, že z hlediska implementace je o něco málo snazší sečít velikosti dvou dokumentů, než určit velikost jejich sjednocení<sup>40</sup>.

$$res_2(A, B) = \frac{|V(A) \cap V(B)|}{|V(A)| + |V(B)|}$$

---

(např. tripletů) již více. Tento předpoklad můžeme obejít tak, že si nadefinujeme funkci  $V()$  tak, že opakování výskyty základních jednotek v tomtéž dokumentu bude ignorovat. Tím ovšem při převodu obsahu dokumentu na množinu základních jednotek ztrátíme jistou část informace. Pokud pracujeme s takovým typem metriky a neuchováváme informaci o opakování výskytech v rámci jednoho dokumentu, musíme o to pečlivěji volit základní jednotku porovnání. Alternativně je možné postupovat podle doporučení Brodera ([Broder1997]) a odlišit jednotlivé výskyty (například očíslováním) v obou dokumentech tak, že x-tý výskyt v prvním dokumentu se bude při porovnání shodovat pouze s x-tým výskytem v druhém dokumentu.

38 Přestože v jiných svých pracích již tuto metriku neuvádí a používá standardní „resemblance“ a „containment“. My se však této metriky podržíme pro zajímavý vztah jejího dvojnásobku k základním asymetrickým metrikám (viz dále).

39 Nelze pouze nahradit znak sjednocení znakem plus, protože součet množin v tomto 'primitivním' smyslu není definován.

40 Své opodstatnění potom zřejmě má pokud opustíme výše zmínovaný předpoklad, že v dokumentu (nebo jeho množinové reprezentaci) se žádná jednotka nevyskytuje vícekrát. Otázkou potom je, jak bude implementován průnik takových opakujících se jednotek.

Lze si představit situaci, kdy stejná jednotka se v jednom z porovnávaných dokumentů objevuje x-krát a ve druhém y-krát. Jeví se jako rozumné započítat ji do „průniku“  $\min(x, y)$  krát. V tomto případě jde o symetrickou metriku. Pokud bychom počítali metriku asymetrickou – pro každý dokument jinou, mohlo by (ale nemuselo) být žádoucí, započítat ji v případě shody pro první dokument x-krát a pro druhý y-krát.

Přestože se jedná opět o symetrický vzorec, vlastnosti této metriky jsou poněkud jiné, než u první verze podobnosti. Předně její hodnota nepřekročí nikdy 0,5. I v případě dvou totožných dokumentů bude totiž jejich průnik mít velikost každého z nich, kdežto součet jejich délek bude dvojnásobný. Čili  $res_2(A, A)=0,5$ . Pro normování na interval od nuly do jedné je tedy potřeba vynásobit ji dvěma.

Pokud bychom zachovali původní předpoklad unikátnosti prvků, liší se jmenovatel číselně od jmenovatele prvního vzorce podobnosti právě o velikost průniku (čili čitatele). Prvky průniku jsou v tomto případě ve jmenovateli obsaženy dvakrát (jednou v délce každého dokumentu).

Když navážeme na náš předchozí příklad, můžeme znova určit míru podobnosti našich dokumentů A a B, tentokrát pomocí této upravené metriky.

$$res_2(A, B) = \frac{|V(A) \cap V(B)|}{|V(A)| + |V(B)|} = \frac{4}{14+6} = \frac{4}{20} = 0,20$$

Pokud bychom hodnotu normovali vynásobením dvěma, vypovídalo by to o 40% podobnosti mezi dokumenty.

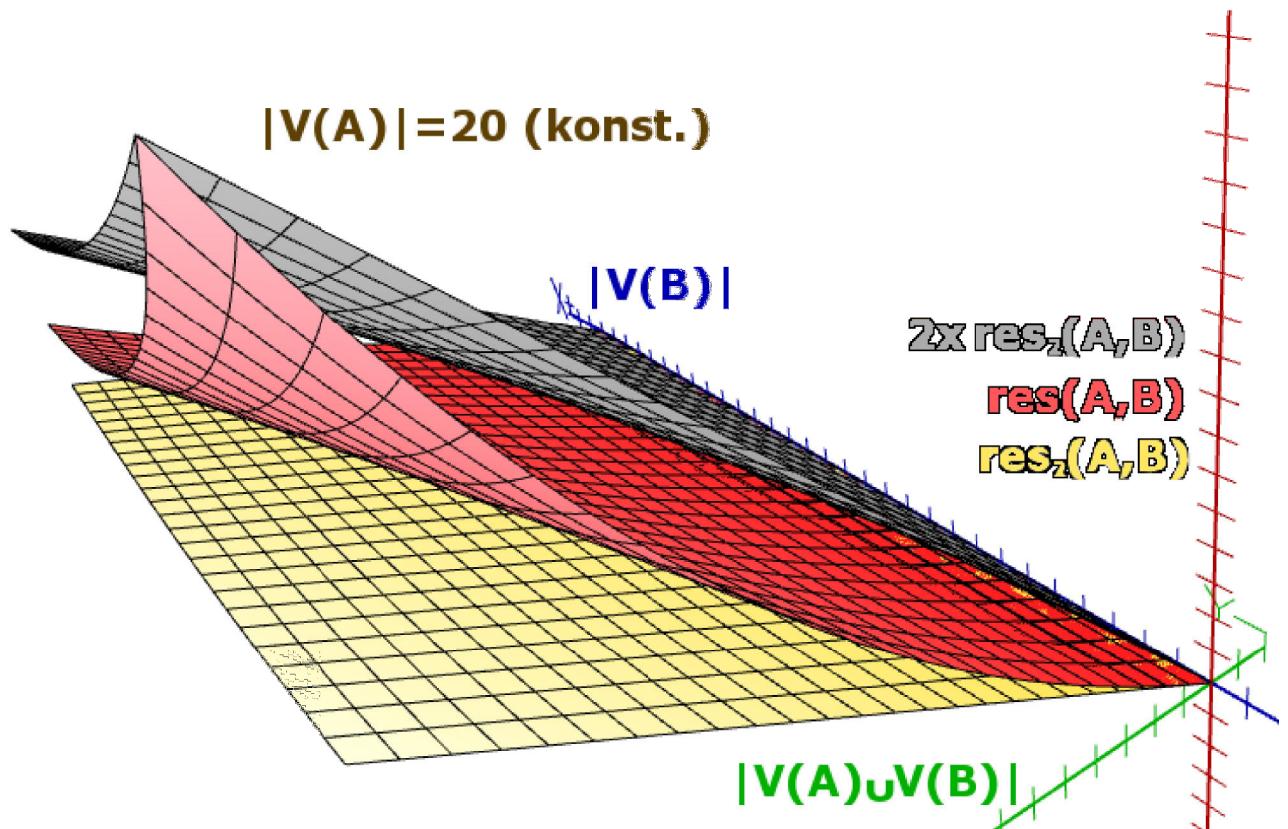
To, že je tato varianta méně selektivní je zřejmé i z obrázku 3. Ten znázorňuje průběh obou variant podobnosti v závislosti na velikosti průniku dvou dokumentů a velikosti jednoho z dokumentů při konstantní velikosti druhého. Předpokládáme, že dokument A má konstantní počet dvaceti jednotek porovnání. Osa Y (zelená) představuje velikost průniku (tj. počet jednotek obsažených v obou dokumentech) od nuly až do dvaceti<sup>41</sup>. Na ose X (modrá) je vynesena velikost dokumentu B (od dvaceti do padesáti jednotek – v průsečíku os je hodnota 20). Osa Z (červená) představuje hodnotu podobnosti (ta má pro přehlednost dvacetkrát zvětšené měřítko).

Zobrazeny jsou plochy, které odpovídají průběhu podobnosti  $res(A, B)$ ,  $res_2(A, B)$  a pro lepší představu i normované  $2 \times res_2(A, B)$ . Je zřejmé, že upravená podobnost v původním i normovaném tvaru je výrazně plošší než klasická podobnost založená na Jaccardově koeficientu. Ta má v okolí bodu (20,20) výrazný vrchol. Jeví se tedy jako o něco vhodnější pro lepší odlišení velmi podobných dokumentů, které jsou podezřelé z toho, že by mohly být plagiáty.

Symetričnost obou těchto metrik s sebou ovšem přináší také značnou nevýhodu. Všimněme si ještě na obrázku 3 levého horního okraje tří ploch. Pokud zafixujeme velikost průniku na maximu (v našem případě na dvaceti jednotkách), a pohybujeme se pouze po ose X (tj. zvětšujeme velikost dokumentu B), hodnoty metrik poměrně razantně klesají. Je to způsobeno růstem hodnoty jmenovatele (at' už velikostí sjednocení pro  $res(A, B)$  nebo součtu velikostí pro  $res_2(A, B)$ ) tím, jak narůstá velikost dokumentu B. To by bylo naprostě v pořádku, pokud by nás zajímala pouze pravděpodobnost, že dokument B je plagiátem dokumentu A. S tím jak v B roste počet slov či vět, které neobsahuje dokument A, je jistě toto podezření menší.

---

41 Fixujeme-li dokument A na délku dvaceti jednotek, nemůže být samozřejmě průnik dokumentů větší.



Obrázek 3: Porovnání průběhu funkcí symetrických metrik

Nezapomínejme ale, že díky symetričnosti nám klesá také podezření z toho, že dokument A je plagiátem dokumentu B. Dokument A má v našem případě konstantní velikost dvacet jednotek. Tím, že se nyní pohybujeme po levé spodní hraně plochy, čili jsme zafixovali velikost průniku také na dvacet jednotkách, jsme vlastně řekli, že dokument A je celý obsažen v dokumentu B. Podezření, že je plagiátem by tedy mělo být velmi vysoké. Hodnota metrik ale klesá s délkou dokumentu B<sup>42</sup>, a pokud tento bude dostatečně dlouhý<sup>43</sup>, hodnota klesne pod výstražnou úroveň<sup>44</sup>. Méně tímto neduhem trpí plošší a méně selektivní  $2 \times \text{res}_z(A, B)$ , která pro průnik 20 a velikost dokumentu B 50 jednotek dává něco kolem 0,57. Oproti tomu  $\text{res}(A, B)$  při těchto poměrech již pouhých<sup>45</sup> 0,40.

Při použití symetrické metriky podobnosti na nestejně velké dokumenty tak může docházet k oponutím (false negatives).

42 Délkou dokumentu v této kapitole rozumíme počet základních jednotek porovnání reprezentace dokumentu. Její konkrétní velikost závisí jak na skutečné délce dokumentu (např. počtu slov), tak na zvoleném způsobu porovnání (co skutečně s čím porovnáváme). Detailněji k této problematice viz kapitolu 5.2.

43 respektive 'dostatečně delší' než dokument A.

44 úroveň podobnosti nastavenou v detektoru, která odlišuje potenciální plagiáty, které vyžadují další pozornost a prozkoumání od dokumentů, které jsou si podobné velmi málo, a pravděpodobně nejdé o plagiáty

45 Čtyřiceti procentní podobnost je v praxi jistě pořád velmi vysoká a většina nástrojů pracujících s touto metrikou by jistě uživatele upozornila na podezřelý pár dokumentů. Nezapomínejme ovšem, že v příkladu máme poměrně nevelký rozdíl v délce dokumentů (20 a 50 jednotek). V praxi se mohou vyskytovat rozdíly až řádové například třístránkový dokument může být plagiátem vytvořeným z části kapitoly třicetistránkové bakalářské práce. Potom by hodnota symetrické podobnosti byla (již opravdu pouhých) řádově deset procent.

#### 4.4 Asymetrická metrika obsahu

Výše popsanou nevýhodou pro nestejnou délku dokumentů netrpí některé asymetrické metriky. Ty vychází z toho, že při porovnání dvou dokumentů nás nezajímá pouze jejich 'průměrná' podobnost, ale kvantifikují to, jak moc je některý dokument obsažen v jiném. Odtud i český název, který jsme pro základní asymetrickou metriku zvolili *obsah* (původní anglický termín je *containment*). Tuto metriku zmiňují například Broder [Broder1997] a Monostori [Monostori2002].

$$con(A, B) = \frac{|V(A) \cap V(B)|}{|V(A)|}$$

Hodnota této metriky závisí pouze na velikosti průniku dokumentů a délce jednoho z nich. Při rozdílné délce dokumentů se tak liší hodnoty  $con(A, B)$  a  $con(B, A)$ . Obsah je opět číslo mezi jedničkou a nulou. Hodnoty jedna nabývá v případě, že dokument A je podmnožinou dokumentu B, je v něm celý obsažen. Nuly nabývá tehdy, jestliže dokumenty nemají žádný společný obsah.

Opět můžeme pokročit v našem příkladu se zvolenými množinami.

$$con(A, B) = \frac{|V(A) \cap V(B)|}{|V(A)|} = \frac{4}{14} = 0,2857$$

Můžeme tedy říci, že dokument A je v dokumentu B obsažen z přibližně 29 %, respektive 29 % obsahu dokumentu A se nachází také v dokumentu B. Stejně tak můžeme určit i druhou hodnotu – obsah A v B.

$$con(B, A) = \frac{|V(B) \cap V(A)|}{|V(B)|} = \frac{4}{6} = 0,6667$$

Dvě třetiny dokumentu B jsou tedy obsaženy v dokumentu A. V našem příkladu jsou to konkrétně čtyři písmena ze šesti. Tato interpretace tak plně odpovídá i intuitivnímu chápání toho, co znamená, že jeden dokument obsahuje část druhého.

Známe-li obsah jednoho dokumentu ve druhém a velikost obou dokumentů<sup>46</sup>, můžeme snadno určit obsah druhého dokumentu v prvním.

$$con(B, A) = con(A, B) \times \frac{|V(A)|}{|V(B)|}$$

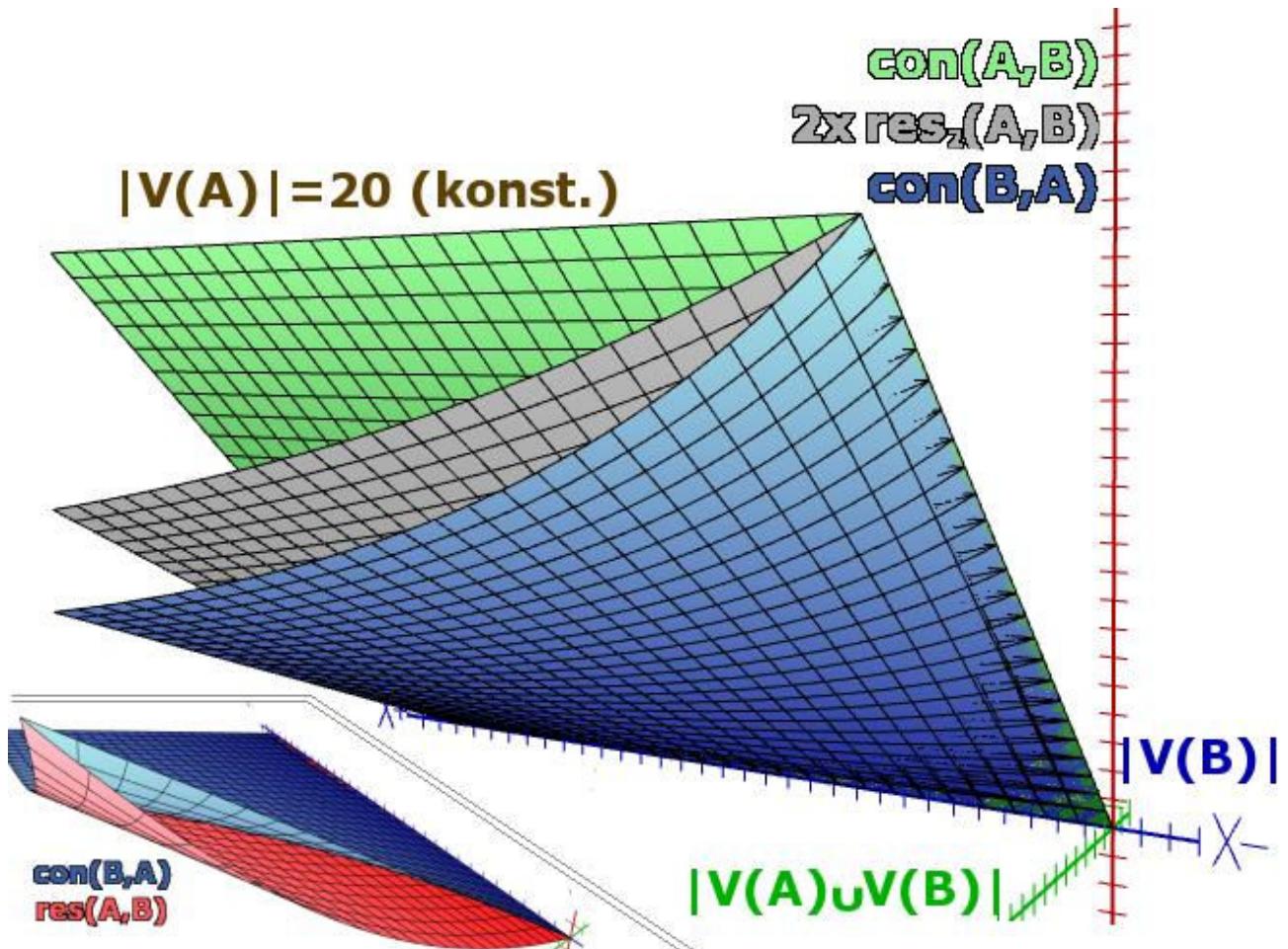
$$\text{V našem případě tedy } con(B, A) = \frac{4}{14} \times \frac{14}{6} = \frac{4}{6} = 0,6667.$$

Hlavní výhodou použití tohoto typu metriky je již zmiňovaná nezávislost na délce dokumentu. Můžeme nyní rozšířit situaci znázorněnou na obrázku 3 na straně 35 o dvě nové asymetrické metriky obsahu. Výsledný graf je zobrazen na obrázku 4. Pro lepší přehlednost již není zobrazena (původně žlutá)  $res_2(A, B)$ . Metrika podobnosti  $res(A, B)$  (červená) je znázorněna pouze vlevo dole na malém výřezu. Obě se nacházejí pod modrou plochou funkce  $con(B, A)$  a o jejich umístění si

---

<sup>46</sup> Teoreticky stačí pouze znalost poměru obou velikostí, ale v praxi je většinou snazší určit konkrétní délku obou dokumentů než jejich poměr.

lze udělat dobrou představu srovnáním s polohou (šedivé) funkce  $2 \times res_2(A, B)$ , která je shodná jako na obrázku 3 případně pomocí zmiňovaného výřezu.



Obrázek 4: Porovnání průběhu funkcí symetrických a asymetrických metrik.

Za zmínu stojí také oblasti, kde naopak obě vykazují hodnoty stejné. Kromě triviálního případu, kdy jsou porovnávány dva naprostě shodné dokumenty (tj. hodnota obou metrik bude jedna), se obě funkce shodují, při libovolné velikosti průniku, také pokud jsou porovnávány dokumenty stejné délky<sup>47</sup>. V tom případě si budou rovny samozřejmě také  $con(A, B)$  a  $con(B, A)$  neboť poměr velikostí dokumentů (viz vzorec výše) bude roven jedné, respektive jmenovatel v obou asymetrických metrikách bude totožný. Pokud budeme porovnávat dva dokumenty tytéž délky, budou hodnoty obou asymetrických metrik rovny symetrické metrice  $2 \times res_2(A, B)$ .

To je vidět i na obrázku 4, kdy v počátku osy X (velikosti dokumentu B) je hodnota 20 tedy stejná jako velikost dokumentu A. Pro  $x=20$  se pro libovolnou velikost průniku (od 0 společných jednotek až do 20) protínají  $con(A, B)$ ,  $con(B, A)$  a  $2 \times res_2(A, B)$  v jednom bodě.

<sup>47</sup> Samozřejmě sem spadá i výše zmiňovaný případ dvou totožných dokumentů. Pokud jsou totožné, mají stejnou délku.

## 4.5 Symerizované nesymetrické metriky

Jistou nevýhodou (i když spíše jde o logický důsledek) asymetrických metrik je, že výsledkem porovnání dvou dokumentů jsou dvě různá čísla. Přesto zřejmě tato nevýhoda (spočívající však spíše v nutnosti vypořádat se v návrhu nástroje s interpretací a prezentací obou čísel) vede k tomu, že se objevuje snaha tyto dvě čísla nějak dát dohromady a vytvořit tak opět symetrickou metriku, která by ale netrpěla výše popsanými neduhy (zejména velkou citlivostí na délku dokumentu).

Jednu z nich používají D. White a M. Joy ([White2004]). Vycházejí ze dvou výše zmínovaných asymetrických metrik obsahu a spojují je dohromady jako jejich průměr. Takto získané číslo bychom tedy mohli nazývat průměrným obsahem (average containment).

$$acon(A, B) = \frac{con(A, B) + con(B, A)}{2} = \frac{\frac{|V(A) \cap V(B)|}{|V(A)|} + \frac{|V(A) \cap V(B)|}{|V(B)|}}{2}$$

Průměr zlepší odolnost proti snižování hodnoty vlivem růstu délky jednoho z dokumentů, ale tento problém neodstraní zcela.

Pokračujme opět v příkladu, který nás provází od začátku této kapitoly a spočítajme si hodnotu pro naše dva dokumenty.

$$acon(A, B) = \frac{\frac{4}{14} + \frac{4}{6}}{2} = \frac{10}{21} = 0,4762$$

Jiné řešení preferují Malkin a Venkatesan ([Malkin2005]). Také vycházejí z jednoduchých asymetrických metrik, ale pracují s jejich maximem. Tak se skutečně zbaví oné hlavní nevýhody a mohou přitom pracovat s jedním číslem jako výsledkem porovnání. Oni svou metriku nazývají prostě  $S_3$ , my si ji dovolíme pojmenovat maximální obsah (maximal containment).

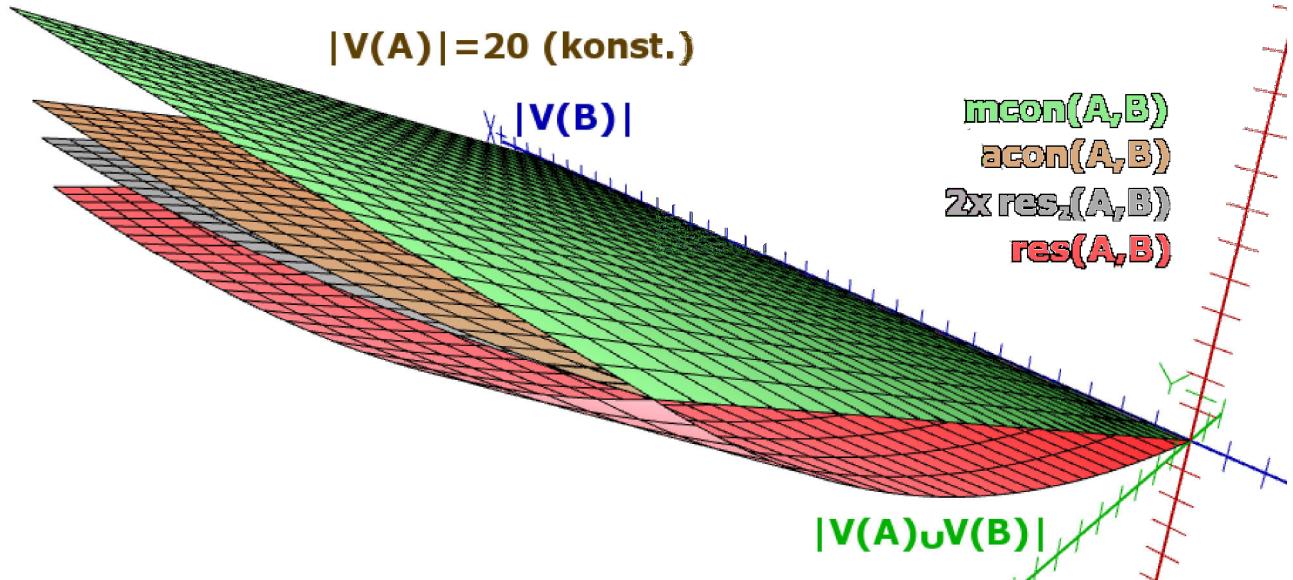
$$mcon(A, B) = \max(con(A, B), con(B, A)) = \max\left(\frac{|V(A) \cap V(B)|}{|V(A)|}, \frac{|V(A) \cap V(B)|}{|V(B)|}\right)$$

$$mcon(A, B) = \frac{|V(A) \cap V(B)|}{\min(|V(A)|, |V(B)|)}$$

Dokončíme náš příklad a spočítáme si hodnotu pro naše dva hypotetické dokumenty.

$$mcon(A, B) = \max\left(\frac{4}{14}, \frac{4}{6}\right) = \frac{4}{6} = 0,6667$$

Zobrazíme-li tyto metriky i do našeho grafu na obrázku 5,  $mcon(A, B)$  se bude pochopitelně v našem případě krýt s  $con(A, B)$ . U  $acon(A, B)$  je vidět oslabení ale ne úplné vymizení, vlivu velikosti druhého dokumentu. I u obou symetrizovaných metrik opět platí jejich rovnost dvojnásobku druhé varianty podobnosti (a tedy i jejich vzájemná rovnost) v případě stejné délky obou dokumentů.



Obrázek 5: Porovnání průběhu základních symetrických a symetrizovaných metrik

## 4.6 Shrnutí

Ukázali jsme si několik velmi jednoduchých základních metrik, které můžeme použít při vzájemném porovnání obsahu dvou dokumentů. Samozřejmě to ani zdaleka není výčet všech možností a toto nebylo ani cílem této části kapitoly. Metriky zdaleka nemusí být normalizované tak jako námi zmiňované a jejich hodnota může být prakticky libovolná.

Symetrické metriky mají jistou výhodu v relativní jednoduchosti jejich implementace a prezentace výsledků uživateli. Ve své základní podobě ale mohou trpět nežádoucí citlivostí na rozdílnou délku dokumentů. Proto tam, kde je pravděpodobné, že se budou porovnávat i dokumenty o růzově různé délce, je vhodnější použít metriky asymetrické, případně pokud je požadavek na symetričnost, metriky tvořené jako symetrizované asymetrické. Shrnutí základních poznatků je v tabulce 2.

Různá empirická vylepšení nebo volba jiného přístupu může vést k použití zcela jiných druhů metrik. Například Kang s kolegy ([Kang2006]) vyvinuli vlastní komplexní metriku, která je na první pohled velmi komplikovaná. Předkládáme ji zde čistě pro ilustraci a jako důkaz, že používané metriky nemusí být jen tak jednoduché, jaké jsme zde uvažovali.

$$sim(A, B) = |V(A)| \times (1/e^{\frac{|V(A)|}{|V(A) \cap V(B)| + a \times |synonym(A, B)|}})^{-1} + \sqrt{|V(A) - V(B)| + |V(B) - V(A)|}$$

Cílem bylo pouze ukázat, že již v takto jednoduchých příkladech je možné se setkat s několika různými numerickými vyjádřeními podobnosti dokumentů. Z toho je mimo jiné zřejmé, že při porovnávání různých nástrojů pro detekci plagiátů tak nelze přímo srovnávat hodnoty jimi naměřené pro jednotlivé páry.

V této kapitole jsme pracovali na úrovni základních jednotek a téměř vůbec jsme se nezabývali tím, jak a na základě čeho jsou z dokumentů získávány. Na tuto problematiku se blíže podíváme v následující kapitole.

Tabulka 2: Shrnutí vlastností základních metrik

<i><b>Český název</b></i>	<i><b>Anglický název</b></i>	<i><b>Symetričnost</b></i>	<i><b>Citlivost na rozdílnou délku dokumentů</b></i>	<i><b>Hodnota (A,B) pro náš příklad</b></i>
Podobnost	Resemblance	ano	velká	0,2500
Podobnost 2	Symmetric similarity	ano	střední	0,2000
Obsah	Containment	ne	žádná	0,6667
Průměrný obsah	Average similarity	ano	malá	0,4762
Maximální obsah	$S_3$	ano	žádná	0,6667

## 5 Detekce plagiátorství volného textu

*The average Ph.D. thesis is nothing but a transference of bones from one graveyard to another.*

*James Frank Dobie, americký folklorista a spisovatel (1888-1964)*

V této části se podrobněji podíváme na hlavní techniky detekce plagiátorství u volného textu v přirozeném jazyce. Navzdory tomu, že (jak jsme uváděli v kapitole 1.3) historicky první automatizované nástroje byly určeny k detekci plagiátů ve zdrojových kódech, my se zaměříme zejména na volný text. Právě na něj je totiž v současnosti, zdá se, soustředěna největší pozornost jak z hlediska výzkumu, tak z hlediska nástrojů a služeb dostupných na trhu. S rozvojem Internetu a zejména webu s mnoha miliardami digitálních dokumentů, spolu s násokem mladé generace v oblasti informační gramotnosti (nikoliv už etiky) je tak plagiátorství stále snazší a potenciální zdroje mnohem dostupnější.

Níže popisované postupy vychází zejména z otevřených zdrojů z akademické sféry. Komerční firmy si často informace o fungování svých nástrojů nechávají pro sebe. To je pochopitelné hned ze dvou důvodů. Prvním je obecná ochrana know-how, běžná ve všech oblastech podnikání. Druhým, pro tuto oblast specifickým důvodem je bezpečnost. Čím méně prozrazují o svých postupech pro hledání plagiátů, tím menší je pravděpodobnost, že se někdo naučí jejich detektory cíleně obcházet. Lze snad ale předpokládat, že komerční nástroje používají zhruba podobné techniky jako ty, jejichž principy jsou popsány v dostupné literatuře zejména z akademických kruhů<sup>48</sup>. Akademický svět má totiž na automatizovaném odhalování plagiátů tak říkajíc osobní zájem a poměrně dlouhou tradici.

Nejprve se zaměříme na obecné porovnávání obsahů dokumentů prováděné na základě jejich reprezentací. Tyto postupy umožňují relativně velmi rychlé zpracování velkého množství dokumentů a jsou v literatuře často popisovány a rozvíjeny. Můžeme je tedy považovat za jakési best practices této oblasti. Mají však i některé nevýhody a proto se v další části této kapitoly stručně podíváme také na přímé porovnání obsahů dokumentu, které je sice výrazně pomalejší, ale umožňuje některé operace, které při porovnání pouhých částečných reprezentací nejsou možné. V praxi se někdy úspěšně kombinují oba přístupy.

### 5.1 Specifika přirozeného jazyka

Dříve než si představíme konkrétněji některé postupy, zaměřme se na to, co je pro volný text v přirozeném jazyce specifické a s čím je tedy nutné při konstrukci takového detektoru počítat.

Předně to je velmi rozsáhlá slovní zásoba. Je nesmírně obtížné určit počet slov v daném jazyce. V [Oxford] se uvádí, že pouze ve slovníku Oxford English Dictionary je minimálně čtvrt milionu různých anglických slov<sup>49</sup>. V to ovšem nejsou počítány jejich různé tvary a slova náležící do technického jazyka a slova užívaná pouze v některých regionech. Jazyková poradna Ústavu pro jazyk český AV ČR odpovídá na stejný dotaz podobně [UJC]. Uvádí, že v doposud největším vydaném

48 I proto, že komerční nástroje nezřídka vycházejí z původně akademických projektů.

49 Slov jakožto skupin znaků, nikoliv významů. Zejména angličtina je známa svými mnoha někdy zcela rozdílnými významy stejně psaných slov. Pokud by byly započítány i ty, takových slov by dle tam zmíněných informací bylo minimálně třikrát více.

českém slovníku je zhruba 250 000 hesel. Jde však o slovník, který vycházel v letech 1935–1957. Novější méně obsáhlý slovník obsahuje kolem 192 000 hesel. V jazyce s bohatou flexí, jako je právě čeština se dá očekávat, že počet slov včetně jejich různých tvarů bude ještě řádově větší.

Kromě značného počtu slov je u volného textu třeba uvažovat také o jeho malé strukturovanosti. Nebereme-li v úvahu nadpisy, tabulky či popisky u některých formátů dokumentů<sup>50</sup>, můžeme text strukturovat na slova, věty a odstavce. Samotná struktura textu však nese pouze minimální informaci pro jeho porovnání – podstatný je hlavně obsah. Praktické aplikace umělé inteligence zatím nedospěly tak daleko, aby v dostatečné míře 'porozuměly' textu a srovnaly jeho obsah s obsahem mnoha jiných dokumentů v rozumném čase. Navíc přibližná shoda významu je obvyklá i tam, kde se o plagiátorství zcela jistě nejedná. Proto se detektory zaměřují zejména na vyhledávání úplných shod v kratších úsecích textu.

Dalším faktorem je relativní volnost jazykových pravidel. Jde o schopnost vyjádřit podobnou myšlenku různými slovy, případně s různou strukturou (rozdelením či spojením vět, vyněcháním části). Můžeme užívat synonyma a různé opisy. Specifickým případem je potom překlad do jiného národního jazyka. Samotná existence mnoha národních jazyků komplikuje univerzálnost detektorů.

## 5.2 Porovnání reprezentací dokumentu

Kromě přímého porovnání dokumentů (viz dále) je v současnosti zřejmě nejpoužívanější technikou převod obsahu dokumentu na jeho snadněji porovnatelnou reprezentaci. Tyto postupy popisují například [Brin1995], [Heinze1996], [Broder1997], [Finkel2002], [Malkin2005] a další. V některých aplikacích potom pro porovnání není použito celých obsahů dokumentu, ale pouze reprezentace pokrývající jeho malou část. To jednak dále urychluje porovnání a navíc zásadně snižuje prostorové požadavky na uložení dokumentů do archivu k dalšímu případnému pozdějšímu porovnání.

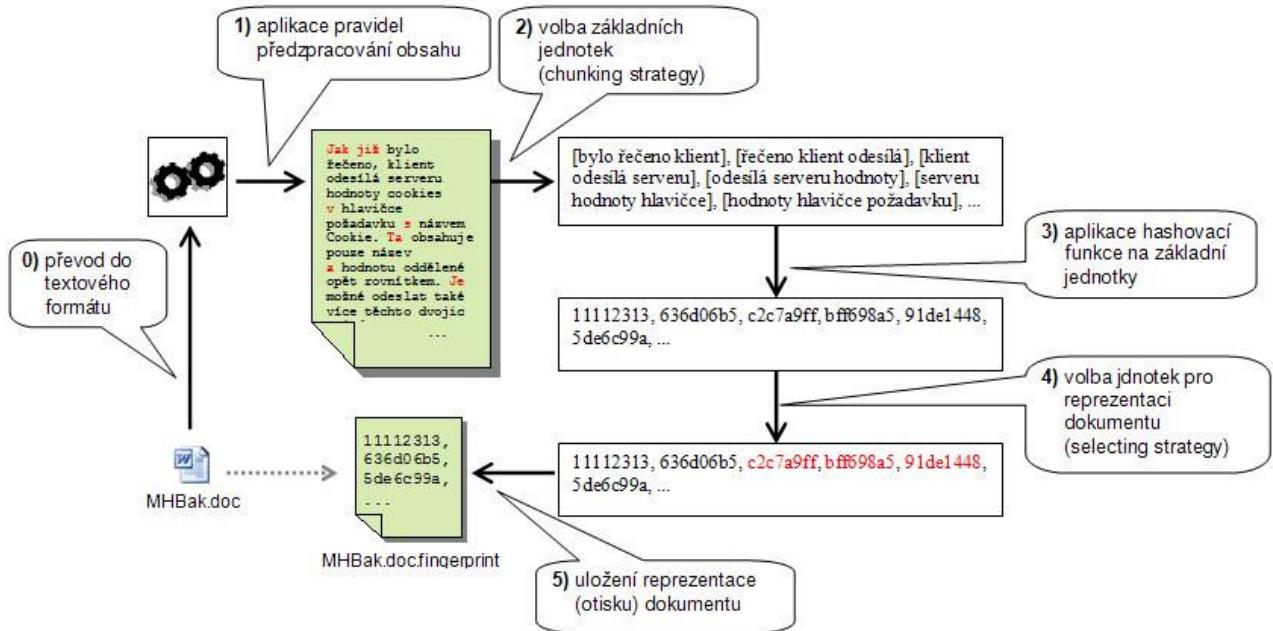
Projděme si nyní společně jednotlivé fáze tohoto procesu. Pro přehlednost je celý proces znázorněn schematicky na obrázku 6.

Dokumenty jsou nejprve převedeny z různých formátů, které zohledňují také formátování do formátu čistého textu. Následně mohou být na tento text aplikována některá speciální pravidla před jeho dalším zpracováním. Může jít například o nahrazení synonym, odstranění často používaných nebo příliš krátkých slov a číselných výrazů a podobně. Dále je dokument převeden na základní jednotky porovnání, které jsou dále pomocí hashovací funkce<sup>51</sup> převedeny na svou číselnou reprezentaci. Nakonec z nich mohou být vybrány ty, které budou celý dokument reprezentovat a teprve ty jsou následně porovnávány (viz také obrázek 7 na straně 45). Jednotlivé kroky si nyní podrobněji rozebereme v následujících kapitolách.

---

50 Většina z tohoto formátování je stejně současnými detektory nejprve odstraňována.

51 podrobněji viz kapitolu 5.5



Obrázek 6: Schema přípravy reprezentace dokumentu

### 5.3 Aplikace pravidel předzpracování

K tomu, dokumenty před samotným zpracováním nejprve podle nějakých pravidel upravit, existuje hned několik důvodů. Může jít o zvýšení úplnosti a přesnosti detekce (k témtoto pojmu detailněji viz kapitolu 7.1). Nezřídka však jdou ve výsledku proti sobě a je nutné dobře transformace zavážovat.

Při zvyšování úplnosti detekce jde o to, aby části textu, které jsou významově shodné, byly pokud možno shodné také fyzicky respektive svým vyjádřením v textovém dokumentu. Pokud pracujeme jako se základními prvky se slovy, je důležité, aby shodná slova byla prezentována shodně. Otázkou je již samotná hranice slova. Přirozenou hraničí slova v textu je mezera<sup>52</sup>. Kromě standardní mezery ale existují i jiné typografické znaky s podobnou funkcí (např. nedělitelná mezera, mezera o šířce m a n). Podobná situace nastává i v případě uvozovek, závorek a pomlček<sup>53</sup>. Je tak třeba mít na paměti i tyto znaky, které se mohou v dokumentu objevit. Interpunktční znaménka navíc nejsou od slova, za nímž následují, oddělena mezerou. Proto bývá někdy předem odstraňována interpunkce. Podobnou obecnou transformací je převod všech písmen na malá.

Další možnou transformací pro zvýšení úplnosti je odstranění všech číselných údajů. Takový přístup zřejmě vychází z praxe, kdy je část dokumentu plagiátorem převzata, ale číselné údaje jsou pozměněny. Alternativně lze toto využít také v případě, že původní dokument byl například ve formátu, který podporuje anotační aparát jako poznámky pod čarou nebo odkazy na bibliografické záznamy. Po jeho převodu do čistého textu nezřídka zůstanou číselné odkazy na ně součástí slov.

<sup>52</sup> Některé nástroje považují prostě za konec slova jakýkoliv nealfanumerický znak ASCII tabulky. Tím sice potenciální problémy se znaky řeší, ale potom zase například v českém textu lámou slova uprostřed i několikrát na znacích s diakritikou, které většinou (jako by to byla mezera) ze slova vypustí.

<sup>53</sup> respektive pomlčka, míinus, spojovník

Příbuzným problémem je také převod dokumentů na společnou kódovou stránku, to se však týká spíše převodu do textového formátu než samotného předzpracování. Přitom již předpokládáme, že fyzicky jsou dokumenty jednotné a zajímá nás zejména jejich obsah.

Z hlediska úplnosti ve vztahu ke smyslu obsahů dokumentů a nikoliv jejich forem se nabízí také převod synonym na jedno slovo za pomocí tezauru, případně převod různých tvarů slov na jednotný či jejich kombinace.

Kromě úplnosti je možné předzpracováním dosáhnout také zvýšení přesnosti<sup>54</sup>. V tomto případě nám jde zejména o snížení množství falešných poplachů (false positives). Těm lze v této fázi předcházet zejména odstraněním často se opakujících slov (stop-words) či frází, alternativně také například velmi krátkých slov.

Otázkou potom je, jak moc takovýto předzpracovaný dokument odpovídá tomu původnímu a jak je přehledný pro toho, kdo následně kontroluje podezřelé dvojice, na které nástroj upozornil. V případě rozsáhlých úprav (např. synonyma, jednotný tvar slov) lze uvažovat o tom, zde má být pro takové ruční porovnání uchován také nepředzpracovaný dokument.

## 5.4 Volba základních jednotek

Pro výkonné charakteristiky detekčního nástroje (zejména přesnost, úplnost a bezpečnost<sup>55</sup>) je při tomto typu porovnání zásadním parametrem, co si implementátor zvolí jako základní jednotku<sup>56</sup> porovnání. Je zřejmé, že pokud by základní jednotkou byl celý dokument (porovnávali bychom dokumenty jako celky), dosáhli bychom vysoké přesnosti – detektor by vracel pouze dokumenty zcela shodné, kdy alespoň jeden z nich je téměř jistě plagiát. Jakákoliv změna obsahu dokumentu by ale vedla k tomu, že by nebyl detekován. Dosahovali bychom mizivé úplnosti a bezpečnosti. Naopak pokud bude základní jednotkou písmeno, úplnost bude vysoká – poměrně jistě budou vráceny dokumenty, které jsou skutečnými plagiáty. Problém ale bude v tom, že budou takto označeny téměř všechny dokumenty a přesnost bude tedy velice nízká.

Cesta k uspokojivým hodnotám přesnosti i úplnosti je rozumná volba základní jednotky. Ta musí mít sama o sobě dostatečný myšlenkový obsah (význam), ale zároveň se nesmí obecně příliš často vyskytovat v dokumentech, které nejsou svými plagiáty či si nejsou jinak podobné. Různí autoři volí různé přístupy. Někteří pracují na úrovni obecných skupin jednotlivých znaků (či bytů<sup>57</sup>), jiní dávají přednost přirozeným skupinám znaků, nejčastěji slovům či větám.

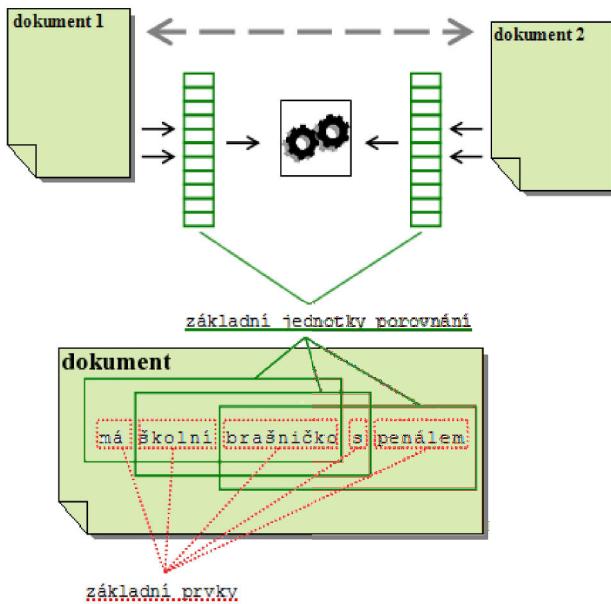
Jemnějším rozlišením než samotné základní jednotky jsou jejich základní prvky. Pokud jako základní jednotku zvolíme tři slova, potom zvoleným základním prvkem je slovo a délka jednotky je tři. Volba základní jednotky tak spočívá ve zvolení základních prvků, délky základní jednotky

54 ale nikoli nutně obojího zároveň

55 Přesnost je podíl skutečných plagiátů na všech dokumentech označených nástrojem jako plagiáty. Úplnost podíl odhalených skutečných plagiátů na všech skutečných. Bezpečnost je minimální počet změn v plagiátu nutných pro to, aby nebyl nástrojem odhalen. Detailněji k témtu ukazatelům viz kapitolu 7.1.

56 Zde užíváme termín jednotka (respektive základní jednotka, či základní jednotka porovnání) pro úseky textu, které se budou porovnávat. V literatuře jsou často nazývány chunks (česky cosi jako odřezek, řádný oddelený kus). Brin například užívá termínu unit (jednotka) pro části, ze kterých sestává chunk. Ty my nazýváme základními prvky. Filnkel hovoří ve stejném smyslu o tokenech a také z nich vytváří chunks. Broder nazývá své základní jednotky (v našem slova smyslu) shingle, což může znamenat šindel (možná inspirace překrývajícími se šindeli na střeše) ale také oblázek.

57 Například v případě, kdy je nástroj určen také k porovnávání libovolných binárních souborů.



Obrázek 7: Vztah mezi prvkem, jednotkou a dokumentem v kontextu porovnání reprezentací dokumentů

a způsobu její konstrukce. Vztah prvků, jednotky a dokumentu v kontextu porovnávání dokumentů je zobrazen na obrázku 7.

Obecně se autoři shodují, že samotné jedno slovo je jednotka poměrně malá s nedostatečně rozlišeným významem a relativně často se opakující. Minimem tak bývá skupina několika slov (používá většina nástrojů či popisuje většina prací). Další úrovní jsou potom celé věty (popisováno v [Brin1995], obdobně s větami pracuje i [White2004]<sup>58</sup>), lehce extrémním případem by byly odstavce.

Volba vhodné jednotky závisí také na účelu, ke kterému má být detektor používán. Pokud budou porovnávány převážně velmi dlouhé texty typu vědeckých prací, je možné uvažovat i o odstavci či několika větách jako relativně vhodných. Z toho zřejmě plynou i rozdílná doporučení autorů pro volbu základní jednotky. Jejich testy neprobíhaly za stejných podmínek a se stejným testovacím korpusem<sup>59</sup>.

Při dalším výkladu se přidržíme volby několika slov (kolika o tom dále). Popisované postupy jsou ale plně aplikovatelné i na ostatní případy<sup>60</sup>.

#### 5.4.1 Volba délky jednotky

Zvolit vhodný počet prvků, čili délku jednotky je také velmi důležité. Příliš krátké jednotky mají nízkou vypovídací hodnotu (podobně jako samostatná slova), příliš dlouhé jednotky mohou způsobit, že kvůli jedné drobné změně v nich nám uniknou případné plagiáty.

58 I když v nástroji Sherlock probíhá porovnání ne zcela v souladu s postupem, který zde uvádíme.

59 Pro testování se často používají bud' uměle připravené materiály, nebo ty, které ale s největší pravděpodobností nevznikly plagiátorstvím ale které má autor zrovna k dispozici.

60 Pokud si místo slov představíme kupříkladu chunks v Brinově pojed. Viz poznámku číslo 56.

Také zde platí, že se různí autoři neshodují. Doporučovaná délka závisí jak na tom, co je zvoleno za základ jednotky, tak ale také zřejmě na osobních zkušenostech s laděním nástrojů v praxi (nebo při laboratorních testech). Pro ukázku uvádíme v následující tabulce souhrn toho, jaké základní jednotky porovnání někteří autoři preferují a jaké volí jejich délky. Některé údaje v tabulce jsou převzaty z [Monostori2002].

Tabulka 3: Doporučené základní jednotky porovnání u různých autorů

<i>Autor</i>	<i>Základní jednotka porovnání</i>
Heinze	20 po sobě následujících souhlásek
Broder	7–10 po sobě následujících slov
Finkel	průměrně 10 po sobě následujících slov ( $V_{H10}$ viz dále)
Lyon	3 po sobě následující slova
Garcia-Molina	1 slovo, 5 slov, 10 slov, 1 věta

Některé přístupy dokonce umožňují mít v dokumentu jednotky rozdílné délky. Otázkou také je, mají-li se jednotlivé základní jednotky navzájem překrývat. Na jednotlivé případy se teď podíváme podrobněji. Následující příklad je převzat a upraven z [Brin1995]. Mějme následující dokument A o délce šesti slov.

*má školní brašničko s penálem dřevěným<sup>61</sup>.*

Přidržíme se přitom značení, které jsme zavedli v kapitole 4. Nyní nás tedy bude zajímat to, co jsme předtím přehlíželi a brali jako jakousi černou skříňku co dělá s dokumentem ona funkce  $V()$ . Nebudeme pro přehlednost předpokládat žádné předzpracování. Nejprve tedy, pro úplnost, jak by situace vypadala, kdybychom jako základní jednotku zvolili jedno slovo.

$$\begin{aligned} V_1(A) &= \{'má', 'školní', 'brašničko', 's', 'penálem', 'dřevěným'\} \\ |V_1(A)| &= 6 \end{aligned}$$

Pokud zvolíme délku tří slov a nebudeme chtít, aby se slova překrývala, bude situace následující.

$$\begin{aligned} V_3(A) &= \{'má školní brašničko', 's penálem dřevěným'\} \\ |V_3(A)| &= 2 \end{aligned}$$

Je zřejmé, že počet jednotek na dokument se snížil třikrát. Nároky na porovnání i uchování jsou tedy menší, ale tím, že se jednotky nepřekrývají, jsme velmi razantně snížili bezpečnost. Stačí totiž na začátek dokumentu přidat jediné další slovo a výsledná reprezentace takového dokumentu (na-

<sup>61</sup> Jde o úvod básně Školní brašnička z představení Dobytí severního pólu z repertoáru Divadla Jíry Cimrmana.

zvěme ho B) nemá s reprezentací toho původního žádnou společnou jednotku. V případě porovnání by tedy dokument nebyl označen jako plagiát.

*ty má školní brašničko s penálem dřevěným*

$$V_3(B) = \{ 'ty má školní', 'brašničko s penálem', 'dřevěným' \}$$

$$V_3(A) \cap V_3(B) = \emptyset$$

Bezpečnost v Brinově pojetí (detailněji k tomuto pojmu viz kapitolu 7.1.3) je tedy u takto pracujícího nástroje rovna jedné stací jediná změna a plagiát nebude odhalen. Řešením je použít překrývající se jednotky. Zbavíme se tak ale výhody jejich nižšího počtu. Pokud bychom ale chtěli v této podobě reprezentaci ukládat, nároky na prostor budou obrovské. Mohou být dokonce větší než na uložení původního dokumentu. Jejich délka (počet slov dokumentu a počet základních jednotek) bude odpovídat, ale průměrná velikost základní jednotky bude větší než průměrná velikost slova<sup>62</sup>.

$$V_{3a}(B) = \{ 'ty má školní', 'má školní brašničko', 'školní brašničko s', 'brašničko s penálem', 's penálem dřevěným' \}$$

$$V_{3a}(A) = \{ 'má školní brašničko', 'školní brašničko s', 'brašničko s penálem', 's penálem dřevěným' \}$$

$$|V_{3a}(A) \cap V_{3a}(B)| = 6$$

Právě tento způsob převodu je zřejmě nejčastější. Jeho vysoké nároky na prostor se řeší buď tak, že se používají pouze vybrané jednotky (viz dále) a nebo jde o intrakorpální nástroje, kde se reprezentace dokumentů neukládají a porovnání se provádí pouze v rámci korpusu, kde se nepředpokládají velmi vysoké počty dokumentů (řádově menší než by byla historická databáze uložených dokumentů).

Brin s kolegy popisuje ještě jeden způsob, který oni preferují (a přebírá ho od nich i [Finkel2002]). Ten může produkovat jednotky nestejné délky. Je založen na hodnotě výstupu hashovací funkce pro jednotlivá slova. Nejprve se vypočítá hash pro každé slovo až následně se na jejich základě vytvářejí ze slov základní jednotky<sup>63</sup>. Jednotky se nepřekrývají a zaberou tak méně místa. Nedochází tu však ke stejnemu efektu jako u varianty s nepřekrývajícími se trojicemi, protože konce jednotek nejsou dány pevně jejich pořadím, ale jejich obsahem. Konec jednotky totiž určuje hodnota hashe respektive její dělitelnost stanovenou konstantou x.

Předpokládejme, že hypotetická hashovací funkce H() přiřadí slovům z našich dokumentů A a B následující číselné hodnoty.

$$\begin{aligned} H('ty') &= 11 \\ H('má') &= 25 \\ H('školní') &= 33 \\ H('brašničko') &= 19 \\ H('s') &= 13 \\ H('penálem') &= 39 \\ H('dřevěným') &= 16 \end{aligned}$$

- 
- 62 Pokud bychom nepoužili hashovací funkci tak bude požadovaná velikost x-násobkem velikosti původního dokumentu, kde x je zvolená délka základní jednotky. I při jejím použití ale (v případě že požadujeme funkci s relativně nízkým počtem kolizí a tedy dostatečně velkým oborem hodnot) může velikost jejich výstupů spolu s režii překročit velikost potřebnou k uložení průměrného slova. Podle [NLP] je přitom průměrná délka slova v českém textu (tj. s opakováním) 5,5 písmen. Přitom kratší slova do pěti znaků jsou relativně častější.
- 63 Hash hodnota se pak pro takovou jednotku, pokud se skládá z více slov, samozřejmě přepočítá, aby ji reprezentovala jako celek.

Když zvolíme jako výběrový parametr číslo 3, říkáme tím, že jednotky budeme dělit za slovy, které mají hodnotu hash bezezbytku dělitelnou třemi. Ta jsou v našem příkladu dvě. Množiny základních jednotek tak tedy budou vypadat následovně.

$$\begin{aligned} V_{Hx}(A) &= \{'má školní', 'brašničko s penálem', 'dřevěný'\} \\ |V_{Hx}(A)| &= 3 \\ V_{Hx}(B) &= \{'ty má školní', 'brašničko s penálem', 'dřevěný'\} \\ |V_{Hx}(B)| &= 3 \\ |V_{Hx}(A) \cap V_{Hx}(B)| &= 2 \end{aligned}$$

Dosáhli jsme tedy úspory místa (základních jednotek je méně než základních prvků) a přitom je zachována lepší úroveň bezpečnosti. Jak upozorňuje [Finkel2002], vložení nebo odstranění části textu má vliv na změny základních jednotek pouze v bezprostředním okolí změny. Případný útočník by musel znát námi použitou hashovací funkci a použitý parametr modulo a následně do textu rozmíšťovat slova, s odpovídající hodnotou H() tak, aby narušil původní zakončení jednotek.

V tomto případě je délka jednotky proměnlivá. Očekávaná průměrná délka jednotky je rovna zvolenému parametru modulo<sup>64</sup>. Nejmenší jednotky budou obsahovat jedno slovo (pokud jsou dvě slova s hodnotou dělitelnou parametrem hned za sebou<sup>65</sup>). Mohlo by se ale stát i to<sup>66</sup>, že dokument nebude obsahovat žádné slovo s hodnotou dělitelnou parametrem a celý se tak stane jedinou jednotkou porovnání. K možným nevýhodám řadí Finkel to, že pokud je hash hodnota velmi často používaných slov dělitelná parametrem, budou tato slova často ukončovat jednotky a ty pak budou poměrně krátké (s dopadem na přesnost nástroje). Této nevýhody se lze zbavit odstraněním častých slov již ve fázi předzpracování.

Volba základních jednotek pro porovnání je stěžejním předpokladem pro nástroj, který má dosahovat vysokých hodnot přesnosti a úplnosti. Výše jsme popsali hlavní možnosti. Následující tabulka (převzato a upraveno z [Brin1995]) je ještě jednou shrnuje. N značí skutečnou velikost dokumentu (počet slov respektive počet základních prvků).

Tabulka 4: Vlastnosti různých možností volby základních jednotek porovnání (převzato a upraveno z [Brin1995])

<b>Naše značení</b>	<b>Popis</b>	<b>Požadavky na prostor</b>	<b>Délka jednotky</b>	<b>Bezpečnost (Brin)</b>
$V_1$	každé slovo představuje zvláštní jednotku	N	1	N
$V_x$	x slov za sebou tvoří jednotku, nepřekrývají se	N/x	x	1
$V_{xa}$	x slov za sebou tvoří jednotku, překrývají se	N-x	x	N/x
$V_{Hx}$	proměnlivá délka jednotek, ukončení podle hodnoty hashe, nepřekrývají se	N/x	průměrně x	N

64 Pokud předpokládáme normální distribuci hodnot hashovací funkce pro běžná slova.

65 Nebo první je dělitelné a druhé je posledním slovem dokumentu.

66 Byť je to u dokumentů standardní délky poměrně nepravděpodobné.

## 5.5 Volba hashovací funkce

Hashovací funkce obecně slouží k převodu prvků relativně velké množiny na vstupu na prvky jiné relativně menší množiny čísel (často bitových sekvencí o fixní délce) na výstupu. Kromě jiných aplikací je používána také pro urychlení vyhledávacích operací.

Právě k těmto účelům je již užíváno i v tomto modelu nástroje pro detekci plagiátů. Základní jednotky porovnání (vesměs textové povahy) jsou převedeny na své číselné reprezentace, se kterými je možné rychleji a efektivněji pracovat.

Ze samotného principu fungování typických hashovacích funkcí, plyne, že při jejím použití dochází ke kolizím. Různé vstupní hodnoty vedou na shodnou výstupní hodnotu. To v případě, že jsou v nástroji pro odhalování plagiátů porovnávány právě výstupy z hashovací funkce a nikoli samotný obsah, vede k větší pravděpodobnosti falešných poplachů (false positives) a snižuje přesnost nástroje. Pro správnou funkcionalitu je tedy důležité, aby kolize dané použité hashovací funkce nebyly pro danou množinu vstupů příliš časté. Pro zvýšení přesnosti je možné dokumenty, které jsou takovýmto nástrojem označeny jako podezřelé považovat za kandidáty a, jak doporučuje [Monostori2001], podrobit je ve druhém stupni odlišnému typu porovnání a to na základě přímého porovnání obsahu (viz dále).

Síla požadavku na nízké procento kolizí se odvíjí také od účelu, ke kterému má nástroj sloužit. Pokud jde o intrakorpální nástroj, který najednou pracuje pouze s korpusem několika desítek až stovek dokumentů, je pravděpodobnost kolize jistě nižší, než pokud se jedná o extrakorpální nástroj pracující s rozsáhlou databází s počtem dokumentů řádově vyšším. Pravděpodobnost kolize tak roste s počtem takto zindexovaných dokumentů (respektive přesněji s počtem zindexovaných základních jednotek). Jde o to, vyvážit případné kolize s požadavky na prostor a rychlosť zpracování. Na rozdíl od Monostoriho, který považuje možnost kolizí za problém, například [Finkel2002] užívá pro hashování svých jednotek funkci MD5, která vrací 128 bitové číslo ve formě dvaatřiceti hexadecimálních znaků. Finkel píše, že kvůli úspoře místa a rychlejší práci pracuje ale pouze s prvními deseti z těchto hexadecimálních znaků. To samozřejmě velmi zvyšuje pravděpodobnost kolize. Jak však uvádí, nevidí pravděpodobnost kolize v řádu jedna na  $16^{10}$  základních jednotek jako problém.

Dalším možným omezením je použití variabilní délky základní jednotky a určování jejího konce na základě hodnoty hashovací funkce ( $V_{Hx}$ ). V tom případě je vhodné volit funkci tak, aby zlomová hodnota (např. dělitelná x) nevycházela na slovo, které se vyskytuje velmi často. Případně je možné volit pro výběr zlomů základních jednotek jinou funkci než pro jejich následné indexování.

Pokud jsme si těchto základních požadavků vědomi, můžeme volit libovolnou funkci z množiny standardních obecně dostupných, ověřených, efektivních a implementovaných hashovacích funkcí.

Specifická situace nastává, pokud jako základní jednotky volíme skupiny slov, postupně se překrývající ( $V_{xa}$ ). V tom případě by bylo možné s úspěchem použít hashovací funkci, která umožňuje výpočet hodnoty pro následující základní jednotku na základě hodnoty pro předchozí jednotku. Takové funkce jsou potom velice rychlé (zejména pokud je délka jednotky velká) a umožňují zrychlit převod dokumentu na jeho reprezentaci. Takovou (rolling hash) funkci předpokládá také Karp-Rabin vyhledávací algoritmus.

## 5.6 Volba jednotek pro reprezentaci

Takto získané hodnoty, které pokrývají celý dokument, by již bylo možné poměrně snadno porovnat se stejně získanými hodnotami pro jiné dokumenty a některé nástroje (viz např. [Lyon2004]) to také dělají. Jiné ale ještě přistupují k redukci takto získaných popisů obsahu dokumentu.

Jak jsme již uváděli, při použití v extrakorpálních nástrojích s využitím databáze dokumentů je jedním z hlavních požadavků velikost uložených reprezentací dokumentů. Nejen, že velké dokumenty zabírají velké množství prostoru na paměťových datových nosičích, ale zejména dlouhé reprezentace dokumentů je časově náročné (i při využití velmi efektivních algoritmů) mezi sebou porovnat. Otázkou tedy je, zda je možné získat nepříliš zkreslené výsledky porovnáním nikoliv celých dokumentů, ale pouze jejich neúplných reprezentací, které budou pokrývat pouze zlomek původního dokumentu.

Jak teoreticky i prakticky dokazují například [Heinze1996] a [Broder1997], možné to je. Vyžaduje to však další transformaci, která může mít zásadní vliv na výkonnost systému. Kromě již zmiňované volby základní jednotky porovnání a její konstrukce je to právě volba toho, kolik a kterých konkrétních základních jednotek bude vybráno, aby tvořily reprezentaci dokumentu.

### 5.6.1 Počet jednotek v reprezentaci

Prvotní otázkou je, jak velká má vlastně být reprezentace dokumentu sloužící k porovnávání. Nabízí se dvě varianty. První z nich předpokládá, že vybíráme konstantní počet základních jednotek pro dokumenty různé délky. Tento přístup preferuje [Heinze1996], který hovoří o sto základních jednotkách na dokument<sup>67</sup>. Výhodou je úspora místa a snadný odhad velikosti případné databáze pro daný počet dokumentů. Zásadní nevýhoda však spočívá v tom, že při porovnání není možné pracovat s délkou dokumentů. Nelze tak využít výhod asymetrických metrik (viz kapitolu 4) a to ani v jejich symetrizovaném tvaru<sup>68</sup>. Druhou možností je namísto fixního počtu jednotek vzít poměrnou část ze všech jednotek. Alternativně lze obě varianty zkombinovat a například pracovat s velikostními třídami dokumentů, pro které je uchováván předem stanovený počet základních jednotek, případně stanovení minimálního a maximálního uchovávaného počtu na dokument.

Počet uchovávaných jednotek na dokument je parametr samozřejmě poměrně důležitý pro výkonnost systému. Příliš nízký počet bude dokument špatně reprezentovat a povede ke snížení úplnosti nástroje. Pokud bude počet naopak příliš vysoký, vzdáváme se výhod rychlosti a nižších prostorových požadavků na práci s částečnou reprezentací dokumentu. Konkrétní hodnoty se na první pohled velmi liší dle jednotlivých autorů. [Heinze1996] uvádí jedno procento původního dokumentu (což v případě kdy pracuje s fixní délkou 100 jednotek, vypovídá o průměrné délce dokumentu, se kterým počítá v řádu 10 000 slov<sup>69</sup>). Také [Broder1997] doporučuje počet okolo 100 až 200 jednotek na dokument. Naproti tomu [Finkel2002] nevolí fixní velikost reprezentace a doporučuje počet jednotek odpovídající (minimálně) odmocnině počtu všech základních jednotek dokumentu<sup>70</sup>.

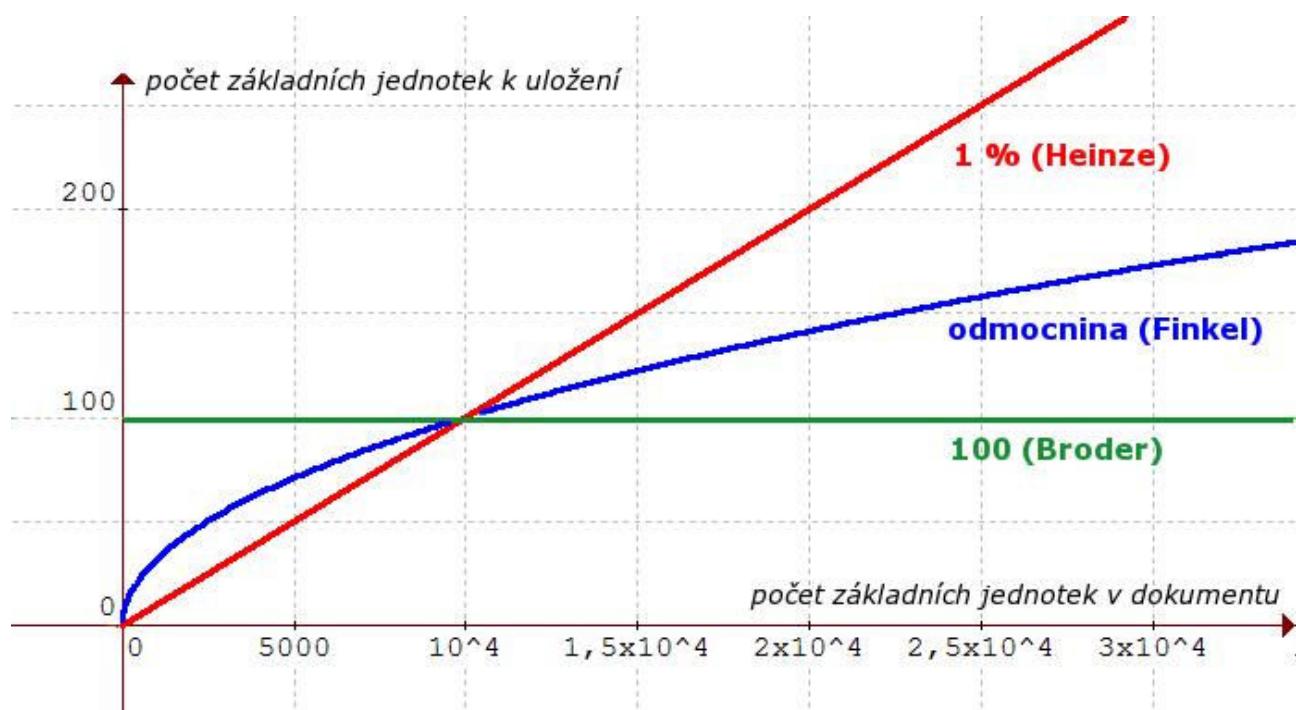
67 Přesněji hovoří o sto jednotkách pro uložení do databáze a tisíci při porovnávání vůči databázi.

68 U jednotlivých dokumentů by sice bylo možné kromě jejich reprezentací o fixní velikosti ukládat údaj také o původní délce dokumentu. Ten by mohl potom sloužit k úpravě výsledku (symetrického) porovnání. Vypovídající hodnota by však byla minimálně sporná.

69 Velmi zhruba. Heinze pracuje s částečně překrývajícími se jednotkami založenými na slovech čili počet slov zhruba odpovídá počtu základních jednotek v dokumentu; viz tabulky 3 a 4.

70 Finkel přitom používá proměnlivou délku základní jednotky; viz tabulky 3 a 4.

Za pozornost jistě stojí to, co je vidět na obrázku 8. Totiž, že obě (respektive všechna tři) doporučení se zhruba shodují u dokumentů o délce okolo 10 000 slov. Pro ty je odmocnina počtu slov rovna právě 100. Zřejmě není až taková náhoda, že je to právě ta velikost dokumentu, kterou implicitně předpokládá Heinze, když hovoří o jednom procentu. Zdá se, že pro dokumenty o velikosti řádově deseti tisíc slov je okolo sto jednotek empiricky ověřenou hodnotou pro spolehlivé porovnání<sup>71</sup>.



Obrázek 8: Shoda počtu základních jednotek pro uložení pro 10 000 základních jednotek v souboru

Pro představu, čemu tato hodnota odpovídá v praxi, si uvědomme následující. Deset tisíc slov znamená cca 50 000–60 000 znaků (k průměrnému počtu znaků ve slově viz poznámku 47<sup>72</sup>). To může odpovídat zhruba třiceti až čtyřiceti stranám<sup>73</sup> textu. To je rozsah, který odpovídá poměrně dobře běžné kratší vědecké (bakalářské) práci. Při použití na kratší dokumenty tedy bude zřejmě fixní velikost 100 jednotek dostačující. V případě propoří velikosti se ale zdá výhodnější ukládat o něco více než jedno procento jednotek. Finkelova odmocnina se jeví jako poměrně dobré řešení. Musíme však počítat s určitou výhradou, že její nelinearita může lehce zkreslit hodnotu metriky<sup>74</sup>.

71 Přestože je třeba mít na paměti, že Heinze a Finkel užívají jinou základní jednotku. Viz poznámky 69 a 70.

72 Předpokládáme, že v angličtině je průměrná délka slova zhruba stejná jako v češtině.

73 Při normostraně 1800 znaků (ovšem včetně mezer). Případně zhruba 250 slov na stranu.

74 Například dokument A má 10 000 slov (resp. základních jednotek), dokument B je pouze kopie první poloviny dokumentu A (tj. má 5 000 základních jednotek). V případě porovnání všech jednotek obou dokumentů by klasická symetrická metrika podobnosti byla  $5\ 000 / 10\ 000$  (průnik) / 10 000 (sjednocení) = 0,5. V případě použití odmocniny pro určení počtu uchovávaných jednotek, by pro první dokument bylo uchováno 100 jednotek a pro druhý 70. Relativní velikost se tak změnila z 2:1 na 10:7. Za předpokladu, že by se všechny jednotky uchované pro dokument B byly uchovány také pro dokument A, hodnota metriky po vzájemném porovnání by byla  $70 / 100$  (průnik) / 100 (sjednocení) = 0,7. Opět by bylo možné uchovávat také původní délku dokumentů a užívat ji k přepočtu hodnot metrik podobnosti, v praxi se tak ale zřejmě téměř nikdy neděje.

Všimněme si také, že v tomto konkrétním příkladě jsme touto transformací získali od symetrické metriky podobnosti lepší (vyšší) výsledek, který více odpovídá tomu, že dokument B je kopí části dokumentu A. Nemusí to však být pravidlem.

### 5.6.2 Volba konkrétních jednotek

Ještě důležitější než počet ukládaných jednotek, je ale volba těch, které se mají uložit. Na významu toto tvrzení nabývá tím více, čím menší část tyto jednotky ve skutečnosti pokrývají. Výše jsme viděli, že v praxi je to i pouhých několik procent.

Vybírané jednotky by měly být v dokumentu rozloženy relativně rovnoměrně tak, aby jej reprezentovaly celý. Z hlediska bezpečnosti je také vhodné, aby byly vybírány jednotky poměrně náhodně tak, aby případný útočník u konkrétního dokumentu pokud možno nevěděl, které to budou. Rovněž je žádoucí, jak nakonec plyne také z příkladu s přidáním slova na začátek dokumentu (viz kapitolu 5.4), aby volba jednotek byla prováděna na základě jejich obsahu, nikoliv pořadí. Posledním, ale pro funkčnost zdaleka nejdůležitějším požadavkem je to, aby z podobných dokumentů byly vybírány podobné či stejné jednotky. Tedy pro extrémní případ, kdy zpracováváme dva totožné dokumenty, se do výběru v obou případech dostaly z velké většiny (pokud již ne naprosto) stejné jednotky. Jedině to při porovnání reprezentací obou dokumentů zajistí nalezení vysokého procenta shody.

Některé tyto požadavky si mohou odporovat a je proto důležité zvolit vhodnou selekční strategii, která je všechny dostatečně naplní. V literatuře se opět objevuje u různých autorů několik takových strategií. My si je zde stručně popíšeme.

Nejjednodušší strategie by byla prostá náhodná. Ze všech základních jednotek porovnání obsažených v dokumentu by se náhodně vybral odpovídající počet<sup>75</sup>. Taková strategie by byla velmi odolná proti útokům, ale jak ukazuje [Heinze1996], nesplňuje veleďležitou podmínu výběru podobných reprezentací pro podobné dokumenty. Ukazuje, že v případě dvou identických dokumentů o délce 50 000 slov (a tedy zhruba 50 000 základních jednotek) s fixní délkou 100 uložených jednotek, je očekávaný výsledek porovnání těchto dokumentů pouze 2 % shody<sup>76</sup>. To je jistě velmi málo pro naprosto totožné dokumenty a taková selekční funkce se tedy nemůže zajisté hodit pro porovnávání dokumentů, které nejsou totožné, ale pouze podobné.

Nevýhoda této strategie spočívá právě v tom, že při výběru nebene v úvahu obsah základních jednotek. Heinze doporučuje jinou, která tento problém poměrně dobře řeší. Navrhuje zaměřit se na hodnoty hash funkce pro jednotlivé základní jednotky. Doporučuje jako reprezentaci dokumentu zvolit ty s nejnižší hodnotou. Za předpokladu, že hashovací funkce přiřazuje hodnoty rovnoměrně, mělo by být zaručeno dostatečné pokrytí celého dokumentu a také relativně náhodný výběr<sup>77</sup>. Ten je přitom založený na obsahu a nikoli na pořadí a proto pro stejné dokumenty vybere stejnou reprezentaci a pro velmi podobné se bude velmi pravděpodobně podstatná část obou reprezentací shodovat.

Také druhý v literatuře ([Broder1997], [Malkin2005]) zmiňovaný způsob využívá obsahu základních jednotek ve formě hodnoty hashovací funkce. Nepracuje ale s jejími nejnižšími hodnotami<sup>78</sup>, ale s jejich dělitelností danou konstantou. Tato strategie na rozdíl od Heinzeho předpokládá spíše délku reprezentace úměrnou délce dokumentu. Obdobně jako při ukončování základních jednotek na základě dělitelnosti hodnoty hashovací funkce ( $V_{Hx}$  viz kapitolu 5.4), také zde je tato

75 Buď fixní například 100, případně takový, který závisí na délce dokumentu viz část o délce reprezentace.

76 A to ještě Heinze pracuje s 1000 jednotkami pro porovnávání a 100 pro uložení.

77 Samozřejmě, že pokud útočník zná použitou základní jednotku a její konstrukci a také použitou hashovací funkci a selekční strategii, může zjistit, které jednotky budou zvoleny jako reprezentace dokumentu. V praxi je taková znalost ale nepravděpodobná a navíc je možné korektně zapojit i prvek náhody. Viz dále.

78 Ani s nejvyššími, což by mohla být také variace na zmiňovanou strategii.

délka proměnná. Očekávaná délka reprezentace odpovídá poměru délky dokumentu (počtu základních jednotek porovnání v něm) a zvolené konstanty. Broder navíc doporučuje měnit konstantu podle velikosti dokumentu tak, že jednotlivé konstanty mají hodnoty  $2^j$  a pro delší dokumenty je vyšší j.

Další dva způsoby popisuje [Finkel2002]. Vychází přitom právě z použití základních jednotek proměnné délky ( $V_{Hx}$  viz kapitolu 5.4). Předpokládá, že kratší základní jednotka má nižší vypovídající hodnotu než delší. Naopak velmi dlouhé jednotky jsou sice pro daný dokument velmi reprezentativní, ale autoři nepodezírají plagiátory, z toho, že jsou tak líní a kopírují velmi dlouhé úseky. Proto považují za nejlepší jednotky střední délky. První způsob uchovává jednotky<sup>79</sup>, které jsou svou délkou nejbližší mediánu délky všech jednotek v dokumentu. Druhý způsob, vycházející z týchž předpokladů, ukládá ty jednotky, jejichž délka snížená o medián všech délek je menší než 0,1 násobek směrodatné odchylky délky<sup>80</sup>.

Ať zvolíme jakoukoliv strategii, měli bychom mít na paměti základní požadavky na ni:

- pro stejné/podobné dokumenty musí vybírat stejné/podobné jednotky
- zaručuje relativně rovnoměrné rozložení vybraných jednotek po dokumentu
- vybírá jednotky spíše na základě jejich obsahu než polohy v dokumentu
- je dostatečně bezpečná (tj. náhodná/utajená)

### 5.6.3 Vylepšení strategií

Heinze nepoužívá svou strategii v její čisté podobě, ale aplikuje na ní některá vylepšení pro zvýšení bezpečnosti a snížení počtu falešných poplachů (false positives).

Posílení bezpečnosti spočívá v tom, že čas od času mohou být přepočítány reprezentace dokumentů s mírně pozměněnými parametry (základní jednotka porovnání, hashovací funkce, selekční strategie)<sup>81</sup>. Také [Brin1995] uvažuje o posílení bezpečnosti tím, že se do celého procesu zavede proces náhody. Může například pro každý dokument existovat několik různých rovnocenných sad reprezentací. Volba té konkrétní, která se použije k porovnání, bude náhodná. Nemusí jít ani o exkluzivní sady, stačí například uložit větší počet základních jednotek a při každém porovnání se náhodně zvolí jejich podmnožina. Můžeme opět využít například dělitelnosti hash hodnoty danou konstantou.

Vyšší bezpečnost je tak ale vykoupena jinými nedostatkami. V Heinzeho případě je to čas potřebný k reindexování celé databáze. Předpokládáme přitom, že dokud není opět zindexována celá databáze, není možné vůči ní plnohodnotně porovnávat nové dokumenty. Přístupy s několika různými reprezentacemi zase předpokládají (pokud chceme zachovat stejný poměr základních jednotek v dokumentu a jeho reprezentaci při porovnání a tím i jeho přesnost) několikanásobně větší prostorové nároky.

---

79 Konkrétně počet odpovídající hodnotě celé části odmocniny počtu všech základních jednotek v dokumentu.

80 Jejich minimální počet je přitom opět roven celé části odmocniny z počtu všech základních jednotek v dokumentu. Pokud by jejich počet nebyl dle těchto podmínek dostatečný, pokračuje se s vyšší hodnotou násobku směrodatné odchylky.

81 Heinze neukládá plné verze textů, ale pouze jejich URL adresu. Nárůst požadavků na prostor tak není velký (řádově několik desítek bytů na dokument).

V přístupu pro snížení pravděpodobnosti falešných poplachů se také Heinze (podobně jako Finkel) zabývá tím, jak jsou různé jednotky reprezentativní. Protože však pracuje s jednotkami konstantní délky, nemůže na reprezentativnost usuzovat z ní, ale musí si pomocí jinak. Místo toho zjišťuje, jak často se dané konkrétní jednotky objevují v dokumentech. Ty, které se tam objevují velmi často, jsou málo reprezentativní a neměly by být součástí reprezentace. Řešení jsou dvě – buď sledovat jejich četnost napříč všemi zpracovávanými dokumenty a postupně se tak učit časté jednotky, nebo (jak to dělá Heinze) jednoduše ztotožnit četnost v rámci všech dokumentů s četností v rámci jednoho dokumentu. Předpokládáme tím, že to, co se často vyskytuje v jednom dokumentu, se nejspíše bude vyskytovat i v dokumentech ostatních a nemá to tedy z hlediska porovnávání velký význam<sup>82</sup>.

#### **5.6.4 Shrnutí volby jednotek pro reprezentaci**

V této části jsme si představili způsoby, kterými lze z původní množiny základních jednotek porovnání, které pokrývají celý dokument, vybrat mnohem menší reprezentativní podmnožinu, která umožní mnohem rychlejší porovnání, zabere méně místa a přitom přináší výsledky podobné těm, které bychom získali při porovnání všech původních jednotek. Tuto skutečnost dokazuje tabulka 5, která je převzata z [Heinze1996] a ukazuje shodu výsledku při porovnání celých dokumentů a jejich reprezentací. Šlo o srovnání několika technických zpráv z Carnegie Mellon University School of Computer Science z roku 1995. Původní korpus obsahoval více než 360 dokumentů, ale pro porovnání celých dokumentů byl zvolen (zřejmě kvůli časové a výpočetní náročnosti úplného porovnání) pouze zlomek dokumentů. Pro tento dílčí korpus se zdá shoda velmi přesvědčivá.

Pro správné fungování nástroje je třeba vhodně zvolit základní jednotky pro reprezentaci dokumentu. Základem k tomu je vhodná selekční strategie a správně stanovená délka reprezentace. Jako vhodné selekční strategie se jeví ty popsáné výše. Při jejich použití byla opakováně experimentálně zjištěna dobrá shoda pro cca 100 ukládaných jednotek na 10 000 jednotek dokumentu. Výsledky se mohou lišit dle konkrétního korpusu. Proto je při volbě parametrů potřeba brát ohled také na typické použití nástroje.

---

<sup>82</sup> Konkrétně Heinze ale takto nepracuje s celými jednotkami jako takovými, ale pouze s prvními pěti písmeny jednotky. Tento přístup údajně přináší mnohem lepší výsledky.

Tabulka 5: Srovnání výsledků porovnání celých dokumentů a jejich reprezentace (převzato z [Heinze1996]<sup>83</sup>)

<b><i>Shoda v procentech při porovnání</i></b>	
<i>reprezentací</i>	<i>celých dokumentů</i>
45	57
9	8,7
5	12
29	55
1	0,01
1	0,01
1	0,20
1	0,08
3	2,60
1	3,00
0	0
0	0,3
0	0,03
0	0,16
0	0,19

## 5.7 Porovnávání

Samotné porovnání reprezentací dokumentů je již poměrně standardní záležitostí na principu invertovaného souboru případně hashtable. Reprezentace dokumentů k porovnání jsou nejprve buď rekonstruovány z obsahu původních dokumentů (v případě, že nepracujeme s databází, respektive užíváme nástroj v intrakorpálním režimu) nebo načteny z databáze. Podle počtu (a velikosti) porovnávaných reprezentací dokumentů můžeme pracovat buď se všemi najednou, případně postupně.

## 5.8 Převod dokumentů do textového formátu

Dosud jsme hovořili o přípravě na porovnání a porovnání čistě textového dokumentu bez formátování. V praxi však požadujeme porovnání dokumentů, které máme v různých formátech s vyznačeným formátováním. Může jít jak o značkovací a typografické formáty textové, tak o formáty binární a to jak standardizované tak uzavřené proprietární. Kromě toho, že implementace porovnávacího algoritmu je samozřejmě mnohem jednodušší pro čistý text, různá struktura různých formátů znemožňuje jejich přímé porovnání<sup>84</sup>. Navíc nezřídka je v těchto formátech možné vyjádřit tutéž myšlenku či obsah (o který nám jde především) několika různými způsoby (například formá-

83 Základní jednotka 20 po sobě následujících souhlásek, fixní délka reprezentace 100, selekční strategie minimální hash hodnota s vyloučením jednotek, které mají na začátku obsahu nejčastěji se opakující pětice znaků.

84 Přestože obdobné porovnání binárních souborů stejného typu by bylo teoreticky možné při vhodné volbě základní jednotky.

tování pomocí stylu případně pomocí přímého přiřazení vlastnosti odstavci, různé způsoby uložení textu ve formátu PDF včetně obrázku).

Z těchto důvodů je tedy žádoucí převádět dokumenty před porovnáním z různých formátů do čistého textu. Na tento úkol existují různé nástroje pro převod, ovšem jejich výsledky mohou mít kolísající kvalitu a záleží také na implementaci aplikace, která příslušný dokument v daném formátu vyprodukovala. Na potíže s převodem upozorňují například [Brin1995] a [Heinze1996]. Nedokonalou konverzí může dojít ke snížení úplnosti (obsahově shodné části dvou podobných dokumentů se vlivem odlišného formátování převedou jinak) i ke snížení přesnosti (různé dokumenty sdílí chybně převedené části např. z hlavičky, které nemají být součástí textu).

Specifickým problémem je potom převod znaků s diakritikou z některých formátů. V PDF bývají někdy pro maximální kompatibilitu uloženy ve vektorové podobě, některé zahraniční nástroje pro převod např. z RTF nebo DOC nepodporují diakritiku.

Takováto konverze není nutnou součástí nástroje pro detekci plagiátů, ale v případě komerčního nástroje či služby by měl být standardem a právě tou přidanou hodnotou (vedle přehledného a intuitivního rozhraní), za kterou je uživatel ochoten zaplatit. Opět ale záleží, k čemu má být nástroj určen. Je zřejmé, že pro vesměs textové zdrojové kódy není žádný převod nutný. Pokud ale chceme porovnávat také dokumenty získané z webu, je nutností převod z HTML. Pro texty v techničtějších oborech je vhodná podpora Postscriptu, TeXu, případně PDF. Méně technologicky zaměřené instituce budou zřejmě vyžadovat plnou podporu minimálně pro textové dokumenty v MS Wordu. Zde je třeba počítat také s nastupujícími novými standardy pro kancelářské dokumenty (at' již OpenDocument Format či OpenXML).

Formátování dokumentu může ale také nést informací o případném plagiátorství. Odstavec formátovaný jinak než zbytek textu může značit, že byl přímo celý zkopirován z jiného zdroje<sup>85</sup>. Podobně například korektně uvedené citace by mohly být na základě formátování (uvozovky, kurziva, křížový odkaz) identifikovány. Takové operace však dle dostupných informací žádný z nástrojů neprovádí<sup>86</sup> a tyto možnosti odhalování plagiátů na základě formátování tak zůstávají pouze pro případnou ruční analýzu dokumentů, na které upozornil právě některý ze standardních nástrojů.

## 5.9 Přímé porovnání obsahu dokumentů a jiné přístupy

Někteří autoři se z různých důvodů nespokojují s porovnáním pouhé reprezentace dokumentu. Například výše zmínované připomínky uvedené v [Monostori2001] se vztahují k možnosti kolizí v hashovacích funkcích. Nejví se také jako vhodné používat tyto metody pro příliš krátké dokumenty. Rovněž jejich použití na dokumenty strukturované s omezenou zásobou výrazů (také často nepříliš dlouhé), jako například u zdrojových kódů v programovacích jazycích, není ideální.

Kromě toho, pokud má být nástroj pro detekci plagiátů efektivním pomocníkem nejen při detekci podezřelých dokumentů, ale také při dalším již ručním ověřování potenciálních plagiátů, je vhodné aby umožňoval přehledné zobrazení shod. Jako osvědčené a v mnoha nástrojích používané se jeví zobrazení plného textu dokumentů s vyznačením shodných pasáží. Uživatel tak může poměrně rychle rozhodnout, zda se shody vyskytují v podstatných částech dokumentu nebo v částech

---

85 V horším případě z jiného zdroje než okolní odstavce.

86 Ten za službou TurnItIn.com alespoň detekuje text v uvozovkách a bibliografické záznamy, ale výchozí nastavení je takové, že se porovnává a vyhledává všechno a až následně je možné některé tyto části vyloučit.

okrajových, kde je lze u daného typu dokumentu očekávat (například zadání, povinná hlavička atp.) Vhodné a velmi přehledné je také provázání shodných či podobných částí pomocí hypertextových odkazů, které umožní rychlý a hlavně přehledný pohyb v podezřelých dokumentech.

Takovou funkci však přímo neumožňuje výše uvedený postup porovnání. Nepracujeme tam s celým dokumentem, ale pouze s jeho relativně malou částí. Navíc u ní neuchováváme informaci o poloze a původním obsahu jednotlivých základních jednotek v originálním dokumentu. Pokud chceme tyto vlastnosti využít, musíme bud' modifikovat nastíněný model (například porovnat všechny základní jednotky) a přijít o výhody jeho rychlosti a vysoké kapacity, a nebo ho využít jako první stupeň pro výběr několika podezřelých kandidátských dokumentů po rychlém porovnání reprezentací velkého počtu primárních dokumentů. V dalším kole se pak budeme již zabývat pouze předvybranými dokumenty, kterých bude již řádově méně. Navíc je již budeme porovnávat pouze s dokumenty, kterým jsou podobné.

Požadavek na rychlosť porovnání již nebude tak silný jako v případě stovek a tisíců dokumentů v databázi. Důležitější bude pro nás uživatelský komfort a přehlednost zobrazených informací. Přímé porovnání kompletních dokumentů je samozřejmě přesnější a proto mu někteří dávají přednost již jako primárnímu. Výpočetně je však mnohem náročnější a proto je použitelné pouze na menší korpusy, například při intrakorpálním porovnávání řádově desítek až stovek dokumentů.

Jeho konkrétní implementace může být různá. Například nástroj Ferret ([Lyon2006]) provádí porovnání kompletních dokumentů stejným způsobem, jaký byl popisován výše s tím rozdílem, že používá všechny základní jednotky (neboli jeho selekční strategie vybírá všechny jednotky). Jiné přístupy většinou využívají různé varianty vyhledávání v řetězcích, zarovnávání nebo grafické reprezentace dokumentů.

### 5.9.1 Vyhledávání v řetězcích

Existují přístupy, které se snaží o přímé porovnávání obsahu dokumentů pomocí algoritmů určených původně pro vyhledávání v řetězcích. Před samotným zpracováním je opět vhodné provést kroky 0 a 1 zobrazené na obrázku 6 na straně 43 tedy převedení na prostý text a aplikace předzpracování obsahu. Částečně se i této metody týká bod 2 téhož obrázku, nejde tu však o volbu základních jednotek porovnání, jako spíše o volbu základních prvků v tom smyslu, jak jsme o nich hovořili v kapitole 5.4. Těmi bývají v případě volného textu nejčastěji slova (oddělená mezerou či koncem odstavce), ale v případě zdrojového kódu to mohou být celé řádky případně menší části oddělené znaky, které mají v daném programovacím jazyce zvláštní význam (například tečka, závorky atp.).

V klasickém vyhledávání řetězců je problém definován jako nalezení (prvního) výskytu relativně krátkého vzoru v relativně delším textu. Čas potřebný ke splnění tohoto úkolu je funkcí délky vzoru a délky textu. Konkrétní funkce závisí na použitém algoritmu, obecně lze říci, že se skládá z času potřebného pro předzpracování a času pro samotné vyhledávání. Existuje celá řada takových algoritmů a není naším cílem je tu jednotlivě popisovat. Jejich velmi pěkný přehled lze najít v [Charras1997]. Obecně jich však většina funguje na principu posouvání vzoru po textu, ve kterém se vyhledává. Primitivní algoritmus posouvá vzor vždy pouze o jeden prvek, efektivnější algoritmy využívají znalosti obsahu vzoru a již porovnané části textu k rychlejšímu posuvu, jak znázorňuje obrázek 9.

```
KDMKDMVHPLBSPSVSSJKZRTNP  

KDMV  

KDMV  

KDMV  

KDMV  

...
```

```
KDMKDMVHPLBSPSVSSJKZRTNP  

KDMV  

KDMV  

...
```

Obrázek 9: Základní princip urychlení algoritmu pro vyhledávání řetězců

V případě porovnávání dvou dokumentů však nemáme jediný vzor, který bychom v dokumentu hledali, ale velký počet potenciálních vzorů obsažených v prvním dokumentu. Jako vzory slouží postupně části jednoho dokumentu a pro ně se hledá shoda v ostatních dokumentech<sup>87</sup>. Shody v dokumentech mohou mít přirozeně různou délku. Většinou je předem stanovena minimální velikost shody, která je považována za významnou. Lze tím tak urychlit vyhledávání, protože je možné se zaměřit až na shody podstatné velikosti (například několik slov).

### 5.9.2 Další práce s jednotlivými prvky

K vyhledávání přímých shod lze také využít invertovaného souboru základních prvků ať již pro oba dokumenty nebo tak, že jeden je procházen sekvenčně a výskyty odpovídajících prvků jsou pomocí invertovaného souboru dohledávány v druhém dokumentu. Následně je v obou dokumentech sekvenčně ověřována shoda minimální délky. Pokud ovšem vyžadujeme shodu nějaké minimální délky a prvky v invertovaném souboru mají délku nižší, prohledávání může být zpomalováno velkým počtem falešných startů, kde sice shoda je, ale je kratší, než je požadovaná minimální délka<sup>88</sup>.

Pokud je ale každý prvek porovnáván zvlášť, můžeme již při zpracování dokumentu rozlišovat prvky různých typů. Při jejich porovnávání pak můžeme využít pestřejší škály výsledků, než pouze souhlasí – nesouhlasí. Lze tak pracovat s podmíněnými porovnáními. Například označíme-li nějaký prvek jako „nepodstatný“, pokud nebude odpovídat jinému při porovnání, nemusí to znamenat konec shody. Rozdíl oproti úplnému vyrazení takového výrazu z porovnání (například tím, že je při při předzpracování zcela vypuštěn)<sup>89</sup> je ale takový, že v případě, že se prvek shoduje, je možné jej započítat do podobnosti dokumentů.

Takové „nepodstatné“ prvky jsou vhodné například pro interpunkční znaménka (pokud je neodstraňujeme) a texty v závorkách. Ještě vhodnější se pro ně jeví použití pro komentáře ve zdrojových kódech. Pokud se komentáře neshodují, výsledek porovnání bude stejný jako kdyby se neporovnávaly. Když se však shodují, jejich shoda ovlivní skóre podobnosti dokumentů. Podobně je můžeme použít pro názvy metod či proměnných. Na rozdíl od toho, když je nebereme v úvahu vůbec, tak získáme

<sup>87</sup> To bývá z hlediska zpracování výhodnější, pokud to není možné, jsou dokumenty porovnávány vždy pouze dva mezi sebou.

<sup>88</sup> Řešením tak je pracovat namísto základních prvků s překrývajícími se jednotkami délkom odpovídajícími minimální požadované délce shody. To je vlastně tentýž přístup, který jsme popisovali v kapitole 5.2 jen s rozdílem, že jsou porovnávány úplné reprezentace dokumentů. Tak to činí již zmiňovaný nástroj Ferret.

<sup>89</sup> Což by bylo dozajista rychlejší, protože k porovnání by pak zbylo celkově méně prvků.

vyšší hodnotu pro dokumenty, které se shodují i v těchto z hlediska fungování nepodstatných detailech<sup>90</sup>.

Odlišení prvků můžeme učinit na základě jejich obsahu, pozice v dokumentu případně kombinace obojího.

### 5.9.3 Nejdelší společná část

Jiným přístupem je hledání nejdelší společné podsekvence dokumentů. Jde o klasický problém ([LCSwiki2006]), kdy je úkolem najít takovou sekvenci prvků, která je zároveň obsažena v několika dokumentech. Nejčastější je případ práce se dvěma dokumenty. Na rozdíl od podobného problému nalezení nejdelšího společného (pod)řetězce (substring) není u podsekvence (subsequence) nutné, aby jednotlivé prvky byly ve všech uvažovaných dokumentech přímo za sebou. Jejich relativní pořadí však musí být zachováno. Dostatečně názorně to ilustruje příklad na obrázku 10.

<b>nejdelší společná/ý</b>	
<b>(pod)sekvence</b>	<b>(pod)řetězec</b>
pro "XMJYAKUZ" a "MZJAKUWXZ"	
X M J Y A K U Z	X M J Y A K U Z
M Z J A K U W X Z	M Z J A K U W X
-----	-----
M J A K U Z	A K U

Obrázek 10: Rozdíl mezi nejdelším společným (pod)řetězcem a nejdelší společnou podsekvenčí

Pro detekci plagiátů se jeví jako vhodnější práce s nejdelší společnou podsekvenčí. Nejdelší společný řetězec by mohl být zkrácen až na polovinu už v případě vložení jediného slova do prostřed jinak zcela zkopiovaného dokumentu.

Některé aplikace těchto přístupů na problém detekce plagiátů byly provedeny pomocí datových struktur typu suffix trees a odpovídajících algoritmů. Podrobně se této problematice věnuje Monostori v [Monostori2001] a zejména [Monostori2002a].

Jinými aplikacemi tohoto problému jsou nástroje pro porovnávání souborů typu diff, a zejména různé bioinformatické aplikace, které hledají shody v genetické výbavě organismů na základě porovnávání jejich sekvencováných DNA. Ty Pracují často na principu co nejlepšího vzájemného uspořádání či zarovnání dvou textů tak, aby byly maximalizovány shody a minimalizovány části, které se nepřekrývají. A to je problém velmi dobře odpovídající také detekci plagiátů.

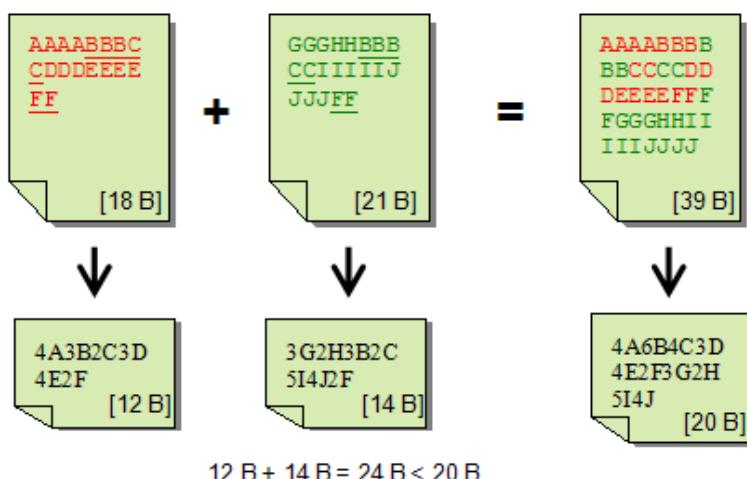
### 5.9.4 Komprese

Zajímavou metodu zvolili autoři [Chen2003]. Jejich přístup k detekci plagiátů vychází z teorie sdílené informace. Definují vzdálenost obsahu dvou dokumentů na základě Kolmogorovovy složitosti. Její approximace provádějí pomocí komprese dokumentů.

90 A které nejspíš pocházejí od nějakého línějšího plagiátora, který neztrácel čas důsledným zamaskováním své činnosti.

Bezztrátová komprese, jak známo, funguje velmi zhruba tak, že stejné části komprimovaného dokumentu se do výsledného zapíší pouze jednou a přidá se informace o jejich počtu a umístění v původním dokumentu. Pokud je součet velikostí dvou komprimovaných dokumentů podstatněji<sup>91</sup> vyšší než velikost zkomprimovaného dokumentu, který vznikl jejich sloučením (ještě před komprimací), pak mají dokumenty nejspíše něco společného. Lze usuzovat na to, že jejich efektivnější společnou komprimaci umožnilo to, že obsahují stejné úseky, které tedy mohly být ve výsledném dokumentu vynechány.

To vše platí pouze při dodržení určitých podmínek pro použitý kompresní algoritmus<sup>92</sup> tak je na základě poměru velikostí komprimovaných dokumentů možné usuzovat na jejich podobnost. V literatuře jsou citovány obdobné způsoby detekce plagiátů na základě komprimovatelnosti i v pracích jiných autorů<sup>93</sup>. Velmi obecné a silně zjednodušené schéma, jak může detekce podobnosti dokumentů na tomto základě fungovat je na obrázku 11.



Obrázek 11: Hrubá představa porovnání s pomocí komprese

## 5.10 Snížení počtu porovnání

Porovnávání velkého počtu dokumentů je obecně velmi náročné a velká většina dílčích porovnání končí konstatováním, že dokumenty nejsou podobné. I proto se hledají způsoby, jak počet nutných porovnání snížit. Zde si ukážeme několik implementačně jednoduchých a důmyslných možností, jak lze poměrně výrazně snížit počet porovnání respektive velikost korpusu při zachování odpovídajících výsledků. Snížením počtu základních jednotek porovnání v kapitole 5.6 jsme snížovali velikost porovnávaných entit a celkový počet porovnání dokumentů byl stejný. Ten byl obecně řádově  $\frac{1}{2}(\text{počet dokumentů v korpusu} + \text{v počet dokumentů v databázi})^2$ . Nyní snížujeme právě počet porovnání. Základní myšlenkou je vůbec neprovádět ta porovnání, která nemají šanci na pozitivní výsledek případně jsou nežádoucí či nezajímavá.

91 Toto je pouze velmi hrubý nástin principu. Ve skutečnosti je proces podstatně složitější jak teoreticky, tak prakticky. Pro detaily odkazujeme na [Chen2003].

92 Při práci s dlouhými shodnými řetězci může být omezující např. velikost bufferu a podobně.

93 Např. Saxon, S: *Comparison of Plagiarism Detection Techniques Applied to Student Code*, 2000, Trinity College, Cambridge Part II Computer Science Project

### 5.10.1 Metadata u dokumentů

Jedním z těchto přístupů použitelným pro extrakorpální nástroje s databází je kategorizace dokumentů (či jejich reprezentací). Společně s dokumentem či jeho interpretací uložíme také metadata, které budou reprezentovat obecně kategorie dokumentu, jeho určení, tematické zaměření, formu a podobně. Uživatel potom před porovnáním nejen zvolí korpus a případné parametry porovnání, ale také zadá dotaz, na jehož základě budou z databáze vybrány ty dokumenty, které se budou s korpusem porovnávat.

Popisné informace i dotazy by měly mít zřejmě spíše podobu uzavřených předem daných kategorií. U klasifikace obsahu si lze představit obdobu knihovních selekčních jazyků. Není totiž žádoucí, aby případným příliš konkrétním dotazem příliš zúžil počet potenciálních kandidátů. Navíc takové širší kategorie je možné přiřadit najednou pro mnoho dokumentů, které jsou indexovány společně (například řešení stejného úkolu od různých studentů). Kromě obsahové charakteristiky je možné použít i jiné kategorie jako například vyučovací předmět či odpovědný vyučující<sup>94</sup>.

U extrakorpálních nástrojů, které mají v databázi uchovány také jiné dokumenty, než ty přímo ručně do nich nahrané k porovnání (například samostatně indexují také dokumenty z webu) by mohl nastat problém s přiřazováním metadat těmto dokumentům. Pokud by byly indexovány dokumenty víceméně náhodně s důrazem spíše na kvantitu než podobnost těm existujícím (stylem, jakým to dělají internetové vyhledávače) jeví se jako nejsnadnější řešení přiřadit těmto dokumentům zvláštní hodnotu pro všechny kategorie (např. hodnotu „WEB“). Při porovnání by se pak uživatel mohl rozhodnout, zda chce své dokumenty porovnávat také s těmi z webu. U nástrojů, které dokumenty z webu nenacházejí preventivně dopředu, ale až na základě obsahu porovnávaných dokumentů (např. vyhledáváním frází z textu ve vyhledávačích) by bylo možné jim přiřazovat některá metadata shodná s těmi, která byla (ručně) přiřazena porovnávanému dokumentu, kterému jsou podobné.

### 5.10.2 Nové a staré dokumenty

Tato metoda je použitelná v případě, kdy jsou porovnávány vždy právě dva dokumenty vůči sobě. Spočívá v rozdelení porovnávaných dokumentů na dvě skupiny, se kterými se pracuje odlišně. Ten-to způsob využívají například intrakorpální nástroje WCopyFind ([wcopyfind]) a Pl@giarism ([plagiarismtk]). Uživatel nahrává vlastně dva korpusy, z nichž jeden obsahuje soubory označené jako „nové“ a druhý „staré“. Odlišnost zpracování spočívá v tom, že zatímco nové soubory jsou porovnávány jak mezi sebou navzájem, tak se starými, staré mezi sebou porovnávány nejsou.

Nástroj je tak vlastně částečně i extrakorpální, když umožňuje porovnání klasického korpusu „nových souborů“ mezi sebou a navíc vůči „databázi“ „starých souborů“. Staré soubory mohou být například dokumenty odevzdané studenty v loňských ročnících. Ty již byly mezi sebou zkонтrolovány po jejich odevzdání a je tedy zbytečné kontrolovat, jestli se mezi nimi náhodou nevyskytuji plagiáty. Naproti tomu nové dokumenty mohou obsahovat jak dokumenty – plagiáty založené na letošní práci jiných studentů, tak dokumenty – plagiáty založené na pracích z dřívějších let.

Pokud je počet „starých“ dokumentů větší než počet „nových“ (a to tato interpretace předpokládá), může být snížení počtu porovnání opravdu významné. Pokud nebudeeme rozlišovat a porovnáváme všechny dokumenty každý s každým, potřebujeme provést následující počet porovnání.

---

<sup>94</sup> Pokud je například z nějakého důvodu nežádoucí porovnávat dokumenty odevzdávané různým vyučujícím.

$$p = \frac{N}{2} \times (N-1) = \frac{N^2 - N}{2}$$

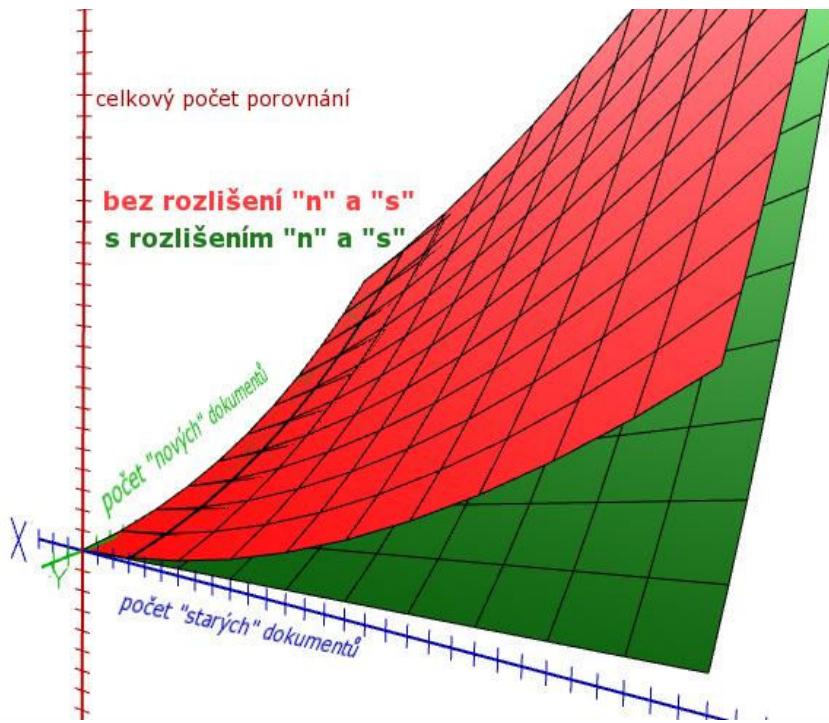
Kde N představuje počet dokumentů.

Rozdělením můžeme dosáhnout úspory takové, že stačí počet porovnání odpovídající následujícímu.

$$p_{sn} = \frac{N}{2} \times (N-1) - \frac{s}{2} \times (s-1) = \frac{(n+s) \times ((n+s)-1) - s \times (s-1)}{2} = \frac{n^2 + n \times (2s-1)}{2}$$

Kde n je počet „nových“ dokumentů, s počtem „starých“ dokumentů a N počet všech dokumentů tj.  $N=(s+n)$ .

Samozřejmě je počet porovnání stále řádově roven druhé mocnině počtu dokumentů, ale pokud je rozdělíme na „nové“ a „staré“, počet porovnání bude řádově odpovídat druhé mocnině pouze nových dokumentů (kterých předpokládáme méně). Staré budou přispívat k počtu porovnání již pouze lineárně. Obrázek 12 ukazuje vysokou kvadratickou citlivost počtu porovnání na počet nových dokumentů a naopak mnohem pomalejší lineární náběh způsobený růstem počtu „starých“ dokumentů. Můžeme srovnat průběh spodní (zelené) plochy představující nástroj, který pracuje tak, že odlišuje „nové“ a „staré“ dokumenty s horní (červenou), která představuje nástroj nerozlišující dokumenty v korpusu. I z grafu tak jasně vidíme výhodu, které se dosáhne, pokud je počet „nových“ dokumentů výrazně nižší než počet těch „starých“.



Obrázek 12: Srovnání počtu porovnání 1:1 při využití metody "nových" a "starých" dokumentů a bez ní

Oba výše zmíněné způsoby optimalizace počtu porovnání lze samozřejmě s úspěchem kombinovat, například porovnávat „nové“ odevzdávané dokumenty navzájem a vůči „starým“ (které se mezi sebou neporovnávají) vybraným navíc z databáze na základě metadat.

## 5.11 Shrnutí

V této kapitole jsme probrali nejčastěji používané metody pro detekci plagiátů ve volném textu. Kromě poměrně podrobného popisu metody založené na porovnání reprezentací dokumentů jsme zmínili i některé další podrobnější a náročnější přístupy založené na porovnávání kompletních obsahů. V poslední části jsme zmínili i několik možností optimalizace založených nikoliv na volbě efektivních algoritmů, ale spíše na organizačních opatřeních při porovnávání.

Jak jsme již uváděli výše, velký rozvoj v detekci plagiátů volného textu provázely také inovace v oblasti detekce ve zdrojových kódech. Ani jich se již dávno zdaleka netýká pouze počítání jednotlivých typů prvků jako kdysi v klasických attribute counting systémech. Užívají se v nich nezřídka metody podobné těm, které byly zmiňovány v této kapitole. Vzhledem k odlišnosti zdrojových kódů od přirozeného jazyka je ale potřeba tyto metody patřičně uzpůsobit. O tom stručně pojednává následující kapitola.

## 6 Specifika detekce plagiátorství u zdrojových kódů

*Within a computer natural language is unnatural.*

*In a 5 year period we get one superb programming language – only we can't control when the 5 year period will begin.*

*Alan Jay Perlis, americký počítačový vědec (1922–1990)*

V předchozí kapitole jsme poměrně podrobně věnovali obecným principům automatické detekce plagiátů. Předpokládali jsme přitom, že detekce probíhá na relativně velkých dokumentech psaném v přirozeném jazyce. Jak jsme ale viděli v kapitole 1.3, samotný vznik oblasti automatické detekce plagiátů před více než třiceti lety byl spojen s výukou programování a tedy nikoliv s texty v přirozeném jazyce, ale se zdrojovými kódy v tehdejších programovacích jazycích.

V této kapitole se proto zaměříme zvlášť zejména na tuto oblast. S mohutným rozšířením osobních počítačů a také masové výuky programování totiž stoupá také potřeba detekce plagiátů ve zdrojových kódech. Nejde ale jen o akademické prostředí. Detekce softwarových plagiátů může být důležitá také v komerční sféře v době existence patentů a tvrdé konkurence. V rámci open source komunity je přitom dostupné velké množství volně publikovaného kódu.

Přestože obecná východiska detekce jsou podobná, programovací jazyky mají jistá specifika, která brání úspěšnému použití některých výše zmínovaných technik a naopak umožňují zvýšit výkonnost nástroje pomocí jiných.

### 6.1 Specifika zdrojových kódů programů

Základní rozdíly mezi přirozeným a programovacím jazykem plynou z jejich rozdílného určení. Zatímco přirozený jazyk vzniklý historicky a spontánně slouží ke komunikaci mezi lidmi, která někdy není zcela jasná a přesná, programovací jazyky byly vytvořeny lidmi za cílem naprosto přesné komunikace se strojem.

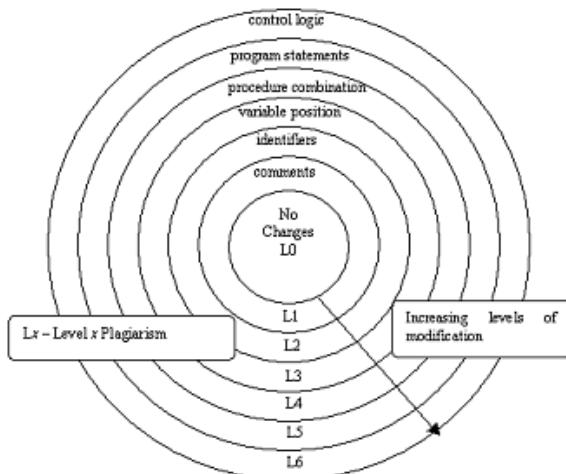
Vyjádření v programovacím jazyce je co nejstručnější<sup>95</sup>. Zdrojový kód je strukturován nejčastěji pomocí řádků, a dále i v několika úrovních pomocí různých sekcí (hlavička, oblast pro definici proměnných, jednotlivé funkce či metody, objekty...). Lze používat různé typy volání funkcí umístěných jinde v dokumentu nebo dokonce i v jiných dokumentech.

Zdrojové kódy jsou tak vysoce strukturované, mají pevně danou a zcela známou syntaxi. Mají přesně daný význam některých definovaných slov a naopak místa, kde se mohou vyskytovat slova, která nejsou v jazyce definována. Změny v některých částech zdrojového kódu (kommentáře, názvy metod, funkcí a proměnných,...) nemají žádný vliv na programem vykonávanou činnost.

V dostupných zdrojích (např. [Clough2000], [Noh2003]) je velmi často citována klasifikace od Faidhiho a Robinsona. V ní rozlišují šest respektive sedm úrovní plagiátorství ve zdrojových kódech. Jednotlivé úrovně se liší stupněm změn. Každá vyšší úroveň zahrnuje náročnější operace a je tedy také náročnější z hlediska detekce. Toto schéma na obrázku 13.

---

<sup>95</sup> Samozřejmě to neplatí vždy



Obrázek 13: Úrovně plagiátorství ve zdrojových kódech dle Faidhi a Robinson (převzato z [Noh2003])

Objevují se také jiné klasifikace, od jiných autorů. V [Lancaster2005a] je uvedena prakticky velmi podobná posloupnost, vytvořená Jonesem, která zachycuje devět stupňů různých transformací, kterými se plagiátoři mohou snažit ve zdrojových kódech zakrýt své činy. Přebíráme ji na obrázku 14.

1. Přesná kopie
2. Změna komentářů
3. Změna mezer, konců řádků (whitespaces) a formátování
4. Přejmenování identifikátorů
5. Změna pořadí bloků kódu
6. Změna pořadí příkazů v rámci bloků kódu
7. Změna pořadí operandů/operátorů ve výrazech
8. Změna datových typů
9. Přidání nadbytečných příkazů nebo proměnných
10. Náhrada řídících struktur jejich ekvivalenty

Obrázek 14: Transformace plagiátů ve zdrojových kódech dle Jonesa (převzato z [Lancaster2005a])

Schémata na obrázcích 13 a 14 jsou si svým obsahem velmi podobná. Zároveň ukazují, na co vše je vhodné dbát při implementaci nebo výběru nástroje pro detekci plagiátů ve zdrojových kódech. Teorie je v této oblasti poměrně dobře rozvinutá a autoři sdílejí ideje svých předchůdců. Situace tu je tak rozhodně přehlednější, než v oblasti detekce volného textu. Zřejmě je to dáno o několik desítek let delší historií a tím, že jak teoretici, tak implementátoři nástrojů pro detekci plagiátů<sup>96</sup> se rekrutují zejména ze sféry programátorů, nebo lidí, kteří o programování a programovacích jazycích vědí teoreticky i prakticky mnohem více, než například o zpracování přirozeného jazyka.

Obdobně je sdílena definice pánů Hamblena a Parkera, která říká, co je to plagiát vzhledem právě k práci se zdrojovými kódy. Tato definice zní: „Plagiát zdrojového kódu je takový zdrojový kód,

<sup>96</sup> Hovoříme zde zejména o akademickém prostředí. V něm mají podobné nástroje dlouhou tradici. V komerční sféře může být situace odlišná. Většina společností si často úzkostlivě chrání myšlenky, na kterých mají založeny své produkty. Vzhledem k větší schopnosti programátorů v akademické sféře naplnit vlastní potřeby detekce plagiátů (tj. práce se zdrojovými kódy) se komerční firmy zaměřují spíše na oblast volného textu.

který byl vytvořen z jiného zdrojového kódu pomocí triviálních operací a bez detailního porozumění kódu.“<sup>97</sup>

Všimněme si, že plagiát je opět definován procesem svého vzniku tak, jak jsme to zavedli již v první kapitole. Ony triviální operace můžeme docela dobře chápat jako ty transformace, které jsou uvedeny na obrázcích 13 a 14. Zajímavostí jistě je, že se zde kromě procesu vzniku hovoří také o absenci porozumění kódu. To velmi dobře odpovídá právě tomu, že jde o akademickou definici. Pokud student kód sice zkopioval, ale porozuměl mu a pochopil ho, nejednalo by se tak o plagiát. Pochopením (a použitím) cizího kódu se student naučil něco nového, cíl výuky byl naplněn, a příště bude zřejmě schopen obdobný kód vytvořit sám<sup>98</sup>.

## 6.2 Zpracování zdrojových kódů

Obsah zdrojových kódů je, jak již bylo řečeno, přímo určen ke zpracování počítačem. Existují tedy nástroje, které jsou jej schopny rozebrat a interpretovat v souladu s definicí toho kterého jazyka. Těmito nástroji jsou překladače případně interpretry tohoto jazyka. Jedna z možností detekce plagiátů tedy je, založit hledání potenciálních plagiátů přímo na funkcionality kódem popsaného programu. Jako podezřelý znak tedy můžeme brát to, že dva programy fungují podobně.

To ovšem může být nevhodné zejména v případě, kdy programy již ze zadání mají vykonávat podobné činnosti. Tak tomu bývá například právě v kurzech základů programování, kdy studenti dostávají relativně jednoduché (a tedy krátké) a relativně podobné úkoly. Navíc takový nástroj založený na překladači či parseru konkrétního jazyka není příliš univerzální. Velkou nevýhodou může být rovněž potřeba, aby porovnávané zdrojové kódy neobsahovaly žádné syntaktické či jiné chyby, které by bránily překladu nebo zpracování.

Druhou možností jak pracovat se zdrojovými kódy je považovat je prostě za texty a zpracovat je obdobně, jak to bylo popisováno v kapitole 5, víceméně na základě obsahu. Přitom je však třeba mít na paměti výše zmínovaná specifika zdrojových kódů. Protože mnoho klíčových slov se objevuje v téměř každém zdrojovém textu a naopak názvy proměnných mohou být velmi individuální, není jistě možné porovnávat pouze několikaprocentní reprezentace. Je tak potřeba porovnávat úplná znění dokumentů<sup>99</sup>. Protože zdrojové kódy jsou relativně kratší než volné dokumenty v přirozeném jazyce, a navíc mnohem snáze zpracovatelné, je to ale akceptovatelné i u většího počtu dokumentů.

Srovnání však musí mít jiné parametry než při zpracování běžného textu. Pokud chceme detektovat i něco více než přesnou kopii textu, musíme pracovat s odlišnými oddělovači prvků a kratším prvkům uzpůsobit i případnou minimální délku shody. Pokud se s takovou znalostí konkrétního programovacího jazyka vhodně převede dokument na základní prvky a aplikuje například na komentáře, názvy proměnných a funkcí textové řetězce parametr „nepodstatný“ (viz kapitolu 5.9.2), je možné s nimi pracovat jako s textem. To se již blížíme další velmi často užívané variantě, která kombinuje znalost syntaxe jazyka s porovnáváním textu.

---

97 V originále se přitom nehovoří o zdrojových kódech, ale přímo o programech. Nám se v kontextu této práce jeví jako lepší (byť možná trochu kostrbatější) hovořit o zdrojových kódech.

98 V komerční sféře a dle práva by naopak docela jistě i takový „pochopený“ softwarový plagiát byl plagiátem a v případě jeho odhalení by proti pachateli mohly být činěny příslušné právní kroky.

99 Případně nějakou zcela jinou formu jejich reprezentace, která reprezentuje zejména strukturu a ne tolik konkrétní obsah.

Třetí možností je totiž kombinace obou předchozích. Propojuje znalost jazyka a textové zpracování. Zdrojové kódy jsou ve fázi (textového) předzpracování převedeny na nějakou reprezentaci, která symbolicky představuje případná volání uvnitř programu, veškeré pro chod nepodstatné informace (komentáře, identifikátory,...) jsou odstraněny a následně proběhne textové porovnání. Příkladem může být nástroj nazvaný Sim, popisovaný v [Gitchell1999], nebo také YAP3 (viz [Wise1996]).

Druhá a třetí možnost navíc umožňují zpracovat případný větší projekt skládající se z více zdrojových souborů odevzdávaných například jedním studentem tak, že je kupříkladu před porovnáním spojí dohromady a vytvoří tak větší dokument. Alternativně je možné pracovat s jakousi nadstavbou, která umožní porovnání původních dokumentů tak, aby nebyly porovnávány dokumenty téhož autora a skóre bylo udáváno pro celý projekt. Podobně pracuje například nástroj SID ([Chen2003]), když projekty mají vytvořené odlišné adresáře v nahrávaném komprimovaném souboru. Pokud taková nadstavba není k dispozici, je potřeba výsledky porovnání jednotlivých dílčích částí zpracovat ručně. Nevýhodou v tomto případě je zbytečné porovnávání dokumentů téhož autora.

Pokud se při detekci ve zdrojových kódech používají variace na výše popsané postupy práce s volnými texty, liší se zejména dvě úvodní fáze tedy předzpracování a volba základních jednotek. Obě vycházejí ze syntaxe daného jazyka a mohou například odstraňovat komentáře, nahrazovat nebo zcela odstraňovat identifikátory proměnných a funkcí a podobně.

### 6.3 Univerzální zpracování zdrojových kódů

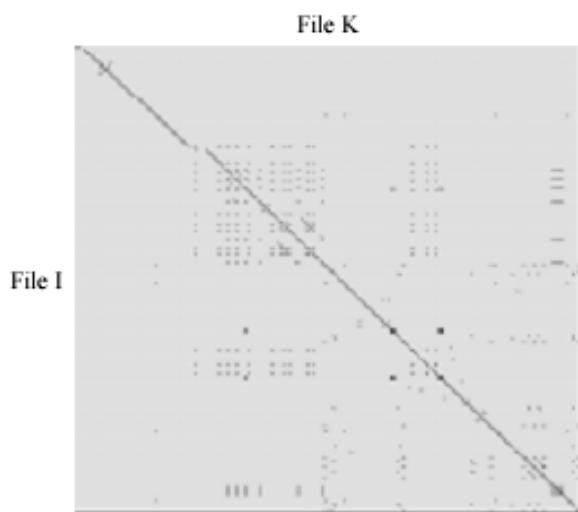
Nevýhodou všech výše uváděných možností je právě nutnost specializace na konkrétní programovací jazyk. Takový nástroj musí mít minimálně pro předzpracování implementovány informace o struktuře a významných slovech konkrétního jazyka. Proto se někteří autoři pokoušejí najít způsob, jak pracovat univerzálně s jakýmkoliv jazykem. Takové snahy se však pochopitelně musí vzdát výhod plynoucích ze znalosti struktury daných jazyků. Na místo známých prvků jazyka tak pracují s univerzálními prvky, které jsou společné většině jazyků. Například Ducasse s kolegy ([Ducasse1999]) ve svém nástroji za základní jednotku berou řádek textu. Ve snaze zůstat co nejvíce univerzální aplikují pouze jednoduché pravidlo předzpracování – odstraní všechny komentáře<sup>100</sup> a také všechny mezery. Dále porovnávají jednotlivé řádky takto upraveného dokumentu s jednotlivými řádky jiného dokumentu<sup>101</sup>. Používá se klasické porovnání celého řádku a výsledek je buď ten, že řádky jsou shodné, nebo rozdílné. Výsledky jsou ukládány do matice. Celkový výstup je potom zobrazen jednak v textové podobě<sup>102</sup> a také jako grafická reprezentace porovnávaných souborů.

---

<sup>100</sup> Takový nástroj tak není zcela univerzální, protože musí umět pracovat s různými typy komentářů v různých jazycích. Je ale mnohem univerzálnější a snáze upravitelný pro nový jazyk než ty, které pracují také s klíčovými slovy a gramatikou jazyka.

<sup>101</sup> Samozřejmě univerzálnost tohoto nástroje nespočívá v tom, že by bylo možné úspěšně navzájem porovnávat dokumenty vytvořené v různých programovacích jazycích.

<sup>102</sup> Jde mj. o jakýsi 'vzorek' (pattern), který sestává z pomlček, svislých čar a teček pro neshodu, shodu a odstraněný komentář.



Obrázek 15: grafické porovnání dvou dokumentů (převzato z [Ducasse1999])

Grafickým výstup vypadá jako na obrázku 15. Je to matice, kde její řádky představují postupně jednotlivé řádky prvního porovnávaného dokumentu a sloupce řádky druhého dokumentu. Tam, kde se řádky shodují, se v matici na příslušném průsečíku objeví tmavý bod. Pokud jsou naopak rozdílné, místo zůstane prázdné. Úseky, které jsou shodné, se projeví jako diagonály. Přerušené diagonály značí, že část kódu v těchto místech byla změněna. Pokud jsou diagonály v některých místech vůči sobě posunuté, značí to přidání nebo naopak odebrání některé části kódu.

Tento způsob vizualizace pochází také z porovnávání kódu DNA různých organismů, jako již některé dříve zmiňované techniky. Kromě univerzálního zpracování zdrojových kódů se tu tak objevuje také poměrně univerzální způsob vizualizace. Za zmínu jistě stojí i to, že obdobný typ grafu využívá pro prezentaci výsledků porovnání volného textu také nástroj VAST.

#### 6.4 Podobné problémy

Nad množinou zdrojových kódů lze provádět vyhledávání jejich vzájemných podobností z různých důvodů. Kromě hledání plagiátů může jít také o snahu o lepší návrh případně snazší údržbu softwaru. Podobné nástroje nezřídka využívají vývojáři systémů pro nalezení shodných či velmi podobných částí kódu, které se vyskytují na více místech a z hlediska údržby a dalšího vývoje by bylo výhodnější vyčlenit je na jedno místo. Taková aplikace sice řeší velmi podobný problém, jako je detekce plagiátů, ale některé aspekty jsou velmi odlišné, a proto není vhodné je volně zaměňovat.

Například autoři [Burd2002] zkoušeli právě za účelem údržby zdrojového kódu (clone detection), tedy vyhledávání kopii roztroušených na různých místech, několik nástrojů. Vybrali si však do srovnání také dva nástroje určené k detekci plagiátů ve zdrojových kódech (JPlag a MOSS)<sup>103</sup>. V některých testech se tak ukázalo, že například žádný z nich není schopen detektovat kopii kódu, která se vyskytuje ve stejném souboru. To je pochopitelné, neboť ten, kdo se zajímá o plagiáty, není až tak interesován tím, že v jednom dokumentu od téhož autora se cosi opakuje. Naopak, tato informace je podstatná pro toho, kdo chce zjistit, zda se v rámci jeho kódu (lhostejno v jakých souborech) neopakuje cosi příliš často.

<sup>103</sup> Není zřejmé jestli úmyslně s účelem prokázat odlišnost určení, či spíše proto, že oba nástroje jsou dostupné zdarma a mohly se zprvu jevit jako vhodné pro srovnání výsledků s komerčními produkty.

## 7 Kritéria testování nástrojů pro detekci plagiátů

*An acre of performance is worth a whole world of promise.*

*William Dean Howells, americký spisovatel a kritik (1837–1920)*

*You get what you measure. Measure the wrong thing and you get the wrong behaviors.*

*John H. Lingle, americký odborník na výkonnost podniků*

Až dosud jsme se zabývali vesměs obecnými možnostmi detekce plagiátů nanejvýš s některými konkrétními příklady určité části funkcionality. Pokud však chceme nyní přistoupit k praktickému srovnání nebo dokonce hodnocení skutečných, reálných existujících nástrojů, musíme si pro to stanovit pravidla. V naší klasifikaci z kapitoly 2.2 jsme se zabývali zejména tím, co ty které obecné nástroje dokáží. Nyní nás bude zajímat, jak dobře a rychle to dokáží některé konkrétní.

### 7.1 Metriky nástrojů na detekci

Existují a v literatuře jsou popsány některé teoretické možnosti porovnání výkonnosti nástrojů pro detekci plagiátů. Jejich praktické provedení k porovnání většího množství nástrojů však může být problematické. V následujícím textu nastíníme jak základní principy takového porovnání, tak i ona praktická úskalí.

#### 7.1.1 Základní měření výkonnosti nástrojů

Mezi základní míry výkonnosti nástrojů pro detekci plagiátů patří ty, zavedené v teorii vyhledávání informací (information retrieval) – přesnost (precision) a úplnost (recall). Při vyhledávání informací pracujeme s tím, jak relevantní dokumenty nám vyhledávací nástroj vrátí na základě dotazu. Vyhledávací dotaz je jakousi zjednodušenou reprezentací informačního požadavku. V případě jednoduchého modelu poměřujeme dokumenty relevantní a nerelevantní (informačnímu požadavku, nikoli vyhledávacímu dotazu!) a dokumenty vyhledávačem vrácené případně nevrácené<sup>104</sup>.

V případě detekce plagiátů je situace obdobná, i když úlohy jsou lehce pozměněny. Většinou pracujeme s porovnávanými dvojicemi dokumentů<sup>105</sup>. Uživatelský informační požadavek je stále stejný („najdi dokumenty, které jsou plagiátem“). Vzhledem k tomu není potřeba zadávat ani žádný vyhledávací dotaz, respektive vyhledávací dotaz by byl také stále stejný. Sestavit tento implicitní „dotaz“ je úkolem implementátora nástroje. Je pouze na něm, jak jej vyladí a zajistí, aby nástroj nalezl co nejvíce (pokud možno všechny) relevantních dokumentů (tj. dosahoval vysoké úplnosti) a zároveň nezahrálcoval uživatele dokumenty nerelevantními (tj. dosahoval vysoké přesnosti).

Relevantní dokumenty jsou takové, které jsou plagiáty. Již v kapitole 1, kde jsme plagiát obecně definovali, jsme zjistili, že je ale téměř nemožné jej přesně na definovat bez vazby na proces jeho

<sup>104</sup> V případě určování úplnosti je nutné znát počet všech dokumentů relevantních požadavku (tedy i těch, které na daný dotaz nejsou vráceny), ke kterým má vyhledávací nástroj přístup. I proto jsou praktické testy úplnosti poměrně problematické. Podobně je tomu i s praktickými testy nástrojů pro detekci plagiátů.

<sup>105</sup> Nejčastější je porovnávání dokumentů nebo jejich reprezentací každý s každým, o dvojice ve výsledku nejde, pokud nástroj zahrnuje clusterování do větších skupin, případně pokud se jedná o intrinsic nástroj.

vzniku (který většinou neznáme). Nezbývá nám tak nic jiného, než se uchýlit k různým heuristickým atď již teoretickým nebo empiricky ověřeným definicím založeným na podobnosti obsahu či struktury dokumentů. Každá implementace nástroje pro detekci plagiátů (případně každé nastavení parametrů jako základní jednotka porovnání, metrika, její výstražná hodnota atp.) je tak vlastně takovou praktickou definicí plagiátu. Navíc jsme se již zmíňovali o tom, že není rozumné požadovat od nástroje pro detekci, aby naprostě jednoznačně rozhodl mezi plagiátem a ne-plagiátem. Tento rozumný praktický požadavek teď pro účely teoretického měření výkonnosti prozatím opustíme a budeme předpokládat, že nástroj o každém dokumentu rozhodne, jestli je, nebo není plagiátem. Prakticky to lze implementovat například chápáním výstražné hodnoty tak, že cokoliv nad ní je automaticky považováno za plagiát.

Implementace nástroje tak může „definovat“ plagiát například tak, že jde o dokument, jehož alespoň 30 % vět, po odstranění často používaných slov a bez ohledu na pořadí slov ve větě, je obsaženo v jiném dokumentu korpusu. V tom případě je nejspíše základní jednotkou srovnání věta<sup>106</sup>, použitá metrika by mohla být obsah a výstražná hodnota 0,3.

Pokud použijeme různé nástroje na stejný korpus, o kterém víme, které dokumenty v něm jsou skutečné plagiáty<sup>107</sup>, budou se výsledky lišit právě tím, že každý nástroj je vlastně svou vlastní implicitní empirickou definicí plagiátu. U některých (zejména komerčních) nástrojů, ke kterým není dostupná dokumentace, navíc není možné snadno tuto definici odvodit.

S vědomím těchto omezení můžeme přistoupit k samotnému měření výkonnosti. Nejprve si označíme jednotlivé množiny dokumentů v hypotetickém korpusu, jehož složení známe a víme, co v něm jsou skutečné plagiáty. Vycházíme přitom ze značení, které používá Moussiades ([Moussia-des2005]). Předesíláme, že budeme vždy hovořit o dvojici porovnávaných dokumentů.

Nejprve budeme pracovat se skutečnými plagiáty, tedy těmi, o kterých je nám (pro účel pokusu) známo, zda plagiáty jsou nebo nejsou, respektive jsme je tak připravili nebo označili<sup>108</sup>. Mějme testovací korpus dokumentů. Množinu všech (neuspřádaných) dvojic těchto dokumentů označíme D. Tu můžeme rozdělit následovně. Jako SP (skutečné plagiáty) označíme množinu všech dvojic skutečných vzájemných plagiátů. Tedy množinu takových dvojic dokumentů, kdy jeden vznikl plagiátorstvím z druhého<sup>109</sup>. Jako SN (skutečné ne-plagiáty) označíme množinu všech dvojic dokumentů, které nejsou svými plagiáty. Čili se zde může objevit dokument – plagiát, ale pouze ve dvojici s jiným, kterého plagiátem není. Je zřejmé, že platí  $D = SP \cup SN$  a  $SP \cap SN = \emptyset$ .

Pokud se například bude korpus sestávat z následujících dokumentů:  $\{A, B, C, a, c\}$ , kdy dokumenty označené malými písmeny jsou plagiáty odpovídajících dokumentů označených velkými písmeny<sup>110</sup>, potom platí následující.

---

106 Respektive neuspřádaná množina slov po odstranění často používaných slov.

107 Atď již proto, že jde o laboratorní předem připravený korpus, nebo proto, že jde o reálný vzorek, který byl ale předtím důkladně prověřen lidským hodnotitelem.

108 Při běžném použití detektorů samozřejmě plagiáty předem neznáme, ale pro určení jejich výkonnosti je toto nezbytné.

109 Bylo by možné uvažovat, také že případně vznikli oba z dalšího třetího dokumentu (který může, ale i nemusí být součástí korpusu), ale pro jednoduchost od těchto alternativ nyní abstrahujeme.

110 Může tomu být i naopak tj. dokumenty označené velkými písmeny mohou být plagiáty dokumentů označených malými písmeny. V případě symetrické metriky to není možné odlišit a v případě asymetrických můžeme pouze předpokládat, že krátký dokument obsažený v dlouhém je jeho plagiátem. Dlouhý dokument ale může být kupříkladu plagiátem složeným z několika krátkých dokumentů a podobně.

$$SP = \{(A, a), (C, c)\} \text{ a } SN = \{(A, B), (A, C), (A, c), (B, C), (B, a), (B, c), (C, a), (a, c)\}.$$

Nyní se odkloníme od skutečných plagiátů a zaměříme se na ty dvojice, které jako plagiáty označí daný nástroj. Jako P označíme množinu dvojic, které daný nástroj označil za vzájemné plagiáty. A jako N označíme množinu dvojic, které daný nástroj neoznačil za vzájemné plagiáty.

Pak můžeme porovnat, jak dobře daný nástroj detekuje skutečné plagiáty. Jako TP (true positives) označíme ty dvojice, které nástroj správně označil jako vzájemné plagiáty, FP (false positives) pak dvojice, které byly nástrojem za plagiát označeny nesprávně. TN (true negatives) pak bude skupina označená nástrojem správně, že nejsou vlastní plagiáty a konečně FN (false negatives) je množina dvojic vlastních plagiátů, které nástroj nešetří.

Vše ještě jednou shrnuje následující tabulka.

Tabulka 6: True/false positives/negatives

<i>Dvojice dokumentů</i>	<i>nástroj označil, že</i>		<i>celkem</i>
	<i>jsou plagiáty</i>	<i>nejsou plagiáty</i>	
<i>ve skutečnosti</i>			
<i>jsou plagiáty</i>	TP	FN	SP
<i>nejsou plagiáty</i>	FP	TN	SN
<i>celkem</i>	P	N	D

Přesnost nástroje pro konkrétní korpus potom určíme následujícím způsobem.

$$pre = \frac{|TP|}{|TP| + |FP|} = \frac{|TP|}{|P|}$$

Čili klasicky jako poměr nalezených skutečných plagiátů a všech dokumentů, na které nástroj upozornil. Vysoká hodnota vypovídá o tom, že pouze malá část nalezených dokumentů, nejsou skutečně plagiáty. Nízká hodnota svědčí o tom, že nástroj je příliš citlivý a upozorňuje často na dokumenty, které plagiáty nejsou. Pro uživatele je samozřejmě výhodná hodnota vyšší, protože při praktickém použití slouží nástroj pouze k upozornění na nebezpečí plagiátorství a tato pozornění jsou dále ručně uživatelem prověřována. Vysoký počet falešných poplachů (FP), které je nutné bezvýsledně ručně kontrolovat, velmi snižuje uživatelskou přívětivost a využitelnost takového automatizovaného nástroje.

Přesnost je možné relativně snadno určovat i u reálných korpusů poté, co následně ručně ověříme dokumenty, na které jsme upozorněni. Naproti tomu pro určení úplnosti by muselo předcházet ruční prověření všech dvojic nezávisle na tom, zda je nástroj označí jako plagiáty.

Úplnost nástroje pro konkrétní korpus určíme jako:

$$rec = \frac{|TP|}{|TP| + |FN|} = \frac{|TP|}{|SP|}$$

Čili poměr skutečných plagiátů, které tak označil nástroj ku všem plagiátům. Vysoká hodnota značí, že se nástroji daří nacházet velké množství plagiátů. Nízká naopak svědčí o tom, že mnoho plagiátů zůstává nástrojem neodhaleno.

Obecně platí, že pokud u jednoho nástroje zvyšujeme úplnost, klesá jeho přesnost a naopak. Pokud budeme do nástroje implicitně zabudovanou definici plagiátu zpřísňovat tak, abychom se zbavili falešných poplachů, můžeme ztráct i část skutečných plagiátů. Vzhledem, k tomu, že implementátor nástroje má toto téměř výhradně pod kontrolou (ať již pevnými nebo volitelnými ale přednastavenými hodnotami parametrů), je právě zde možnost nástroje rozvíjet a vylepšovat. Různými empiricky vyzkoušenými vylepšeními lze dosáhnout pro určitou oblast použití poměrně vyváženého poměru mezi přesností a úplností.

### 7.1.2 Srovnávání výkonnosti různých nástrojů

Pro jednotné ohodnocení nástroje je možné použít nějaký kompozitní ukazatel, který spojí přesnost a úplnost. Kromě tradičních F-poměrů užívá Moussiades jednoduchý vážený součet, který nazývá TPDP total plagiarism detection performance, čili celkový výkon detekce plagiátů.

$$TPDP = pre + m \times rec$$

Parametr  $m$  udává, kolikrát je pro nás důležitější úplnost než přesnost. Je přitom doporučovaná hodnota větší než jedna. Vychází se z předpokladu, že potenciální ztráta skutečného plagiátu je pro uživatele horší, než několik zbytečně ručně prohlédnutých dokumentů, které se ukáží, že plagiáty nejsou. S touto argumentací můžeme v celku souhlasit, za předpokladu, že nástroj nabízí nějaké pokročilejší vizualizační nástroje pro snadnější ruční porovnání dokumentů.

Přímému srovnání výstupů jednotlivých nástrojů brání použití různých metrik respektive různých interních definic plagiátu. Takové srovnání by bylo velmi hrubé. Bylo by zřejmě možné provádět srovnání tam, kde bezpečně známe příslušné metriky a jsme schopni je případně normalizovat na stejný interval (např.  $<0,1>$ ). Kromě toho, že bychom museli řešit koncepční rozdíl mezi symetrickými a asymetrickými metrikami (nejspíše symetrizaci přes maximum pro zachování jejich výhod – viz kapitolu 4.5), nemohli bychom ale porovnávat výstupy nástrojů u nichž nám metrika není známa v případě, že není normalizovaná.

Určité lepší srovnání by bylo možné pomocí hodnot přesnosti a úplnosti. Jeho výhodou je také to, že srovnává nejen nástroje mezi sebou, ale také se skutečným stavem tj. co je skutečně plagiát. Přesnost a úplnost ovšem tak, jak jsme jejich použití definovali pro detektory plagiátů, (tedy ostré rozhodování nástroje<sup>111</sup> – buď je plagiát, nebo není plagiát), závisí na zvolené výstražné (tedy tedy přímo zlomové) hodnotě metriky, jak Moussiades upozorňuje. Nelze přitom pracovat s jedinou zlomovou hodnotou pro různé nástroje opět proto, že různé použité metriky mají, jak jsme viděli i v první části kapitoly, odlišný průběh.

Pro určení optimální velikosti výstražné hodnoty pro každý nástroj (a daný korpus) zavádí Moussiades další metriku. Nazývá ji OCC optimum cutoff criterion (optimální výstražná/zlomová hodnota). Její hodnotu pak definuje tak, že je to taková hodnota, která pro daný nástroj (a korpus) maximizuje hodnotu TPDP. Výstupy takto vyladěných nástrojů jsou pak již lépe srovnatelné.

<sup>111</sup> Pokud bychom připustili nebooleovské rozhodování nástroje (např. místo „dokument A je plagiátem dokumentu B“ bychom připustili výstup „dokument A je z 60 % plagiátem dokumentu B“), narazili bychom opět na problém různých metrik.

Obdobně by bylo možné rozšířit tuto úvahu OCC i na další parametry konkrétního nástroje. OTS optimum tool setting (optimální nastavení nástroje) by potom byla množina hodnot volitelných parametrů<sup>112</sup> (včetně zlomové hodnoty) konkrétního nástroje, která maximalizuje jeho TPDP pro konkrétní korpus. V tom případě by bylo teoreticky možné poměrně objektivně porovnat nejlepší možný výkon, který je ten který daný nástroj schopen podat. Prakticky si to ale lze představit pouze obtížně nebo s několika málo parametry.

Přesnost takového srovnání pomocí OCC (a tím spíše námi uvažované OTS) je ale vyvážena jeho menší vypovídací hodnotou pro reálné použití. Odpovídá situaci, kdy uživatel nastaví výstražnou hodnotu resp. všechny parametry nejlépe, jak může. Praktičtější se nám proto zdá, porovnávat hodnoty TPDP pro implementátorem určené případně doporučované hodnoty.

Je přitom stále třeba mít na paměti, že všechny tyto hodnoty platí pro vždy jen pro konkrétní korpus dokumentů a mohou se velmi lišit, pokud jsou nástroje použity na korpus jiný<sup>113</sup>.

### **7.1.3 Bezpečnost nástrojů**

Brin s kolegy se ve své práci věnované vyhledávání kopií dokumentů ([Brin1995]) také mimo jiné zabývali bezpečností takových nástrojů. Nejde teď o zabezpečení v klasickém smyslu přístupových práv nebo možnosti napadnout systém. Hovoříme tu o bezpečnosti nástroje z hlediska obejtí jeho detekční funkce.

Autoři definují tuto bezpečnost jako minimální počet změn, které je nutné ve skutečném plagiátu (respektive kopii jiného dokumentu) provést, aby jej systém nedetekoval. Nástroj, který má bezpečnost rovnu jedné jistě nebude tím nejlepším, protože stačí jediná změna v plagiátu a tento projde bez povšimnutí. Takto definovaný pojem bezpečnosti tedy také úzce souvisí s pojmem úplnosti detekce. Bezpečnosti si autoři všímají zejména z hlediska různých strategií zpracování, volby základních jednotek porovnání a podobně (blíže k témtoto tématům viz kapitolu 5).

Podobně je možné se ptát, zda má v případě daného nástroje vliv na jeho bezpečnost to, že plagiátor zná způsob, kterým detekce pracuje. Může jít jak o znalost konkrétních použitých algoritmů, tak o přístup k nástroji k provedení většího množství testů<sup>114,115</sup>. Z tohoto hlediska je pak nutné vážit, zda například posílit důvěru studentů tím, že jim umožníme zkontolovat, zda jejich práce nebudou označeny jako plagiáty. O možných opatřeních pro zvýšení bezpečnosti viz dále.

---

112 Například minimální požadovaná shoda, minimální délka slova, ale i parametry předzpracování jako například množina slov, která se odstraní.

113 Například nástroj silně optimalizovaný pro anglické jazykové prostředí (synonyma, stop-words) tak, aby dosahoval vysoké úplnosti při poměrně vysoké přesnosti, může při použití na korpus českých dokumentů selhat a být „poražen“ jednodušším nástrojem pracujícím na obecnějším principu.

114 Ve druhém případě nemusí mít přímo fyzicky (nebo elektronicky) přístup přímo ke stejné kopii nástroje, ale může si (u nástrojů dostupných zdarma na Internetu dokonce velmi snadno) opařit kopii vlastní.

115 A nemusí jít ani o testy provedené jednou osobou. Ve větším např. školním kolektivu se společným „nepřítelem“ - detektorem plagiátů může být takový „útok“ proveden v delším časovém období mnoha jedinců, z nichž každý provede pouze několik testů. V tom případě by za takový test mohlo být považováno i běžné použití nástroje s jehož detailními výsledky (co bylo detekováno) jsou studenti seznámeni.

### 7.1.4 Několik poznámek ke kapacitě a době odezvy

Doposud jsme se věnovali výkonnosti jakožto kvalitě výstupu. Na výkonnost se můžeme dívat také z hlediska kapacity zpracování. Může nás zajímat doba, za kterou je daný nástroj (na daném stroji) schopen zpracovat určitý korpus. Ta závisí zejména na efektivnosti použitého algoritmu. Obecně porovnání N dokumentů každého s každým vyžaduje řádově  $N^2$  porovnávacích operací. Vhodnou volbou datových struktur a algoritmů lze výrazně zvýšit výkon ve smyslu rychlosti zpracování. Zdaleka tedy nezáleží pouze na základní použité metodě či myšlence, ale také na detailech její implementace. V tomto textu nemáme prostor se podrobně různými takovými optimalizacemi zabývat a není ani naším cílem tak činit.

Vhodným uspořádáním korpusu a popisem jednotlivých dokumentů lze eliminovat porovnání, u kterých je předpoklad nízké pravděpodobnosti shody<sup>116</sup> a dále tak navýšit rychlosť zpracování. Limitujícím faktorem může být také práce s pamětí. Některé nástroje kvůli rychlejší práci načítají celý korpus do paměti najednou a tím je logicky limitována maximální velikost korpusu. Kromě použitých algoritmů je pro výkon podstatné také implementační prostředí a v neposlední řadě výkon stroje, na kterém nástroj běží.

U nástrojů či služeb pracujících v dávkovém režimu s frontou požadavků je kromě samotné doby zpracování (od počátku zpracování do jeho ukončení) podstatná také informace, za jak dlouho od odeslání požadavku se k výsledkům dostane uživatel/zákazník.

## 7.2 Sestavení korpusu pro testování nástrojů

V kapitole 8 představíme a porovnáme některé dostupné nástroje a služby pro detekci plagiátů. Vzhledem k výše zmiňované problematičnosti kvantitativního srovnávání výkonnosti různých nástrojů v praxi pomocí přesnosti, úplnosti, případně TPDP, jsme u nich ale takové testy neprováděli.

Mnohé námi testované nástroje jsou z typologického hlediska naprostě odlišné. Pro důkladné a vykovádající změření přesnosti (natož úplnosti) by bylo zapotřebí použít alespoň několik různých korpusů, jejichž strukturu bychom přesně znali. Rozdílné použité metriky v různých nástrojích by vyžadovaly individuální (a tedy nepříliš objektivní) stanovení zlomové hodnoty. O problematičnosti OCC a OTS jsme se již také zmiňovali.

Tyto obecné metriky mohou být vhodné například při ověřování výkonnosti (případně ladění parametrů) nově vyvíjeného nástroje oproti již existujícím referenčním nástrojům stejného typu, případně pro ověřování efektivity různých variant předzpracování, velikosti reprezentace dokumentu nebo selekční strategie. Nehodí se však dle našeho názoru pro obecné srovnání mnoha nástrojů, navíc pokud jsou i rozdílných typů. Takové srovnání by bylo časově velmi náročné a namáhavé a jeho výsledky by ani tak nebyly přesné a neměly by obecnou platnost. Protože u neznáme detaily většiny implementací, nezjišťovali jsme ani bezpečnost nástrojů.

V duchu určení této práce jsme porovnání pojali zejména jako uživatelské a praktické. Na vzájemné srovnání různých nástrojů pomocí testů jsme nerezignovali, ale připravili vlastní praktické až heu-

---

<sup>116</sup> Velký školní korpus se všemi pracemi studentů je jistě cenný zdroj pro porovnání. Při hledání plagiátů například ve slohových pracích prvního ročníku je ale zbytečné srovnávat je s fyzikálními protokoly třetího ročníku. Podobně je možné odlišovat i na detailnější úrovni například povinnou četbu a slohové práce atp.

ristické testy vlastností, které se nám zdály pro použití v českém prostředí z uživatelského hlediska nejpodstatnější.

Samozřejmě primární bylo zejména porovnávání přirozených i uměle vytvořených dokumentů v českém i anglickém jazyce. Jistě není možné brát námi dosažené výsledky jako obecné hodnocení, ale spíše jako náznak možností a konkrétní příklad použití na vybraném korpusu. I tak ale mohou takové testy mnohé naznačit o použitelnosti daného nástroje minimálně v českých podmínkách.

Kromě těchto základních testů jsme podnikli i některé doplňkové, které měly určit některé další parametry. Podnikli několik testů podpory českého jazyka a zpracování různých formátů dokumentů (včetně češtiny v nich), pokud jejich podporu nástroj deklaroval. Snažili jsme se také zjistit, zda je metrika použitá v některých nástrojích symetrická.

### 7.2.1 Intrakorpální testy

Schopnost detekce v intrakorpálním režimu jsme testovali zvlášť pro volný text a zdrojové kódy. V případě prostého textu jsme použili korpus z reálných seminárních prací. Šlo primárně o stovku vesměs náhodně vybraných dokumentů dostupných v archivu seminárních prací Ing. Jindřicha Zeleného [Zeleny2006]. Výhodou bylo nesporně to, že v některých letech se téma prací opakují, případně vyvíjejí. Navíc vycházejí nezřídka ze stejné oficiální dokumentace k daným produktům. Mezi stovkou dokumentů byly nejméně čtyři dvojice takové, které obsahovaly některou shodnou delší část textu. Samozřejmě to nelze v žádném případě automaticky považovat za plagiátorství. Tyto dokumenty jsme zvolili proto, že se jedná o dostupný publikovaný a zejména skutečný vzorek studentských prací<sup>117</sup> a je přitom poměrně dobrá šance, že některé dokumenty v něm mohou být podobné. Všechny testy byly prováděny prakticky s výchozím nastavením nástrojů.

Vzorek A tvořilo sto prací, kde čtyři dvojice byly významně podobné<sup>118</sup>. Tento korpus byl použit zejména pro základní intrakorpální testy a testy rychlosti zpracování. Text byl předem z HTML převeden na prostý text (se zachováním původní znakové sady jazyka<sup>119</sup>). Velikost byla v rozmezí od sedmi do třiceti kilobytů. Celkem mělo sto prací velikost cca 1,5 MB.

Vzorek B byl rozšířením vzorku A o dalších 181 souborů podobných charakteristik jako ve vzorku A, navíc zde byly rozdělené na skupiny po 10, 20, 50, 100 a 101 pro testování škálovatelnosti a schopnosti zpracovat větší množství dokumentů. Zde se již objevovala pouze jedna další dvojice velmi podobných dokumentů, ale tento vzorek sloužil zejména pro test schopnosti práce intrakorpálních nástrojů s větším počtem dokumentů<sup>120</sup> a rychlosti zpracování.

V oblasti zdrojových kódů jsme pracovali zejména s těmi v jazyce Java. Využili jsme přitom ukázkové soubory pocházející z demoverze produktu JPlag (vzorek J) a také testovací soubory pro různé případy modifikace kódu použité autory nástroje SID (vzorek S).

---

<sup>117</sup> Na rozdíl od například různými autory často testovaných technických dokumentů, RFC či dokonce různých překladů Bible.

<sup>118</sup> Jedna se shodovala téměř z poloviny, ostatní v několika odstavcích případně v částech více vět. Pátá dvojice potom obsahovala pouze shodný ukázkový zdrojový kód vložený v textu.

<sup>119</sup> Takže ta se mohla v konkrétním případě lišit od ostatních.

<sup>120</sup> Při porovnávání každý s každým je potřeba pro 280 dokumentů přes 39 tisíc jednotlivých porovnání.

### **7.2.2 Extrakorpální testy**

Pro test schopností extrakorpální detekce jsme použili několik různých dokumentů tvořících vzorek W. První z nich byl kratší anglicky psaný dokument vytvořený téměř výhradně textem z hesla „Money“ z anglické verze encyklopédie Wikipedia.org<sup>121</sup>. Velmi zajímavým (a téměř vždy odhaleným) byl dokument vytvořený doktorkou Hannah Dee z University of Leeds. Ten je celý složen z různých částí doslovně zkopiovaných z mnoha internetových stránek. Obdobně jako jsme využili testovací dokumenty z dema některých intrakorpálních nástrojů, jsme použili také dokument o Napoleonovi z dema služby TurnItIn.com. Test míry indexace placených zdrojů spočíval v testování dokumentu vytvořeného jako koláž z několika dokumentů získaných z databáze ProQuest. Jednalo se o náhodně vybrané dokumenty vyhledané na základě dotazu „Money“. Pro lepší regulérnost<sup>122</sup> byly vybrány pouze takové dokumenty, které byly vydány již v roce 2006.

Schopnost detekce v českém jazyce byla testována podobně dokumentem, který obsahoval několik odstavců z hesla „Peníze“ v české sekci encyklopédie Wikipedia. Lehce náročnější byl dokument pospojovaný z českých hesel „Peníze“, „Numizmatika“, „Měna“ a „Koruna česká“ z české verze Wikipedie. Druhým českým dokumentem byl popis Čapkova dramatu Matka dostupný na několika webech, které se zabývají čtenářskými deníky.

Podobně pro extrakorpální testy posloužily částečně i vybrané seminární práce ze vzorku A. Všechny totiž pochází z webu [Zeleny2006] a jsou standardně k nalezení vyhledávači<sup>123</sup>.

### **7.2.3 Další dílčí testy**

#### **Test podporovaných formátů dokumentů**

Při testování nástrojů jsme se nespokojili pouze s tvrzeními autorů o podporovaných formátech, ale také jsme je otestovali. K tomu sloužil vzorek C, který obsahoval tentýž dokument<sup>124</sup> v různých formátech. Šlo o DOC uložený v MS Word 2003, RTF, HTML výstup z MS Wordu, HTML výstup z OpenOffice.org<sup>125</sup> a dvě PDF vygenerované pomocí OpenOffice.org a znovu pomocí nástroje PrimoPDF. Pro kontrolu funkčnosti byl také připojen dvakrát původní dokument ve formátu prostého textu.

Protože šlo navíc o dokument český, byl to zároveň test podpory češtiny při převodu z různých formátů, což jak se ukázalo, není vždy bezproblémový proces.

#### **Test podpory češtiny**

Pro testování podpory českého jazyka při porovnání dokumentů byl vytvořen vzorek E. Ten tvořil opět několikrát stejný text<sup>126</sup> tentokrát uložený za použití různých českých znakových sad. Pracovali jsme s Windows-1250, UTF-8, ISO-8859-2, a dokument bez diakritiky (ASCII). První dva byly

---

121 Jejíž licence dovoluje volné použití obsahu a proto je tento také dostupný a dohledatelný i v jiných internetových zdrojích.

122 Např. kvůli zpoždění indexace.

123 Samozřejmě různými v různé míře.

124 Obsah Čapkova dramatu ze vzorku W.

125 Kde většina písmen s diakritikou byla zapsána pomocí HTML entit.

126 Jedna ze seminárních prací ze vzorku A.

ve vzorku zastoupeny dvakrát kvůli otestování schopnosti případného porovnání nepodporovaných ale stejných znakových sad<sup>127</sup>.

### **Test symetričnosti**

O rozdílu vypovídací hodnoty symetrických a asymetrických metrik jsme hovořili podrobně v kapitole 4. Protože často není vůbec jisté, jakou metriku daný nástroj používá, testovali jsme alespoň tuto základní vlastnost. K tomu sloužil vzorek E. Ten obsahoval dva dokumenty. Jeden byl výrazně (cca desetkrát) delší než druhý. Ten kratší byl přitom vytvořen téměř výhradně z obsahu většího dokumentu.

## **7.3 Shrnutí**

Z výše uvedeného je poměrně dobře patrné, že obecné praktické srovnání výkonnosti několika různých nástrojů pro detekci plagiátů je poměrně problematické. Tyto nástroje jsou založeny na různých principech a jejich vstupy a výstupy jsou různé, tedy jejich srovnávání musí probíhat na obecnější úrovni. Proto je potřeba přijmout řadu zjednodušujících předpokladů, jako například zvolit vhodnou zlomovou hodnotu a tím v rámci daného nástroje definovat plagiát, nebo se rozhodnout, jaké parametry nástroje nastavit. Výsledky takového srovnání jsou pak platné pouze pro daný korpus a nelze je jednoduše zobecňovat. Přesto je však takové srovnání možné a pro velký počet různých korpusů lze usuzovat na to, který nástroj obecně dosahuje lepších výsledků.

Trochu snazší situace je při porovnávání výkonnosti z hlediska kapacity a rychlosti zpracování různých nástrojů. Můžeme srovnávat, jak dlouho trvá zpracování korpusu dané velikosti v různých nástrojích běžících ve stejném prostředí. Rychlosť a kapacita ale nesmí být na úkor rozumných hodnot přesnosti a úplnosti.

S úplností úzce souvisí i pojem bezpečnosti nástroje definované jako minimální počet změn v dokumentu nutných pro oklamání nástroje. Na rozdíl od úplnosti, která pracuje s celým korpusem a je důležitá zejména pro uživatele nástroje, je pojem bezpečnosti pohledem z druhé strany. Vztahuje se (byť obecně) k jednomu dokumentu a je zajímavá zejména pro tvůrce nástroje a případně ty, jejichž dokumenty budou nástrojem testovány.

Pro naše uživatelské srovnání vybraných (nezřídka velmi rozdílných) dostupných řešení se nám jeví jako vhodnější použití jiného, praktičtějšího, typu testů. Pro intrakorpální testování volíme jeden, víceméně náhodný (ale reálný) korpus českých dokumentů. Pro extrakorpální testování volíme několik různých českých i anglických dokumentů. Kromě toho testujeme některé další vlastnosti. Výsledky přinášíme v následující kapitole.

---

<sup>127</sup> Např. ukončování slov jiným znakem než ASCII, nebo práci s vícebytovým kódováním v UTF-8.

## 8 Dostupná řešení

*If the only tool you have is a hammer, you tend to see every problem as a nail.*

Abraham Maslow, americký psycholog (1908–1970)

V této kapitole se podíváme na několik konkrétních dostupných nástrojů pro odhalování plagiátů. Jak je z dosud řečeného zřejmé, situace na tomto trhu není zdaleka stabilizována a neustále vznikají nové komerční i nekomerční produkty a jiné přestávají existovat.

Také v oblasti akademických výtvarů je situace složitá. Některé výše zmiňované koncepty různých autorů nemusí být dostupné jako skutečně fungující a veřejně dostupný nástroj. Může jít o prototypy, případně výsledky již ukončených projektů, které dále nepokračují a nerozvíjejí se. Lze jistě očekávat, že také existuje celá řada akademických implementací, které si autoři<sup>128</sup> vytvořili a dále upravují pro své vlastní potřeby a nepublikovali o nich žádné detailnější informace<sup>129</sup>. Stejně tak může jít i o rozsáhlejší systémy vytvořené na míru pro interní potřeby konkrétních institucí a často i jejich vlastními prostředky.

V tomto přehledu také nebereme v úvahu nástroje a služby, které jsou zaměřeny čistě na některou menší jazykovou oblast a o nichž jsou informace dostupné pouze v jiném než anglickém jazyce. Je ho tak třeba brát s těmito vsemi výhradami jako skutečně pouze částečně reprezentativní a mapující daný segment trhu v konkrétním časovém okamžiku.

Při charakteristice jednotlivých produktů se přidržíme vlastností důležitých zejména pro uživatele. Využijeme přitom námi upravenou klasifikaci Lancastera a Culwina podrobně probíranou v kapitole 2.2. Jako základní hledisko rozdelení přitom zvolíme licenci, respektive to, zda je nástroj dostupný zdarma nebo na komerční bází.

Po krátkém popisu základních vlastností a případně některých zjištěných detailů bude následovat také obrázek nástroje v činnosti (pokud je k dispozici nebo bylo možné jej pořídit). Zejména se budeme snažit takto graficky zprostředkovat proces zadávání dokumentů, souhrnné výsledky porovnání a detail. Na závěr budou u každého nástroje shrnutы jeho základní vlastnosti dle výše zmiňované klasifikace v podobě přehledné tabulky. Dále uvedeme nejzásadnější informace, které vyplynuly z našeho testování<sup>130</sup>. Podrobnější výsledky testů shrnují na závěr tabulky 19 a 20. Metodiku testování a důvod volby právě těchto testů jsme probírali v předchozí kapitole.

### 8.1 Výběr nástrojů

Jak již bylo zmiňováno výše, výběr nástrojů rozhodně není vyčerpávající. Představujeme zde zejména ty, které jsou často zmiňovány v dostupných zdrojích, je o nich známo něco více, a zejména jsou v současnosti stále dostupné a fungující. Úmyslně jsme volili různé nástroje intrakorpální, extrakorpální, i smíšené. Zahrnuli jsme ty, které jsou určeny k lokálnímu použití, tak i distribuované

128 To se dá očekávat zejména v oblasti výuky programování.

129 Tím spíše, že jejich publikování by mohlo ohrozit úspěšnost a bezpečnost (dle Brina) použití nástroje.

130 Pokud jsme byli schopni ho provést. V přehledu uvádíme i ta řešení, která jsme neměli možnost testům podrobit, ale dle ohlasu v dostupné literatuře jsou považovány za významné.

a služby. Šlo nám o to, pokrýt celou škálu dostupných produktů zejména typově, nikoliv konkrétně. Přehled vybraných nástrojů je uveden v tabulce 7.

*Tabulka 7: Vybrané nástroje a služby a jejich základní charakteristika*

<b>Název nástroje/služby</b>	<b>Základní charakteristika</b>
CatchItFirst	komerční extrakorpální služba pro volný text přístupná přes WWW
Glatt Plagiarism Services	komerční intrinsic nástroj pro volný text na principu doplňování slov autorem
CopyCatch	komerční extrakorpální a intrinsic nástroj pro volný text
Eve 2	komerční extrakorpální nástroj pro volný text
Plagiarism Finder	komerční extrakorpální nástroj pro volný text
MyDropBox	komerční smíšená služba pro volný text přístupná přes WWW
TurnItIn.com	komerční smíšená služba pro volný text přístupná přes WWW
WCopyFind	volně dostupný (vč. zdrojových kódů) intrakorpální nástroj pro volný text a zdrojové kódy
Pl@giarism	volně dostupný (po registraci) intrakorpální nástroj pro volný text
Ferret	volně dostupný intrakorpální nástroj pro volný text a zdrojové kódy
CISE Tools	sada volně dostupných experimentálních intrakorpálních nástrojů pro volný text
JPlag	volně dostupná (po registraci a schválení) intrakorpální služba pro zdrojové kódy a volný text přístupná přes WWW
Sherlock	volně dostupný (vč. zdrojových kódů) intrakorpální nástroj pro zdrojové kódy a volný text

## 8.2 Komerční nástroje a služby

### 8.2.1 CatchItFirst

CatchItFirst ([catchitfirst]) je online služba kanadské společnosti Vancouver Software Labs. Původní služba nazvaná Scriptum byla placena na bázi časového předplatného (cca 50 dolarů za čtyřměsíční období pro jeden kurz do padesáti studentů), ale zřejmě byla již zrušena a nahrazena právě službou CatchItFirst, která je placena podle použití. Současná cena je 5 dolarů<sup>131</sup> za pět dokumentů nebo 8 dolarů za 10 dokumentů. Dokumenty zřejmě nejsou porovnávány mezi sebou, ale pouze vůči nějaké interní databázi či obsahu vrácenému internetovým vyhledávačem.



Obrázek 16: Prostředí služby CatchItFirst

Uživatelské rozhraní je velmi strohé, na placenou službu možná až příliš. K dispozici není ani nápočeda, ale u takto jednoduché funkcionality to snad ani nevadí. Jako velmi nevhodné se jeví to, že každý jednotlivý dokument je možné vložit pouze tak, že uživatel zkopíruje text do vstupního pole webového formuláře. Navíc dokument může mít nejvýše 20 tisíc znaků. Delší text služba úplně odmítne přjmout.

Zpracování probíhá automaticky po vložení každého dokumentu. Není možné nastavit žádné parametry, pouze zaslání upozornění na výsledek na registrovanou e-mailovou adresu. Nepříjemné je, že pro každý dokument pak přijde jeden takový dopis. Ten v sobě navíc obsahuje celý porovnávaný text a vedou z něho odkazy na případné nalezené stránky, odkud mohl být text zkopiovaný.

Zpracování je hotovo většinou do několika minut. Výsledky jsou ale poněkud rozporuplné. Testovací anglickou stránku připravenou pro podobné účely doktorkou Hannah Dee odhalil nástroj bravurně (dokonce neodkazoval tam, odkud jednotlivé části pocházejí, ale přímo na tento komplít na webu Dr. Dee). Jiný anglický text, který byl plagiátem (dva odstavce doslovně zkopiované

<sup>131</sup> Amerických, v kanadských je cena o něco vyšší.

z hesla Money v encyklopedii Wikipedia) ale nepoznal a místo toho malou částí tohoto textu odkázal na jakýsi pochybný<sup>132</sup> web plný reklam, kde se dotyčný text zcela jistě nenacházel. Jedna ze seminárních prací z kontrolního souboru A byla v pořádku nalezena na své originální adrese, ale jako plagiát byl označen pouze malý zlomek textu. Podobně dopadl i test obsahu Čapkova dramatu. Stránka s plným textem byla nalezena korektně, ale odkaz zahrnoval pouze dva krátké úseky.

Tabulka 8: Charakteristika služby *CatchItFirst*

<i>Název</i>	<i>CatchItFirst</i>	
<i>Výrobce/autor/provozovatel</i>	Vancouver Software Labs, Kanada	
<i>Adresa</i>	<a href="http://www.catchitfirst.com">http://www.catchitfirst.com</a>	
<i>Typ dokumentů</i>	<i>obsah</i>	volný text
	<i>jazyk</i>	libovolný (nespecifikováno, nezdá se, že by byly problémy s diakritikou)
	<i>formát</i>	pouze plaintext (vkládá se kopírováním)
<i>Způsoby detekce</i>	pouze extrakorpální vůči Internetu	
<i>Dostupnost</i>	<i>licence</i>	komerční služba předplácená na určitý počet zpracovaných dokumentů (5 za \$ 5, 10 za \$ 8); první tři dokumenty zdarma
	<i>připojení k síti</i>	nutné při vkládání a čtení výsledků
	<i>zpracování</i>	distribuované
	<i>použité metody</i>	informace nejsou dostupné, zřejmě hledání frází v interní databázi nebo internetovém vyhledávači
<i>Další</i>	<i>lokalizace</i>	ne (rozhraní anglicky)
	<i>prostředí</i>	internetový prohlížeč (univerzální)

Pokud služba v plném placeném režimu funguje stejně jako v demoverzi se třemi kredity zdarma (a není důvod se domnívat, že by fungovala jinak), nejedná se z našeho pohledu o příliš zdařilý nástroj. Může pomoci při kontrole volného textu ale vzhledem ke způsobu odesílání (a placení) je vhodný pouze pro malé skupiny. Rozporuplné výsledky přitom nepříliš odpovídají ceně. Pro používání je nutná registrace, po které v současnosti získá uživatel analýzu tří dokumentů zdarma<sup>133</sup>.

132 možná reklamní?

133 V praxi bylo z nějakého důvodu možné v tomto režimu jednou provést i čtyři porovnání.

### 8.2.2 Glatt Plagiarism Services

Společnost Glatt Plagiarism Services (o které již byla také řeč výše [Glatt1999]) poskytuje tři produkty pro boj s plagiátorstvím. Jedním z nich je výukový software, který má naučit studenty nedopouštět se plagiátorství. My se však zaměříme spíše až na nástroj pro detekci plagiátů nazvaný Glatt Plagiarism Screening Program. Ten pracuje na výše zmínovaném principu doplňování vynechaných slov autorem. Cena tohoto detekčního nástroje pro fakultu<sup>134</sup> je 300 dolarů (250 pokud si zákazník zakoupí také vzdělávací program za tutéž cenu) Zdarma je k dispozici také online verze testu, u které ale výrobce upozorňuje, že je výrazně méně přesná než dodávaný software<sup>135</sup>. Před každým provedením tohoto online testu je potřeba vyplnit e-mailovou adresu a vyřešit několik jednoduchých úkolů<sup>136</sup> (sčítání, násobení, přesmyčky). Vzhledem ke koncepci tohoto nástroje jej není možné vyzkoušet jinak, než při reálném nasazení. Výrobce také uvádí, že za 65 dolarů je možné zakoupit offline verzi tohoto testu pro Windows.

*Tabulka 9: Charakteristika nástroje Glatt Plagiarism Screening Program*

<b>Název</b>		<b>Glatt Plagiarism Screening Program</b>
<b>Výrobce/autor/provozovatel</b>		Glatt Plagiarism Services, Inc., USA
<b>Adresa</b>		<a href="http://www.plagiarism.com/screening.htm">http://www.plagiarism.com/screening.htm</a>
<b>Typ dokumentů</b>	<b>obsah</b>	volný text
	<b>jazyk</b>	libovolný (nespecifikováno, nezdá se, že by byly problémy s diakritikou)
	<b>formát</b>	není známo (online verze pouze plaintext)
<b>Způsoby detekce</b>		intrinsic na základě doplňování autorem
<b>Dostupnost</b>	<b>licence</b>	komerční software cena licence \$ 300, offline verze testu \$ 65, demonstrační verze pouze v podobě online testu
	<b>připojení k síti</b>	zřejmě ne
	<b>zpracování</b>	zřejmě lokální

134 Jak uvádí výrobce. Není přitom zcela jisté zda se jedná o distribuovaný systém či multilicenci (spíše to druhé).

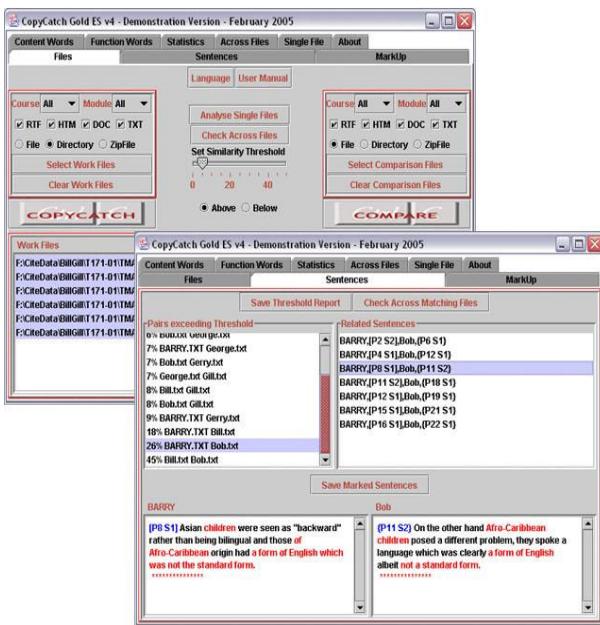
135 Těžko to můžeme posoudit, protože demo verze dodávaného programu není k dispozici. Jediný rozdíl by snad mohl být v tom, že volná verze pracuje pouze s několika prvními větami, respektive s prvními deseti vynechanými slovy tj. cca s padesáti až padesáti pěti slovy.

136 Možná je to jeden ze způsobů jak přimět uživatele aby nepoužívali toto demo v ostrém provozu, ale také by to mohl být jakýsi psychologický trik aby se ten, kdo bude následně doplňovat slova do svého dokumentu, uvolnil. Není jasné, zda plná verze programu obsahuje také podobné testy. V online verzi jsou tyto testy stále stejně.

	<b>použité metody</b>	autor doplňuje každé páté vynechané slovo, na základě toho a dalších (utajených) parametrů se vypočítá skóre
<b>Další</b>	<b>lokalizace</b>	ne (rozhraní anglicky)
	<b>prostředí</b>	Windows

### 8.2.3 CopyCatch

Produkt CopyCatch Gold ([copycatch]) od britské společnosti CFL Software Development je další komerční nástroj pro detekci plagiátů. Funguje ve dvou možných režimech. V prvním porovnává klasicky dokumenty v korpusu, druhá možnost je práce pouze s jedním dokumentem (čili intrinsic režim). Poradí si i s dokumenty ve formátech RTF, HTML a DOC. Může pracovat se základní jednotkou na bázi slova ale také celé fráze či věty. K dispozici je také novější verze CopyCatch Web, která umožňuje také asistovat při vyhledávání možných zdrojů plagiátů na webu. Toto vyhledávání nevyužívá žádnou vlastní databázi, ale pracuje s rozhraním Google API.



Obrázek 17: Prostředí nástroje CopyCatch Gold

Ceny se pohybují podle toho, jestli se jedná o licenci pro celou školu (podle počtu studentů) nebo pouze některou část. Verze Web stojí 250 liber. Ač samotný program ani webová prezentace výrobce nevypadá zrovna nejmoderněji, systém je zřejmě stále aktivně udržován a poslední úpravy byly provedeny v roce 2006.

Tabulka 10: Charakteristika nástroje CopyCatch Gold

<i>Název</i>		<i>CopyCatch Gold/Web</i>
<i>Výrobce/autor/provozovatel</i>		CFL Software Development, UK
<i>Adresa</i>		<a href="http://www.copycatchgold.com/">http://www.copycatchgold.com/</a>
<i>Typ dokumentů</i>	<i>obsah</i>	volný text
	<i>jazyk</i>	angličtina, francouzština, holandština, němčina a některé další
	<i>formát</i>	plaintext, RFT, HTML, DOC
<i>Způsoby detekce</i>		intrakorpální s možností rozdelení korpusu, intrinsic na základě stylometrie, asistované vyhledávání v Google u verze Web)
<i>Dostupnost</i>	<i>licence</i>	komerční software cena licence £ 100–250 za rok, případně 5 penny za studenta a rok, demonstrační verze není k dispozici
	<i>připojení k síti</i>	ne (pouze při vyhledávání ve verzi Web)
	<i>zpracování</i>	lokální
	<i>použité metody</i>	informace nejsou dostupné, zřejmě porovnání na základě slov či vět, stylometrie
<i>Další</i>	<i>lokalizace</i>	ne (rozhraní anglicky a v některých dalších jazycích)
	<i>prostředí</i>	Java

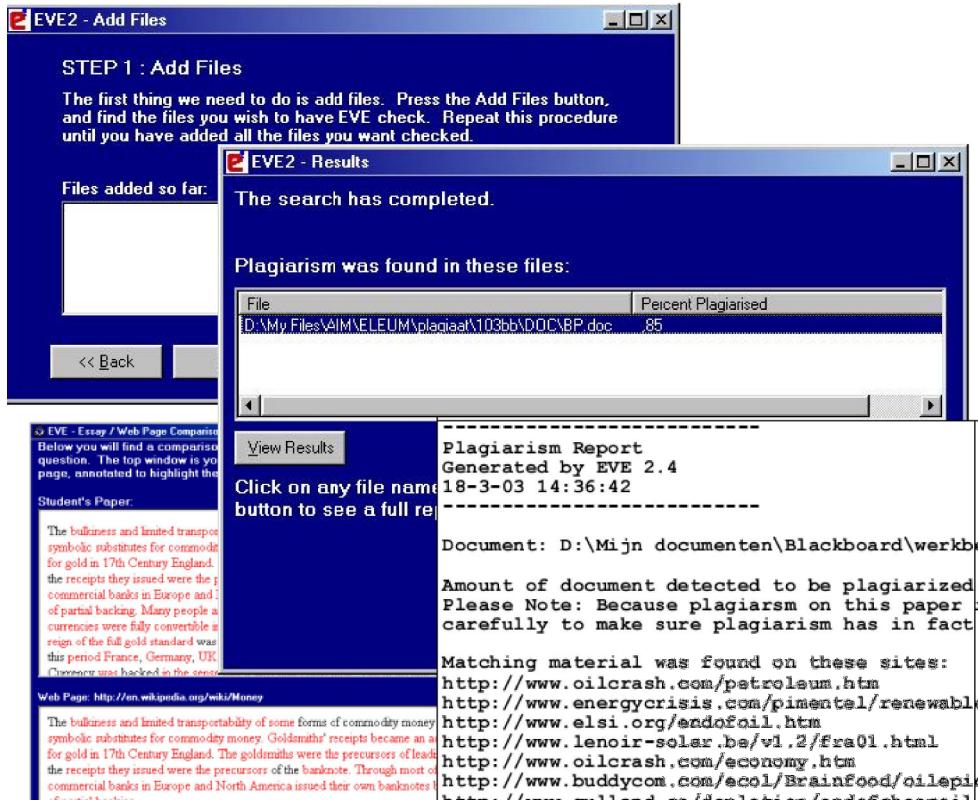
#### 8.2.4 Eve 2

Kanadská společnost CaNexus je tvůrcem dalšího komerčního nástroje nazvaného tentokrát Eve2 (Essay Verification Engine [eve2]) aktuálně ve verzi 2.5. Pracuje pouze v extrakorpálním režimu a poskytnuté dokumenty vyhledává na internetu. Výsledkem pro každý zpracovávaný dokument je nový RTF dokument obsahující míru možného plagiátorství a adresy, ze kterých mohlo být kopirováno. Jednorázová cena produktu je necelých 30 dolarů, neplatí se žádné roční poplatky. Demo k aktuální verzi není k dispozici, ale je poskytována desetidenní záruka vrácení peněz. Starší verze 2.4 lze ale najít na internetu<sup>137</sup> jako shareware s patnáctidenní lhůtou k vyzkoušení.

Případné detailní informace a zde prezentované výsledky jsou tedy založeny právě na této verzi a mohou se v novější verzi lišit. Nástroj si každopádně poradí i se soubory ve formátu DOC a WPD

137 Například na adrese <http://www.tucows.com/preview/293122>

(WordPerfect). Jak se uvádí v recenzi [ARTS2003] této starší verze tohoto produktu (2.4) je Eve2 poměrně náročný jak na výpočetní výkon, tak na rychlosť připojení k internetu. Délka zpracování jednoho dokumentu (o délce 2500 slov) je uváděna v řádu několika minut.



Obrázek 18: Prostředí nástroje EVE2

Prakticky je skutečně prováděno několik desítek až několik stovek vyhledávacích dotazů na jeden dokument (záleží na jeho délce). Podle analýzy síťového provozu se zdá, že vyhledávání probíhá výhradně na serveru search.lycos.com a navíc písmena s českou diakritikou jsou považována za odělovač slov. V testech si vedl poměrně dobře na anglických dokumentech, ale ty česky psané nebyl schopen detektovat vůbec. Na vině bude zřejmě jednak použitý vyhledávač, který není pro české prostředí nevhodnější a zejména to, že si nástroj neporadí s diakritikou.

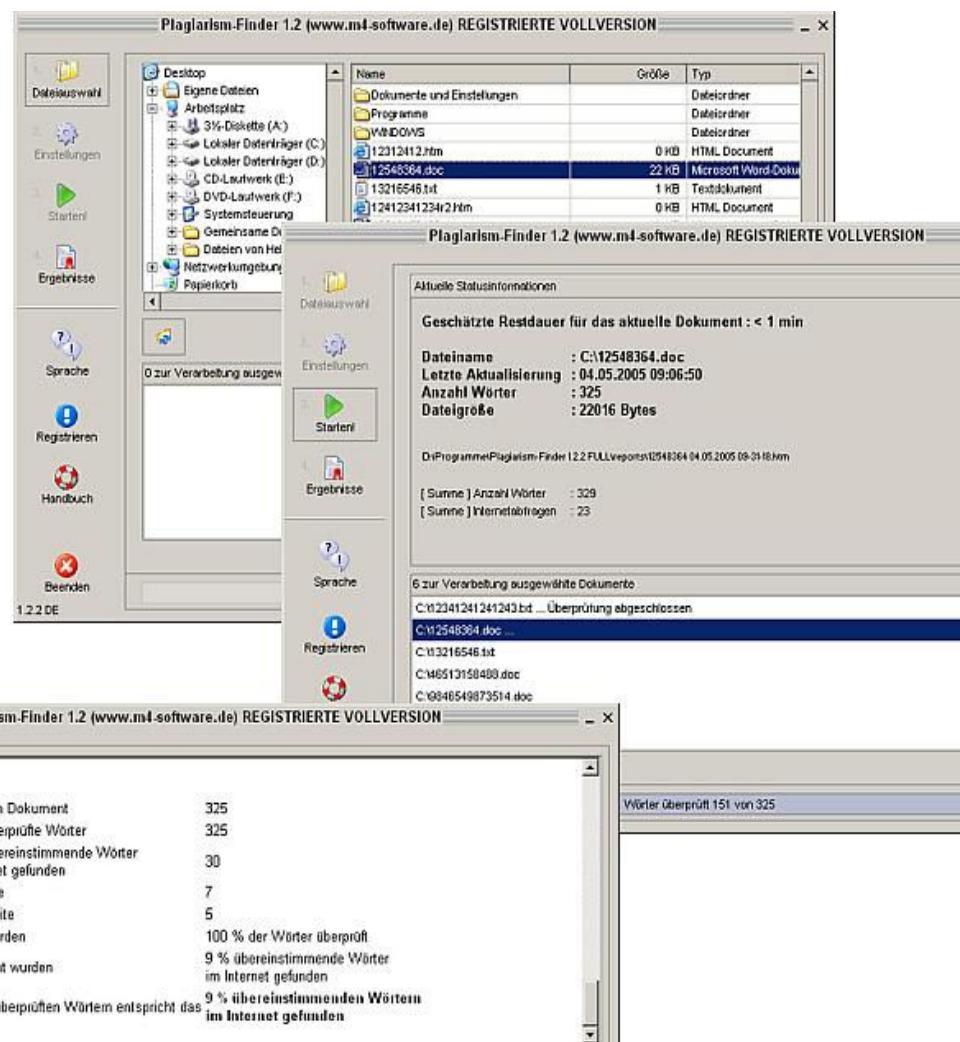
Z uživatelského hlediska je velmi nepříjemný fakt, že soubory musí být přidávány pouze po jednom. Nelze dokonce ani použít vícenásobný výběr pomocí ctrl. Všechny dokumenty jsou porovnávány najednou a výsledky jsou dostupné až po kompletním dokončení. Navíc lze po prohlédnutí výsledků program pouze zavřít a pro další porovnání je nutné jej znova spouštět. V detailu lze zobrazit také porovnání testovaného dokumentu a potenciálního zdroje s vyznačenými shodnými částmi. Nástroj zřejmě z nějakého důvodu vyžaduje možnost zápisu do analyzovaných souborů (nebo do stejného adresáře) při analýze stejných dokumentů ale na CD hlásil chyby a nepracoval.

Tabulka 11: Charakteristika nástroje EVE2

<i>Název</i>	<i>EVE2</i>
<i>Výrobce/autor/provozovatel</i>	CaNexus, Kanada
<i>Adresa</i>	<a href="http://www.canexus.com">http://www.canexus.com</a>
<i>Typ dokumentů</i>	<p><i>obsah</i> volný text</p> <p><i>jazyk</i> nespecifikováno (čeština, zdá se, není podporována)</p> <p><i>formát</i> plaintext, DOC, WPD</p>
<i>Způsoby detekce</i>	pouze extrakorpální
<i>Dostupnost</i>	<p><i>licence</i> komerční software, cena licence \$ 29,99, demo verze není k dispozici (10 denní záruka vrácení peněz)</p> <p><i>připojení k síti</i> ano vyžadováno (poměrně rychlé)</p> <p><i>zpracování</i> lokální</p> <p><i>použité metody</i> informace nejsou dostupné, zřejmě nějaká forma analýzy dokumentu a na jejím základě vyhledávání na serveru search.lycos.com, následně stažení kandidátů na zdroj a porovnání</p>
<i>Další</i>	<p><i>lokalizace</i> ne (s českou diakritikou má zřejmě problémy) rozhraní v angličtině</p> <p><i>prostředí</i> Windows</p>

### 8.2.5 Plagiarism Finder

Plagiarism Finder ([plagiarismfinder]) je produktem německé společnosti Mediaphor Software Entertainment AG. Podobně jako EVE2 pracuje pouze v extrakorpálním módu a vyhledává potenciální zdrojové dokumenty na internetu pomocí standardního vyhledávače. Také v tomto případě je vyhledávání náročné na výkon a připojení k Internetu, i když subjektivně se zdá, že o něco méně než v předchozím případě. Analýzou síťového provozu jsme zjistili, že vyhledávání probíhá výhradně na serveru ask.com (případně jeho německé mutaci de.ask.com). Je to tentýž, který používá k vyhledávání portál lycos.com užívaný nástrojem popisovaným výše. Uživatelské rozhraní je však rozhodně mnohem lepší než u EVE2.



Obrázek 19: Prostředí nástroje Plagiarism Finder

Kromě čistě textových je možné zpracovat i soubory ve formátech PDF, DOC, RTF a HTML. Cena licence je uváděna<sup>138</sup> 98 € včetně DPH za instalaci na jeden počítač. V případě nákupu více licencí jsou poskytovány slevy až na 68 € v případě pořízení deseti a více. Alternativou je „přenosná“ verze za 169 €, která je dodávána na USB disku, nevyžaduje instalaci a lze ji používat na více počítačích (ovšem ne zároveň). K dispozici je třicetidenní demoverze s některými omezeními. V ní je porovnáváno pouze prvních 5000 znaků a není možné přiřadit k porovnání více dokumentů najednou.

Zarázející ale bylo to, že při výchozím nastavení nenalezl tento nástroj té měř žádné zdroje plagiátů. A to platilo jak pro české, tak pro anglické dokumenty. Teprve při volbě detailního (a mnohem pomalejšího a náročnějšího) prohledávání nalezl zlomek některých zdrojů (např. shoda dokumentu doktorky Dee byla pouze 17 %). Písmena s diakritikou, zdá se, převádí správně, takže je možné, že vyhledávání je silně orientováno na německé prostředí. Někdy sice našel potenciální zdroje, ale odkazované stránky již bud' vůbec neexistovaly, nebo neměly s testovaným dokumentem ani jeho tématem vůbec žádnou souvislost.

138 V online ceníku se tato cena objevuje u verze 1.0, současná verze 1.3 je však na jiném místě poskytována za stejnou cenu.

Tabulka 12: Charakteristika nástroje Plagiarism Finder

<i>Název</i>		<i>Plagiarism Finder</i>
<i>Výrobce/autor/provozovatel</i>		Mediaphor Software Entertainment AG
<i>Adresa</i>		<a href="http://www.m4-software.de/">http://www.m4-software.de/</a>
<i>Typ dokumentů</i>	<i>obsah</i>	volný text
	<i>jazyk</i>	nespecifikováno (němčina a angličtina jistě)
	<i>formát</i>	plaintext, PDF, DOC, RTF, HTML
<i>Způsoby detekce</i>		pouze extrakorpální
<i>Dostupnost</i>	<i>licence</i>	komerční software, cena licence 98 € vč. daně v případě nákupu více licencí sleva, licence vázána na konkrétní počítač, možnost „přenosné“ licence za 168 € vč. daně a USB disku, dostupná 30 denní demonstrační verze
	<i>připojení k síti</i>	ano vyžadováno (poměrně rychlé)
	<i>zpracování</i>	lokální
	<i>použité metody</i>	detailní informace nejsou dostupné, zřejmě nějaká forma analýzy dokumentu a na jejím základě vyhledávání na serveru ask.com a následné porovnání s
<i>Další</i>	<i>lokalizace</i>	ne (s českou diakritikou zřejmě nejsou problémy), rozhraní v němčině a angličtině
	<i>prostředí</i>	Windows

### 8.2.6 MyDropBox

Dvojkou na aktuálním trhu služeb pro detekci plagiátů je služba poskytovaná pod názvem MyDropBox ([mydropbox]). Historie okolo tohoto systému je poměrně košatá a zajímavá. Stejná technologie která se zde používá a zřejmě i stejní autoři stáli již za několika službami pro odhalování plagiátů. Někdy v roce 2001<sup>139</sup> stáli Olexiy Shevchenko, Max Litvin, a Sasha Lugovskyy (dle [ARTS2003]) za vznikem zdarma poskytované služby pro odhalování plagiátů nazvané plagiserve.com. Stejná technologie a stejní lidé stáli za placenou službou nazvanou edutie.com. Mezičítim se objevily

<sup>139</sup> Téměř veškerá datace v tomto odstavci je založena na analýze výpisu historie obsahu příslušných domén ve volně dostupném internetovém archivu archive.org.

spekulace (šířené zřejmě konkurencí z řad TurnItIn.com) o tom, že stejní lidé, kteří stáli za těmito službami, se angažovali i v některých webech, které slouží k prodeji vypracovaných dokumentů studentům ([Young2002])<sup>140</sup>. Na konci roku 2003 byly obě tyto služby zrušeny a návštěvníci byli přesměrováni na novou službu založenou na stejné technologii právě současně fungující mydropbox.com. Tu v květnu roku 2005 zakoupila i s jejím provozovatelem MyDropBox LLC kanadská investiční společnost Sciworth Inc.

MyDropBox je komplexní systém pro správu kurzů, který zahrnuje jak vyučující, tak asistenty a studenty. Kromě managementu kurzů a úkolů umožňuje tento systém také asistovat při opravování prací a známkování. Kompletně integrovanou součástí je část nazvaná SafeAssignment, která se stará o detekci plagiátů a poskytuje zásadní přidanou hodnotu této služby. K dispozici je také možnost integrovat tento systém do školských informačních systémů od výrobce Blackboard Inc.

Provozovatel standardně nenabízí zkušební lhůtu, organizace se mohou zúčastnit zkušebního Pilot programu, který jim umožní zjistit, které služby jsou pro ně vhodné. Díky laskavosti obchodního oddělení<sup>141</sup> nám byl bezplatně poskytnut týdenní účet pro důkladné otestování. Ceny pro organizace závisí zejména na počtu studentů. Dle dostupných informací se cena pohybuje kolem \$ 0,60 za studenta a rok. Pro organizace, které mají více než 5000 studentů, jsou poskytovány slevy (od pěti až do čtyřiceti procent). Další možností je individuální licence za \$ 90 ročně v tomto případě je poskytována čtrnáctidenní záruka vrácení peněz.

Protože systém je opravdu poměrně komplexní, zaměříme se zde hlavně na jeho část pro detekci plagiátů. Kromě prostého textu pracuje systém s dokumenty ve formátech DOC, RTF, PDF a HTML. Je možné jich nahrávat i více najednou v komprimovaném souboru ZIP, ale pouze pokud nejsou přiřazovány konkrétním studentům (a tedy nemohou být známkovány a podobně). Standardní postup je ten, že studenti přímo osobně odevzdávají (nahrávají) své práce do příslušné části systému MyDropBox. Alternativně může vyučující nahrávat dokumenty po jednom a přiřazovat je k jednotlivým studentům.

Nástroj pracuje jak extrakorpálně, tak intrakorpálně, čili bychom ho mohli označit jako smíšený. Porovnávání je prováděno jak mezi odevzdanými dokumenty navzájem, tak i vůči internetovým zdrojům. Dle informací provozovatele jde o použití indexu vyhledávače MSN od Microsoftu, dále by měly být prohledávány články z databáze ProQuest, články dostupné přes službu FindArticles a vlastní seznam abstraktů prací nabízených za poplatek (paper mills). Navíc jsou do porovnání zařazeny také všechny dokumenty odeslané v rámci téže instituce. Nepracuje se tak se všemi odeslanými dokumenty, ale kvůli ochraně soukromí je databáze rozdělena podle jednotlivých organizací.

Práce s českou diakritikou v této službě byla střídavě úspěšná. Některé nahrané dokumenty byly převedeny a zpracovány bez problémů, u jiných (včetně plaintextu) se vyskytly potíže. Při testování plaintextových souborů v různých znakových sadách úspěšně prošel pouze ten v UTF-8. Při vkládání prostého textu klasickým zkopirováním bylo vše bez problémů. Toto může být v českém prostředí poměrně nepřijemné a je třeba brát to v úvahu, ale nebrání to přímo použití.

140 Takový konflikt zájmů by mohl být dotyčným ku prospěchu hned z několika důvodů. Zaprvé by testované (a tedy odesílané) dokumenty mohly sloužit jako snadný zdroj poměrně kvalitních esejí k prodeji. Navíc detekce plagiátů by mohla být upravena tak, aby neupozorňovala na dokumenty koupené od příslušných spřízněných prodejců a zvyšovala tak jejich atraktivitu. Žádný z těchto scénářů však dle našich informací nebyl nikdy prokázán.

141 konkrétně paní Anna Yashkina, které tímto ještě jednou děkujeme

Porovnání samotné se spouští zřejmě automaticky po nahrání dokumentu. První výsledky pro kratší dokumenty byly v našich testech zpracovány většinou do pěti, až deseti minut u delších to může trvat déle. Nelze to však přesně odhadnout a konkrétní časy jsou proměnlivé. V případě individuální licence je uváděna garantovaná doba 12 hodin.

Porovnání probíhá nejspíše na základě vět. Věty nemusejí být zcela stejné a lze tedy usuzovat na to, že se pracuje i s jednotlivými slovy v nich. Podobnost každých nalezených vět je ohodnocena pomocí Sentence Matching Score (SMC). Celkové skóre pro dokument je potom vážený průměr jednotlivých SMC. Váha je dána jednak délkom věty a také tím, jak obvykle se vyskytuje.

V celkových výsledcích se objevuje u každého dokumentu právě toto číslo. Detailní zobrazení ukazuje v horní části barevně rozlišený seznam internetových adres či ostatních dokumentů, se kterými se daný dokument shoduje. Odpovídající barvou jsou potom vyznačeny příslušné věty také v textu. Případné korektně označené citace (které ale služba automaticky nerozpoznává) je možné označit a zkontrolovat daný dokument znovu bez jejich započítání.

Jednou z možností je také zpřístupnění výsledků porovnání studentovi po odevzdání, aby mohl zkontrolovat, zda je jeho práce systémem vyhodnocena jako originální a případně aby ji mohl vhodně upravit. To může sloužit jako účinná prevence a zároveň posílí důvěru studentů v takový systém. Samozřejmě tím může být částečně narušena bezpečnost detekce. Zřejmě i proto se jedná pouze o volitelnou možnost.

The screenshot shows the MyDropBox.com administrator homepage. On the left, there's a sidebar with 'Statistics' (Number of students: 0, Number of instructors: 0, Number of courses: 0), 'Latest Announcements', and a 'SafeAssignment Report' section. The main area has tabs for 'Submit Paper' and 'Info'. In the 'Submit Paper' tab, a student named 'Hauzirek Michal' is selected, and the title is 'Možnosti automatické detekce plagiatu'. Under 'Submission Options', there are two radio buttons: 'Upload File' (selected) and 'Copy/Paste Document'. A file input field shows 'C:\temp\plag\dipites' and a 'Choose...' button. Below this is a large text area for pasting the document. A 'Submit' button is at the bottom. In the bottom right corner, there's a 'Private Quick Submit for Hause' section showing a list of submitted papers:

Filename	File	Matching	SA Report	Submitted
IC-D001.txt	g	-	-	Apr 06 2007 07:43:44 EDT
IC-D002.txt	g	-	-	Apr 06 2007 07:43:45 EDT
IC-D003.txt	g	-	-	Apr 06 2007 07:43:46 EDT
IC-D004.txt	g	-	-	Apr 06 2007 07:43:47 EDT
IC-D005.txt	g	-	-	Apr 06 2007 07:43:48 EDT
IC-D006.txt	g	-	-	Apr 06 2007 07:43:49 EDT

Obrázek 20: Prostředí služby MyDropBox.com

Výsledky vyhledávání a porovnání byly ale v našich testech poněkud rozporuplné. Některé i české zdroje byly řádně nalezeny i na českých webech, ale nebylo to zdaleka pravidlem. Někdy však byla označena pouze jediná věta, přestože z daného zdroje pocházel celý odstavec nebo i několik odstavců. Někdy byly za relevantní věty považovány i velmi krátké úseky textu (například datum) a odkazovaný zdroj tak neměl s původním dokumentem téměř nic společného (kromě toho, že např. v tomto případě obsahoval stejné datum). Naproti tomu však dva téměř totožné české dokumenty někdy označil jako shodné z pouhých několika procent a naprosto shodné věty měly SMC pouze kolem šedesáti až sedmdesáti procent. Testovací anglické dokumenty dopadly o něco lépe, zejména ty pro testování extrakorpálních schopností. Jak kombinovanou „ukradenou“ stránku od doktorky Dee, tak vzorový text o Napoleonovi z dema služby TurnItIn spolehlivě detekoval s velmi vysokým skóre. Ale náš anglicky psaný text vytvořený z několika zdrojových dokumentů přístupných přes databázi ProQuest nebyl detekován příliš úspěšně. Malá část čítající několik vět byla nalezena, ale vzhledem k tomu, že v něm nebyl ani odstavec originální, čekali bychom rozhodně více.

<i>Název</i>		<i>MyDropBox.com</i>
<i>Výrobce/autor/provozovatel</i>		Sciworth, Inc., Kanada
<i>Adresa</i>		<a href="http://www.mydropbox.com">http://www.mydropbox.com</a>
<i>Typ dokumentů</i>	<i>obsah</i>	volný text
	<i>jazyk</i>	většina evropských jazyků (jmenovitě nejméně angličtina, albánština, baskičtina, katalánština, dánština, holandština, estonština, francouzština, finština, němčina, irština, italština, norština, portugalština, španělština, švédština); nové jazyky je možné doplnit; s češtinou jsou problémy pouze při nahrávání některých dokumentů (zejm. pokud nejsou v UTF-8)
	<i>formát</i>	plaintext, PDF, RTF, DOC, HTML, ZIP
<i>Způsoby detekce</i>		smíšený – intrakorpální i extrakorpální (Internet, vlastní databáze, některé komerční databáze textů)
<i>Dostupnost</i>	<i>licence</i>	komerční služba placená ročně pro instituci, standardní cena \$ 0,60 za studenta na rok, pro instituce nad 5000 studentů slevy (5–40 %); pro instituce možnost speciální testovací licence až na 150 dní
	<i>připojení k síti</i>	vyžadováno
	<i>zpracování</i>	distribuované

	<b>použité metody</b>	informace nejsou známy; zřejmě porovnání dokumentů na základě shodných či podobných vět, využití indexu vyhledávače MSN
<b>Další</b>	<b>lokalizace</b>	ne, rozhraní v několika jazycích, s českou diakritikou jsou někdy problémy při nahrávání dokumentů
	<b>prostředí</b>	internetový prohlížeč (univerzální)

### 8.2.7 TurnItIn

Služba TurnItIn.com ([turnitin]) od společnosti iParadigms, LLC je v současnosti s poměrně velkým náskokem zřejmě leaderem na trhu detekce plagiátů. U vzniku této společnosti stalo kolem roku 1996 několik vědců z University of California v Berkley. Online služba plagiarism.org pro detekci plagiátů začala být provozována na přelomu let 1998 a 1999<sup>142</sup> a poměrně brzy získala i značný mediální ohlas<sup>143</sup>. Okolo roku 2000 vznikla služba turnitin.com.

V současnosti její služby údajně využívá několik tisíc institucí v několika desítkách zemí. Samozřejmě kromě detekce plagiátů, která stála u zrodu této služby a je její zásadní součástí, nabízí v současnosti také sadu integrovaných nástrojů pro správu kurzů, dále rovněž poskytuje nástroje pro asistenci při opravování odevzdaných elektronických dokumentů a to včetně možnosti kolaborativního opravování více vyučujícími a správu známek studentů. Základní nabídka funkcí je tedy shodná jako u konkurence MyDropBox. Také základní filosofie je stejná, studenti by měli sami odevzdávat dokumenty do příslušných kurzů a ty jsou následně otestovány a připraveny pro vyučujícího ke zpracování. I zde je dostupné velké množství tutoriálů a animovaných průvodců pro různé typy uživatelů. Rovněž je možné integrovat jak službu pro detekci plagiátů, tak i známkovací nástroje do některých dalších systémů pro správu kurzů.

Provozovatel standardně neposkytuje možnost vyzkoušení. Nereagoval ani žádost o poskytnutí informací o cenové a licenční politice, takže zde uváděné údaje je třeba brát s rezervou jako neoficiální informace, které se ale přesto s vysokou pravděpodobností budou blížit skutečnému stavu. Pro testování je dispozici kromě animovaných manuálů také velmi stručné demo, které ale nemá žádnou reálnou funkčnost a nabízí pouze náhled použitého rozhraní. To se nám zdálo subjektivně poněkud méně přívětivé než u menšího konkurenta. Obsahuje ale naproti tomu některé užitečné funkce, kterými jej převyšuje.

Příkladem může být zobrazení detailu obou podobných dokumentů vedle sebe včetně provázání shodných částí odkazy. Zajímavou možností je také schopnost vyřadit z porovnávání části textu uzavřené v uvozovkách a dokumenty uvedené v bibliografii. Obojí by mělo být detekováno automaticky. To by mělo zaručit, že z porovnání budou vyloženy korektně citované dokumenty. Bohužel jak tato jistě pozoruhodná vlastnost funguje v praxi<sup>144</sup>, nebylo možné zjistit. Dokumenty je možné nahrávat ve formátech čistého textu, DOC, PDF, RTF, WPD a HTML.

142 Vzhledem k nedostupnosti přesnějších informací je téměř veškerá datace v tomto odstavci založena na analýze výpisu historie obsahu příslušných domén ve volně dostupném internetovém archivu archive.org.

143 Podle informací provozovatele. Není jisté, nakolik se o něj sami zasloužili aktivním marketingem.

144 Včetně takových zajímavostí, jako jaké formáty bibliografických záznamů rozeznává a jak si poradí s neanglickými formáty uvozovek (např. české spodní a horní, francouzské, ...) a podobně.

The screenshot shows the Turnitin Originality Report interface. At the top, it displays the Overall Similarity Index as 88%. Below this, the text "In Corsica in 1769 Napoleon Bonaparte rose through the midst of the chaos of the French Revolution to become Emperor of the French. He is regarded by many as a military genius, by others as an opportunist. Perhaps he was both." is analyzed. A modal dialog box is overlaid, titled "Submit a paper by:" with "file upload" selected. It includes fields for "author" (Shah), "first name" (Nilay), "last names" (Shah), and "submission title". Below the dialog, a list of assignments is shown, each with a checkbox, author, title, report, and grade. One assignment by Shah, Nilay has a grade of 98%.

Obrázek 21: Prostředí služby TurnItIn.com

Detekce plagiátů probíhá samozřejmě jak intrakorpálně, tak extrakorpálně. Ve vlastní databázi jsou indexovány jak dokumenty volně dostupné na internetu, tak také články a materiály publikované v časopisech a novinách a díky spolupráci s některými vydavatelstvími i obsahy knih a učebnic. Samozřejmostí jsou také dokumenty ze známých<sup>145</sup> paper mills. Její součástí jsou také veškeré práce odevzdané prostřednictvím této služby. Bližší informace o použité technologii nejsou veřejně k dispozici.

Porovnání probíhá kompletně vůči všem materiálům, tedy i těm, které byly odevzdány v rámci jiných institucí. Toto je zásadní rozdíl oproti konkurenční službě, která porovnává odevzdané dokumenty pouze v rámci dané instituce. Za toto lehce kontroverzní nakládání s cizími dokumenty si získala tato služba poměrně dost odpůrců z řad studentů i některých vyučujících (viz např. [Glod2006]). Objevilo se i několik žalob a to jak z řad studentů na provozovatele služby tak i v opačném gardu.

Jak již bylo uvedeno výše, cenová politika není obecně známa a společnost poskytuje cenové nabídky pouze pro předběžné objednávky. Podařilo se nám ale z několika nezávislých zdrojů zjistit, jaké mohou být přibližné cenové relace.

<sup>145</sup> zejména amerických.

Profesorka Rebecca Moore Howard ze Syracuse University na svém osobním webu [Howard2006] uvádí, že roční cena za studenta se pohybuje okolo \$ 2,50. Z diskuse tamtéž vyplývá, že toto je cena pro kompletní balík obsahující kromě detekce plagiátů i nástroje pro hodnocení a známkování (balík služeb nazvaný TurnItIn Suite). Cena pouze za detekci plagiátů (modul Plagiarism Prevention) by se měla pohybovat okolo \$ 0,87 za studenta (březen 2007) s předpokládaným růstem do 10 % za rok.

Podobná čísla potvrzuje také druhý zdroj (z února 2006) [Aintegrity2006]. Z tam obsažené oficiální cenové nabídky plyne, že cena se skládá z několika součástí. První je fixní roční poplatek \$800 za používání. K němu se přičítá \$ 0,80 za každého studenta v případě Plagiarism Prevention balíku nebo \$ 1,50 (minimálně však \$ 2300) za kompletní TurnItIn Suite.

Další ze zdrojů ([Dye2006]) uvádí, že University of Kansas (která má v současnosti asi 26 000 studentů<sup>146</sup>) nehodlala koncem roku 2006 prodloužit smlouvu na využívání této služby. Hlavním důvodem měly být právě neustále se zvyšující licenční poplatky. Jak podle vyjádření jednoho z představitelů univerzity rostly ceny za roční používání, uvádí následující tabulka. Dále se tam uvádí, že cena 22 tisíc dolarů je téměř o čtyřicet procent nižší, než platí University of Arizona (s téměř 37 tisíci studentů<sup>147</sup>). Pokud předpokládáme fixní cenu a rozpočítáme zbytek na uváděný počet studentů, získáme opět číslo blízké osmdesáti centům za studenta. Každopádně je zřejmé, že cena služby je minimálně přibližně o třetinu vyšší než u konkurenční MyDropBox. Přesto se zdá, že je v současnosti mnohem využívanější.

*Tabulka 13: Vývoj ročních nákladů na službu TurnItIn.com pro University of Kansas (podle [Dye2006])*

<i>Rok</i>	<i>Cena</i>
2003	\$ 6 000
2004	\$10 000
2005	\$ 14 000
2006	\$ 22 000

*Tabulka 14: Charakteristiky služby TurnItIn.com*

<i>Název</i>		<i>TurnItIn.com</i>
<i>Výrobce/autor/provozovatel</i>		iParadigms, LLC, USA
<i>Adresa</i>		<a href="http://www.turnitin.com">http://www.turnitin.com</a>
<i>Typ dokumentů</i>	<i>obsah</i>	volný text

146 dle [http://en.wikipedia.org/wiki/University\\_of\\_Kansas](http://en.wikipedia.org/wiki/University_of_Kansas)

147 dle [http://en.wikipedia.org/wiki/University\\_of\\_Arizona](http://en.wikipedia.org/wiki/University_of_Arizona)

	<i>jazyk</i>	nespecifikováno, pravděpodobně více (češtinu nebylo možné otestovat)
	<i>formát</i>	plaintext, DOC, RTF, PS, PDF, WPD, HTML
<b>Způsoby detekce</b>		smíšený – intrakorpální i extrakorpální (Internet, vlastní databáze, některé komerční databáze textů, knihy a časopisy)
<b>Dostupnost</b>	<i>licence</i>	komerční služba placená ročně pro instituci, odhad aktuální ceny služby detekce plagiátů cca \$ 0,80–0,90 za studenta na rok + \$ 800 fixně, ceny jsou individuální a zřejmě zahrnují slevy pro velké zákazníky; zkušební verze zřejmě po domluvě pro potenciální velké zákazníky
	<i>připojení k síti</i>	ano vyžadováno
	<i>zpracování</i>	distribuované
	<i>použité metody</i>	informace nejsou známy, zřejmě porovnání dokumentů vůči vlastní databázi včetně porovnání napříč institucemi
<b>Další</b>	<i>lokalizace</i>	ne, rozhraní zřejmě v několika jazycích?, zpracování českého textu nebylo možno otestovat
	<i>prostředí</i>	internetový prohlížeč (IE 6+, Firefox 1.5+, Safari) <sup>148</sup>

## 8.3 Volně dostupné nástroje a služby

### 8.3.1 WCOPYFIND

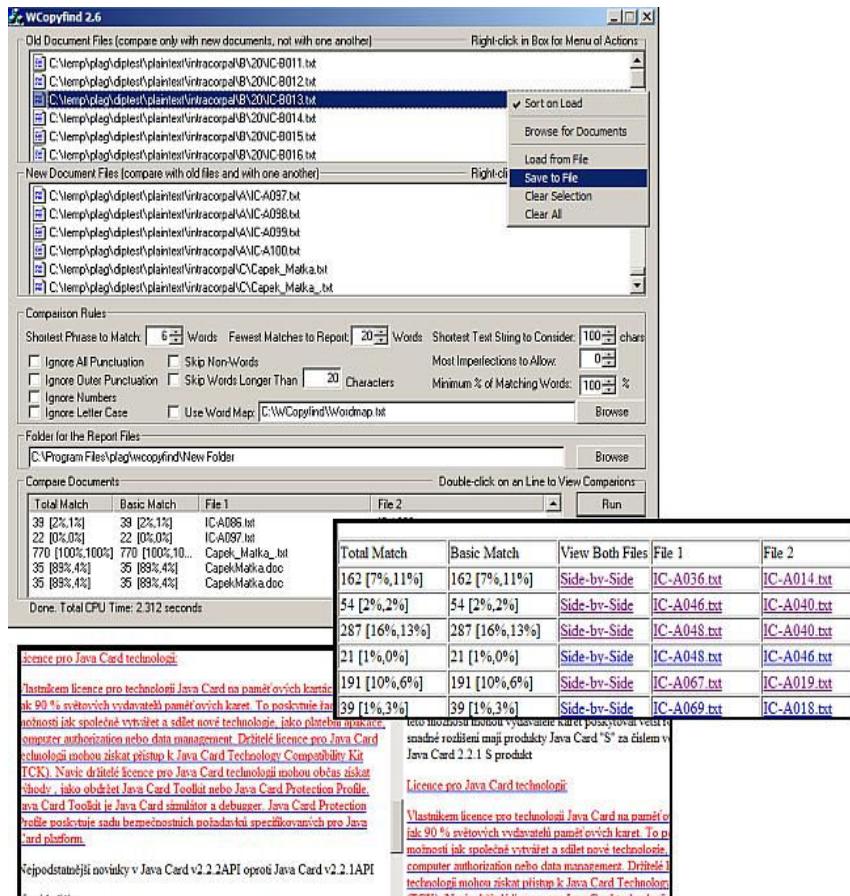
WCopyFind ([wcopyfind]) je nástroj vytvořený Lou Bloomfieldem, profesorem fyziky na University of Virginia. Slouží k porovnávání dokumentů mezi sebou a vyhledávání shodných částí. Je překvapivě velice rychlý a s výchozím nastavením (které je poměrně dobré pro detekci) zvládne porovnání více než stovky kratších dokumentů navzájem každý s každým za pár vteřin. Výsledky jsou prezentovány ve formě HTML. Nástroj pracuje s prostým textem a údajně by měl zvládat také DOC formát. V praxi však tento převod nebyl vůbec spolehlivý a výsledkem byly pouze zlomky původního dokumentu<sup>149</sup>. Je také možné vložit odkazy na online HTML dokumenty a pracovat s nimi jako by to byly fyzické soubory. Zde se však někdy vyskytuje problém s převodem češtiny<sup>150</sup>. Také při porovnání lokálních souborů se může vyskytnout problém, pokud mají odlišně kódovány

<sup>148</sup> Funkce opravování funguje pouze s těmito prohlížeči, funkce detekce plagiátů je univerzální a lze ji využít i v jiných.

<sup>149</sup> Poslední verze programu je z roku 2004. Neporadila si s dokumentem z Wordu 2003. Extrahovala pouze část textu kratší než jeden odstavec. Pokusný dokument přitom obsahoval pouze minimální formátování.

<sup>150</sup> Zdá se, že problémy jsou hlavně s texty v UTF-8.

české znaky. Uživatelské rozhraní je strohé ale přehledné a funkční. Návod není k dispozici, ale veškeré volby jsou přehledně popsány. Uživatel má možnost rozdělit korpus na nové a staré dokumenty jak bylo popisováno v kapitole 5.10.2.



Obrázek 22: Prostředí nástroje WCopyFind

Výstup je ve formě HTML otevírané v systémovém prohlížeči. Kromě souhrnu s vyznačením míry podobnosti dokumentů (oboustranné čili je použito asymetrické metriky) je k dispozici také text každého z podezřelých dokumentů s vyznačeným obsahem, který se shoduje. Shodné pasáže jsou provázány odkazy. Takové porovnání je vytvořeno pro každou podezřelou dvojici zvlášť. Výsledky tak nejsou sloučeny dohromady pro každý soubor. Vytknut lze řazení výsledků podle názvu dokumentu a nikoliv podle nalezené podobnosti.

Poslední verze programu 2.6 pochází z roku 2004. K dispozici je zdrojový kód pod licencí GNU GPL jak pro WCopyFind pro Windows, tak i pro jednodušší o dva roky starší verzi pro Linux (copyfind).

Tabulka 15: Charakteristika nástroje WCopyFind

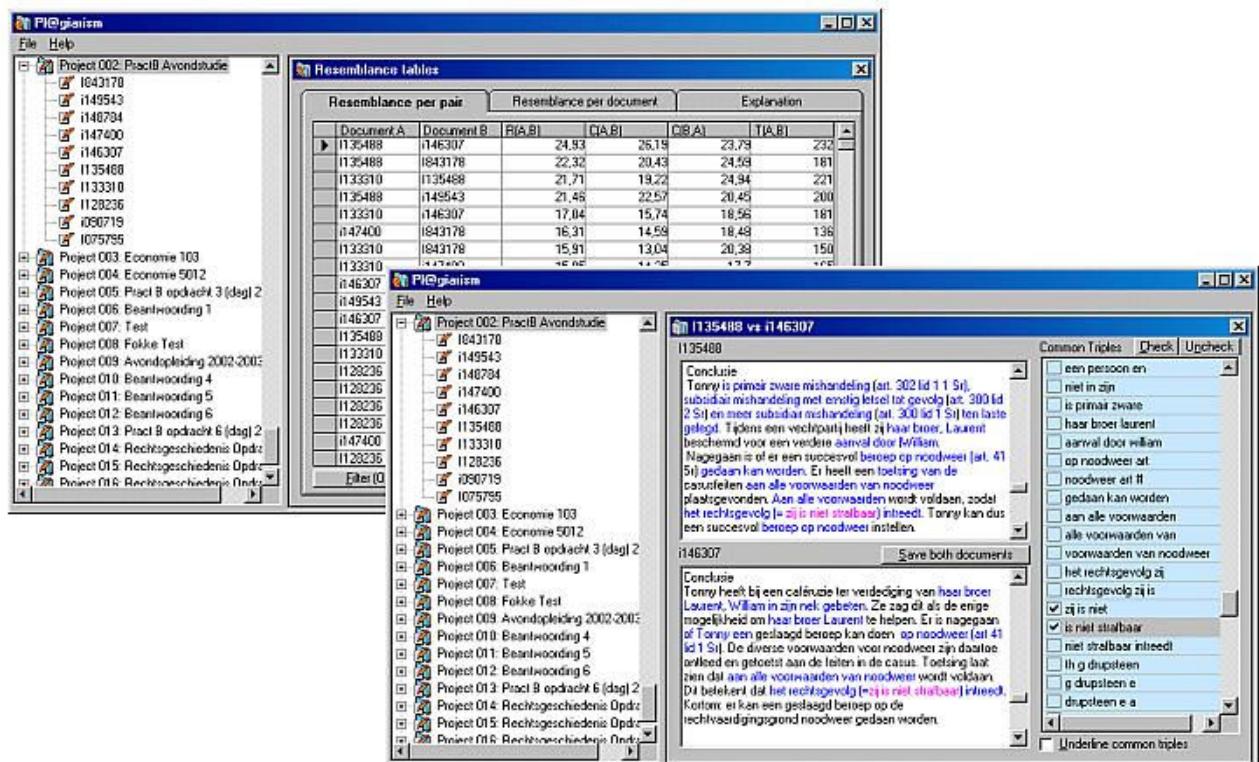
<i>Název</i>		<i>WCopyFind</i>
<i>Výrobce/autor/provozovatel</i>		Lou Bloomfield, University of Virginia
<i>Adresa</i>		<a href="http://plagiarism.phys.virginia.edu/">http://plagiarism.phys.virginia.edu/</a>
<i>Typ dokumentů</i>	<i>obsah</i>	volný text
	<i>jazyk</i>	nespecifikováno (zvládá korektně i češtinu)
	<i>formát</i>	plaintext, HTML, URL (odkazy na online HTML), DOC (s podstatnými chybami)
<i>Způsoby detekce</i>		pouze intrakorpální s možností rozdělení korpusu na nové a staré
<i>Dostupnost</i>	<i>licence</i>	GNU GPL, zcela zdarma včetně zdrojového kódu
	<i>připojení k síti</i>	není potřeba (pouze pro práci s URL)
	<i>zpracování</i>	lokální
	<i>použité metody</i>	k dispozici zdrojový kód (MS Visual C++.Net); vzájemné porovnávání plného obsahu dokumentů, základní jednotkou je slovo, citlivost lze volit minimální délku fráze a minimálním shodným počtem slov, případně i umožněním přeskakování některých slov; asymetrická metrika na principu obsahu
<i>Další</i>	<i>lokalizace</i>	ne, diakritiku zvládá bez problémů (pouze stejné znakové sady); rozhraní v angličtině
	<i>prostředí</i>	Windows (starší verze i Linux)

### 8.3.2 Pl@giarism

Holand'an Georges Span z Faculty of Law na University of Maastricht zřejmě stojí za dalším akademickým projektem pro detekci plagiátů. Je jím nástroj nazvaný Pl@giarism ([plagiarismtk]). Jedná se opět o čistě intrakorpální nástroj, který porovnává kompletní obsah dokumentů. Alternativně je možné asistovat při vyhledávání v přednastavených vyhledávačích, ale tato možnost spočívá pouze v tom, že se v systémovém prohlížeči otevře vyhledávač s výsledky pro daný jeden konkrétní dotaz (text pro vyhledávání se vybírá v Pl@giarism myší). Vyhledávací modul navíc není propojen s detekčním.

Ten tentokrát pracuje se základní jednotkou, kterou jsou tři po sobě jdoucí slova (triplet). Základní jednotky se překrývají a posunují se o vždy o jedno slovo vpřed (čili strategie, kterou jsme výše označili jako V<sub>3a</sub>). Je možné rozdělit korpus na nové a staré dokumenty a navíc specifikovat dokument obsahující text, který má být z porovnání zcela vyřazen.

Pl@giarism je znatelně pomalejší než WCopyFind, ale výsledky dodává stále ještě v rozumném čase. Výstup je formou tabulky a obsahuje jak symetrickou (klasickou podobnost), tak obě asymetrické metriky (obsah v obou směrech). Výsledky je možné třídit podle libovolného údaje a navíc je možné přepnout do zobrazení, kde jsou zhruba sloučeny pro všechny dokumenty. Součástí je i porovnávací modul, kde je v horním a spodním okně vidět text obou dokumentů a vpravo jsou společné triplety. Výběrem textu v jednom okně lze najít příslušný text v druhém okně.



Obrázek 23: Prostředí nástroje Pl@giarism

Tento nástroj umí pracovat i s dokumenty ve formátech DOC a RTF (pro převod volá přímo knihovny MS Word takže ten musí být na daném počítači nainstalovaný ale převod díky tomu probíhá bez problémů) a HTML. Češtinu korektně zvládá pouze dle znakové sady Windows-1250, ale je schopen porovnat i dva dokumenty v jiné (ale též) znakové sadě. V případě UTF-8 potom ale například detailní zobrazení shodných částí dokumentů nefunguje správně.

Nevýhoda je, že tento nástroj přímo zapisuje do adresáře s původními dokumenty jak převedené dokumenty z jiných formátů, tak i jakési své indexy a dokonce i databázový soubor pro MS Access pro celý projekt. Naštěstí je ale možné jedním tlačítkem v nastavení projektu téměř všechny tyto soubory automaticky vymazat.

Trochu zvláštní je také licence tohoto programu. Nejnovější verze pochází z května 2006 a je označena jako beta verze a zároveň omezená demo verze. Ta je omezena na možnost pracovat najednou

pouze s dvěma projekty (zhruba korpusy). V každém může být navíc nanejvýš 20 dokumentů<sup>151</sup>. Na webu autora se píše, že po odeslání licenčních souborů, které program vygeneruje<sup>152</sup>, bude obraťem zaslán<sup>153</sup> zdarma licenční klíč pro plně funkční verzi.

<i>Název</i>	<i>Pl@giarism</i>	
<i>Výrobce/autor/provozovatel</i>	Georges Span, University of Maastricht	
<i>Adresa</i>	<a href="http://www.plagiarism.tk/">http://www.plagiarism.tk/</a>	
<i>Typ dokumentů</i>	<i>obsah</i>	volný text
	<i>jazyk</i>	nespecifikováno (zvládá korektně i češtinu ve Windows-1250)
	<i>formát</i>	plaintext, RTF, DOC (vyžaduje MS Word),
<i>Způsoby detekce</i>	zřejmě pouze intrakorpální s možností rozdělení korpusu na nové a staré soubory	
<i>Dostupnost</i>	<i>licence</i>	údajně zdarma po registraci mailem, částečně omezená demo beta verze
	<i>připojení k síti</i>	není vyžadováno (pouze pro práci s vyhledávači)
	<i>zpracování</i>	lokální
	<i>použité metody</i>	detailní informace nejsou známy; zřejmě porovnání plného obsahu dokumentů, základní jednotkou překryvající se triplet slov, použité metriky: klasická podobnost, obsah
<i>Další</i>	<i>lokalizace</i>	ne, diakritiku zvládá (bez problémů pouze Windows-1250); rozhraní v angličtině
	<i>prostředí</i>	Windows (+MS Word)

### 8.3.3 Ferret

Ferret ([ferret]) je jeden z nejmladších nástrojů pro detekci plagiátů pocházející z akademické sféry. Je dílem skupiny nazvané Plagiarism Detection Research Group na britské University of Herfordshire. V současnosti se dále rozvíjí a nedávno byla uvolněna nová verze. Jde opět o klasický intra-

151 V praxi jich bylo možno porovnávat více, přestože program upozorňoval na toto omezení.

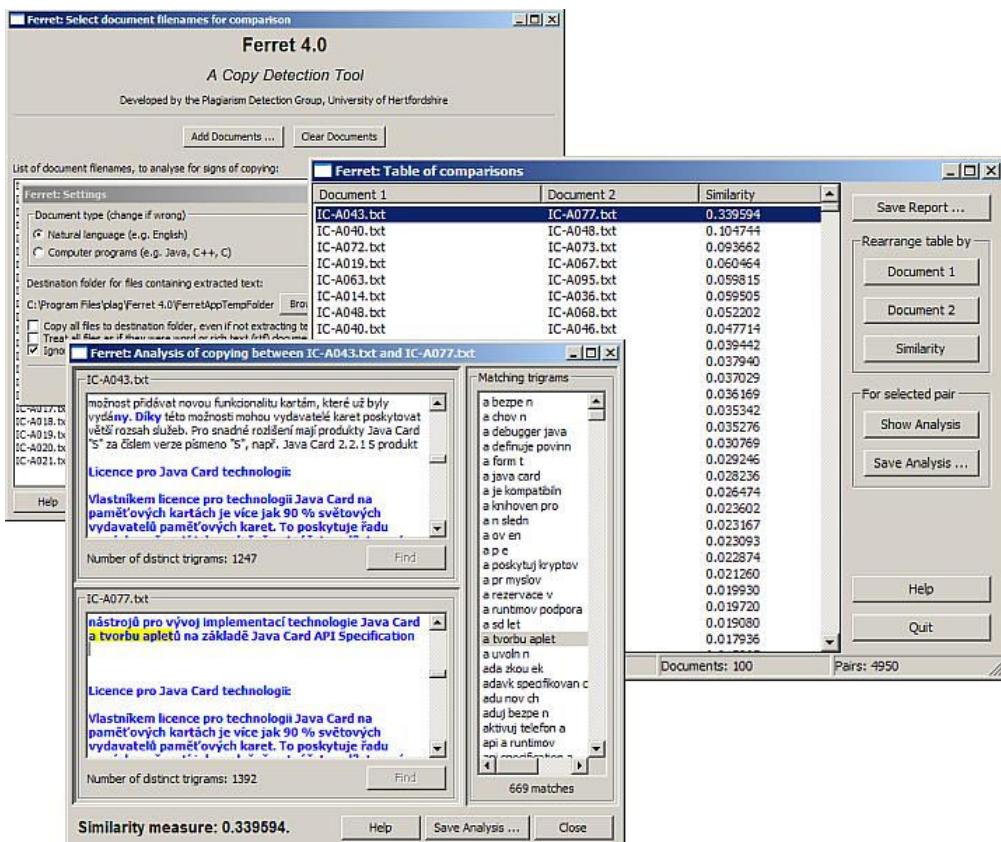
152 Snaží se je generovat (přestože již existují) při každém spuštění programu samotného a nebo některé jeho části, což je pro uživatele obtěžující.

153 Příslušný dopis se nejprve vrátil jako nedoručený, ale za několik dní skutečně přišla odpověď s příslušnými klíči.

korpální nástroj, který je tentokrát schopen ve dvou režimech pracovat jak s přirozeným jazykem, tak s programovacími jazyky.

Nová verze 4.0 podporuje (prostřednictvím dodaných nástrojů třetích stran<sup>154</sup>) také dokumenty ve formátech RTF, DOC, PDF<sup>155</sup>. Kupodivu však nelze porovnat dokumenty v HTML. Ferret je při porovnávání velice rychlý, výsledky jsou srovnatelné s WCopyFind<sup>156</sup>. Obsahuje integrovaný prohlížeč obdobného rozvržení jako ten v Pl@giarism. Navíc je možné exportovat výstup do PDF a to jak celkový přehled, tak porovnání každé dvojice zvlášť.

Z našich testů však vyplynulo, že i přes tyto výhody nepracuje korektně s češtinou písmena s diakritikou jsou používána jako ukončení slova stejně jako mezera nebo jiný znak. Autoři o tomto problému vědí a na svém webu nabízejí zájemcům, kteří by chtěli Ferret používat v jiných jazycích, aby je kontaktovali. Tento problém ale (na rozdíl od některých výše zmiňovaných zejména extra-korpálních nástrojů<sup>157</sup>) nebrání v úspěšné detekci<sup>158</sup>.



Obrázek 24: Prostředí nástroje Ferret

Souhrnné výsledky jsou zobrazeny v tabulce a je možné je seřadit jak podle názvů dokumentů, tak podle míry jejich podobnosti. Detailní zobrazení tradičně ukazuje oba podobné dokumenty s modré

154 Jde o zejména o AbiWord a pdftotext.

155 Problém však byl při převodu českých znaků v PDF dokumentech ty prostě vypadly. Ovšem dva obsahově shodné (ale graficky odlišné) takto špatně převedené dokumenty byly detekovány jako 100% kopie.

156 Když jsme zkusmo porovnávali více než 280 dokumentů, byl dokonce rychlejší.

157 Chybě převedená slova nelze najít v internetovém vyhledávači, protože taková často neexistují natož aby se vyskytovala v takových frázích jako v původním dokumentu.

158 Slova s diakritikou se téměř vždy převedou na stejný shluk tripletů a je možné je porovnat.

vyznačenými shodami. Nejsou provázány odkazy, ale je možné se v obou najednou pohybovat výběrem některého ze shodných tripletů, které jsou také zobrazeny v seznamu (ale jsou řazeny pouze abecedně, nikoliv podle polohy v textu). Jak tyto detailní, tak souhrnné výsledky pro celý korpus je možné vyexportovat do formátu PDF.

Tento nástroj neobsahuje žádné pokročilejší možnosti nastavení detekce. Jistou nevýhodou je také použití pouze základní symetrické metriky podobnosti, která není vhodná pro porovnávání souborů výrazně různých délek. Jeho největší výhodou tak je opravdu velmi rychlé porovnání i několika stovek dokumentů a poměrně dobré možnosti práce s výsledky.

<i>Název</i>		<i>Ferret</i>
<i>Výrobce/autor/provozovatel</i>		Plagiarism Detection Research Group University of Hertfordshire, UK
<i>Adresa</i>		<a href="http://homepages.feis.herts.ac.uk/~pdgroup/">http://homepages.feis.herts.ac.uk/~pdgroup/</a>
<i>Typ dokumentů</i>	<i>obsah</i>	volný text, zdrojové kódy
	<i>jazyk</i>	volný text nespecifikováno (určitě angličtina a čínština), problémy s diakritikou nebrání výrazně v detekci češtiny zdrojové kódy určitě Java, C a C++, možná i další <sup>159</sup>
	<i>formát</i>	plaintext, RTF, PDF, DOC
<i>Způsoby detekce</i>		pouze intrakorpální
<i>Dostupnost</i>	<i>licence</i>	zdarma k použití bez záruk, zdrojový kód není veřejně k dispozici
	<i>připojení k síti</i>	není potřeba
	<i>zpracování</i>	lokální
	<i>použité metody</i>	částečně publikovány, zdrojový kód není k dispozici, porovnání plného obsahu dokumentu, základní jednotkou překrývající se triplet slov, použitá metrika pouze klasická podobnost
<i>Další</i>	<i>lokalizace</i>	ne (českou diakritiku neumí, ale nebrání to výrazněji v detekci); prostředí v angličtině
	<i>prostředí</i>	Windows <sup>160</sup>

159 Univerzální tokenizace zřejmě předpokládá styl odpovídající jazyku C.

### 8.3.4 CISE tools

Tady se podíváme na několik nástrojů pro boj s plagiátorstvím z dílny Centre for Interactive Systems Engineering<sup>161</sup> (CISE) na London South Bank University ([cise]). Tyto nástroje jsou spíše experimentální, a jak upozorňují sami autoři, nemají produkční kvalitu. Nástroje již nejsou nejspíše několik let dále vyvíjeny a jejich poslední verze pocházejí z let 2003 až 2004.

Všechny dostupné nástroje z této sady je možné spustit v internetovém prohlížeči jako Java applety. Předtím je však nutné upravit bezpečnostní pravidla pro přístup Java modulu k lokálním souborům a pro práci se vzdáleným serverem.

První z nástrojů je nazván **OrCheck** (originality checker). Slouží k asistenci při ručním extra-korpálním vyhledávání pomocí Google API. Bohužel se v současnosti již nezdá funkční. Analýzou síťového provozu jsme zjistili, že to je nejspíše změnami v rozhraní Google API, protože server na libovolný dotaz nástroje vrací chybový kód 500. Nástroj pracuje vždy pouze s jedním testovaným dokumentem a vypočítává frekvenci a četnost jednotlivých slov a trojic slov v dokumentu. Na základě těchto statistik může uživatel některé z nich vybrat a ty jsou následně vyhledány pomocí Google API. Potom (pokud by nástroj stále fungoval) by bylo možné porovnat obsah dokumentu s obsahem možných zdrojů nalezených na základě dotazu a případně zobrazit shodné části.

Nástroj **PRAISE** (Plotted Ring of Analysed Information for Similarity Exploration) slouží k intrakorpální analýze plagiátů a zobrazení vzájemných vztahů a vazeb mezi navzájem podobnými dokumenty. Základní jednotkou mohou být znaky, slova nebo věty podle volby uživatele. Pracuje zejména s prostým textem a celkové výsledky zobrazuje graficky pomocí spojnic mezi dokumenty. Ty jsou znázorněny jako body na obvodu kružnice. Jejich spojnice představují podobnost a umožňují přehledně znázornit nejen dvojice, ale případně i shluky (clustery) podobných dokumentů. Pokud je dokumentů málo, je to poměrně přehledné, při větším počtu již ale toto zobrazení na přehlednosti výrazně ztrácí. Výsledek může být zobrazen i ve formě tabulky a detail pomocí integrovaného nástroje VAST.

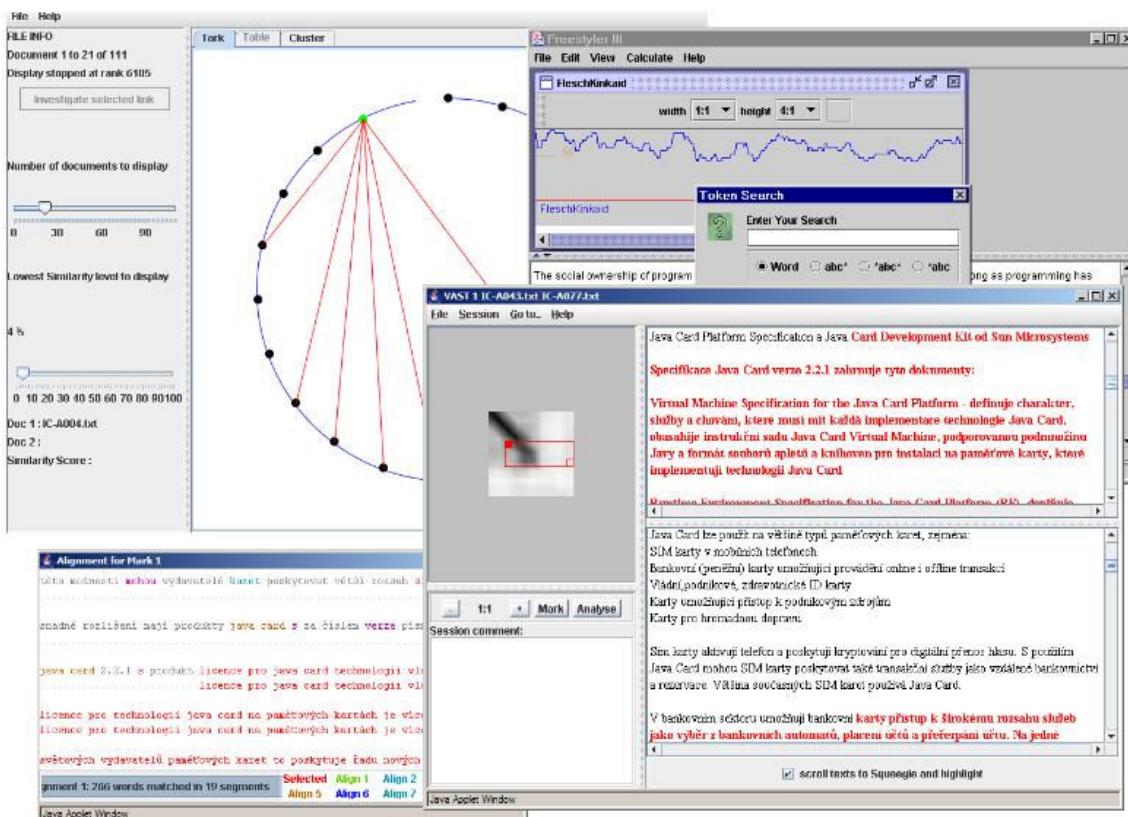
**VAST** (Visual Analysis of Similarity Tool) je nástroj pro porovnání dvou dokumentů. Kromě textu obou zobrazuje také graficky shodu jejich obsahu (formou diagonálních pruhů) a umožňuje i jejich vzájemné porovnání pomocí zarovnání. Pracuje tedy zejména s přístupy známými z bioinformatiky.

Kombinace PRAISE a VAST se zdá jako poměrně dobrá volba zejména z hlediska pokročilé, ale stále poměrně přehledné prezentace výsledků porovnání<sup>162</sup>. Samotné porovnání ale není nejrychlejší (zejména při srovnání s Ferretem nebo WCopyFind) a prostředí Java appletu neposkytuje takový komfort, ale i tak se jedná o poměrně zajímavý nástroj, který může dobře sloužit zkušenějšímu individuálnímu uživateli. Při zpracování velmi dlouhých dokumentů je ale potřeba počítat s delšími prodlevami odezvy uživatelského rozhraní. Výsledky testů dopadly poměrně úspěšně a české diakritika ve Windows 1250 nebyla problémem. Ostatní kódové stránky se nezobrazovaly korektně, ale nebránilo to v detekci.

<sup>160</sup> UI programu je od verze 3.0 založeno na multiplatformní knihovně wxWidgets, ale k dispozici je, zdá se, pouze verze pro Windows.

<sup>161</sup> Členy byli také pánové Lancaster a Culwin, kteří se nástroji pro detekci plagiátů dlouhodobě zabývají a jsou mimo jiné i autory klasifikace těchto nástrojů, ze které vycházíme v kapitole 2.2.

<sup>162</sup> Na rozdíl od například nástroje Sherlock, který se nám zdál již poměrně překombinovaný (viz dále).



Obrázek 25: Prostředí nástrojů PRAISE, VAST a FreeStyler

Posledním nástrojem je **FreeStyler**, který slouží pro vizualizaci některých stylometrických údajů dokumentu.

Tabulka 16: Charakteristika nástrojů PRAISE a VAST

<b>Název</b>		<b>CISE tools (zejm. PRAISE a VAST)</b>
<b>Výrobce/autor/provozovatel</b>		Centre for Interactive Systems Engineering
<b>Adresa</b>		<a href="http://cise.lsbu.ac.uk/tools.html">http://cise.lsbu.ac.uk/tools.html</a>
<b>Typ dokumentů</b>	<b>obsah</b>	primárně volný text
	<b>jazyk</b>	nespecifikováno, s češtinou si poradí bez problémů
	<b>formát</b>	plaintext
<b>Způsoby detekce</b>		intrakorpální

<b>Dostupnost</b>	<b>licence</b>	zřejmě volně k použití bez jakýchkoliv záruk; zdrojový kód není veřejně k dispozici
	<b>připojení k síti</b>	ano vyžadováno
	<b>zpracování</b>	převážně lokální <sup>163</sup>
	<b>použité metody</b>	detajná informace nejsou k dispozici; zřejmě porovnání úplného obsahu dokumentů; základní jednotky znaky, slova, věty; clusterizace výsledků; vizualizace; zarovnání textu
<b>Další</b>	<b>lokalizace</b>	ne, s češtinou nemá problémy, rozhraní anglicky
	<b>prostředí</b>	Java applet

### 8.3.5 JPlag

Zřejmě nejznámější současnou službou specializovanou na odhalování plagiátů ve zdrojových kódech je služba JPlag ([jplag]). Její základ vznikl v roce 1996 jako studentský projekt na univerzitě v Karlsruhe. V roce 2005 až do současnosti je provozována jako webová služba. K dispozici je klient napsaný v Javě a dokumentace pro tvorbu vlastního klienta. Služba je poskytována zdarma, ale uživatel se musí registrovat a uvést důvod a účel, za jakým ji chce využívat. Registrace není automatická a podléhá schválení.

Služba je schopná zpracovat zdrojové kódy v několika programovacích jazycích. Jde o Java<sup>164</sup>, Scheme, C/C++ a C#. Je také možné porovnávat volný text v plaintextovém formátu, případně jako TeX.

Jedná se o čistě intrakorpální nástroj. Podle dostupných informací [Prechelt2002] porovnání zdrojových souborů probíhá tak, že po načtení jsou modulem pro příslušný jazyk převedeny na sekvenci zástupných symbolů. Výsledek převodu je následně porovnán pomocí algoritmu pro porovnání řetězců<sup>165</sup>.

Poskytovaný Java klient je možné spustit pomocí JavaWebStart přímo z webu služby JPlag. Je uživatelsky velmi přívětivý a umožňuje uživateli výběr dokumentů k porovnání (celý adresář) a nastavení parametrů. Následně dokumenty zabalí a odešle ke zpracování a poté přijme odpověď s výsledkem. Porovnání probíhá poměrně rychle, i výsledek pro několik stovek dokumentů je vrácen do několika minut od odeslání požadavku. Doba odezvy záleží také na počtu požadavků čekajících na serveru na zpracování, ale během testů toto čekání nikdy netrvalo příliš dlouho.

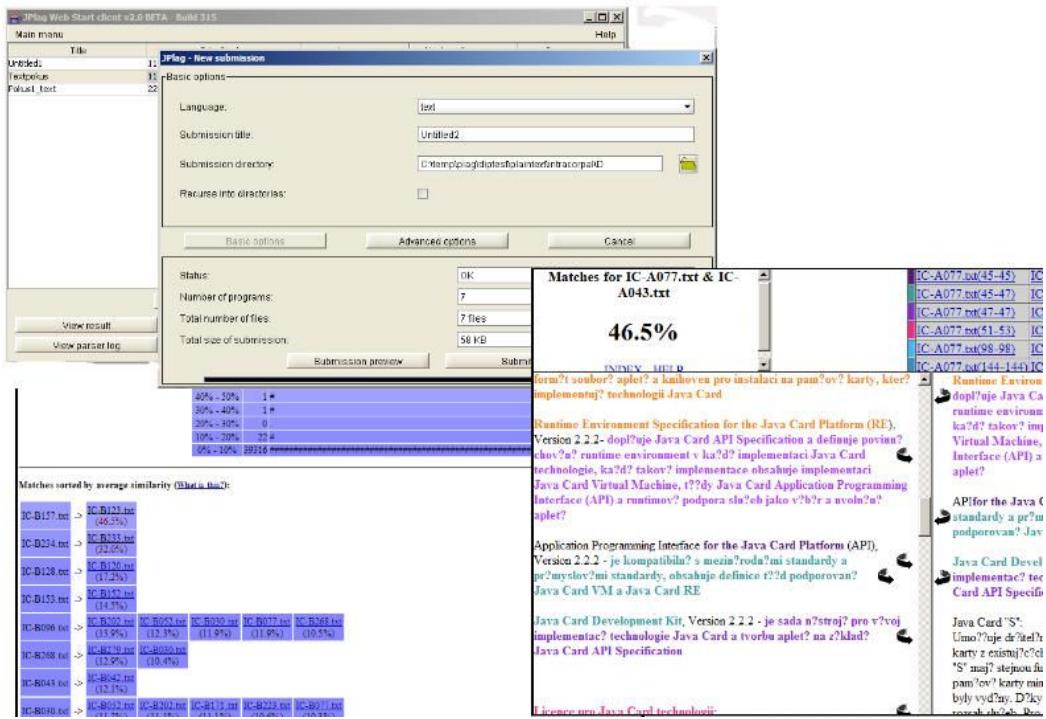
Výsledky nejsou zobrazeny v klientovi ale ve formě HTML reportu. Ten obsahuje seznam všech souborů, parametry porovnání a odkazy na detailní zobrazení dokumentů, které překročily výstražnou úroveň. Tato výsledková sekce má dvě části, které se liší uváděnými procenty shody.

<sup>163</sup> Stažený Java applet je spouštěn lokálně v běhovém prostředí prohlížeče.

<sup>164</sup> Je možno volit mezi syntaxí verzí 1.2 a 1.5

<sup>165</sup> Modifikace Greedy String Tiling od M. Wise, autora série nástrojů YAP.

V první je zobrazena tzv. průměrná podobnost a v druhé maximální podobnost. V praxi jde o klasickou symetrickou metriku a maximalizací symetrizovanou asymetrickou metriku na principu obsahu. V detailním zobrazení jsou shodné části dokumentu provázány odkazy a barevně odlišeny.



Obrázek 26: Prostředí služby JPlag

V našich testech si vedl tento nástroj poměrně velmi dobře. Zdá se však, že příliš nezvládá zobrazovat českou diakritiku. Nebrání mu to však v porovnávání, dokumenty jsou porovnány korektně, ale ve výsledcích se místo českých znaků objevují otazníky.

Tabulka 17: Charakteristika služby JPlag

<i>Název</i>		<i>JPlag</i>
<i>Výrobce/autor/provozovatel</i>		Guido Malpohl <sup>166</sup> , Universität Karlsruhe, Německo
<i>Adresa</i>		<a href="https://www.ipd.uni-karlsruhe.de/jplag">https://www.ipd.uni-karlsruhe.de/jplag</a>
<i>Typ dokumentů</i>	<i>obsah</i>	zdrojové kódy, volný text
	<i>jazyk</i>	zdrojové kódy – Java, Scheme, C, C++, C#  volný text – oficiálně podporovány angličtina, němčina, francouzština a španělština; s českou diakritikou má problémy při zobrazení ne při porovnávání

166 Hlavní a původní autor, od té doby se na vývoji a provozu podílí více lidí.

	<i>formát</i>	plaintext, TeX
<i>Způsoby detekce</i>		intrakorpální
<i>Dostupnost</i>	<i>licence</i>	služba poskytována zdarma bez garance a po nenárokové registraci
	<i>připojení k síti</i>	ano vyžadováno
	<i>zpracování</i>	distribuované
	<i>použité metody</i>	k dispozici pouze částečné informace, nahrazení obsahu tokeny a jejich porovnání pomocí modifikovaného Greedy String Tiling algoritmu; použitá symetrická metrika na bázi podobnosti a symetrizovaná metrika na bázi obsahu
<i>Další</i>	<i>lokalizace</i>	ne; jazyky rozhraní angličtina, němčina, francouzština a španělština
	<i>prostředí</i>	webová služba, dostupný Java klient (Java 5)

### 8.3.6 Sherlock

Nástroj Sherlock ([sherlock])<sup>167</sup> je v současnosti součástí open source nástroje BOSS Online Submission System, který slouží pro správu vyučovacích kurzů zejména programátorských. Sherlock poskytuje podporu pro detekci plagiátů v odevzdávaných dokumentech. K dispozici je ale i samostatně a to včetně grafického uživatelského rozhraní.

Ve zdrojových kódech probíhá detekce v několika fázích. Dokumenty jsou porovnávány několikrát po různém vstupním zpracování. To by mělo umožňovat odhalit více potenciálních transformací, které mohli plagiátori provést, než klasické univerzální jednostupňové předzpracování. Nejprve jsou tedy dokumenty předzpracovány podle daných pravidel a to do několika na sobě nezávislých podob. K dispozici jsou tyto možnosti

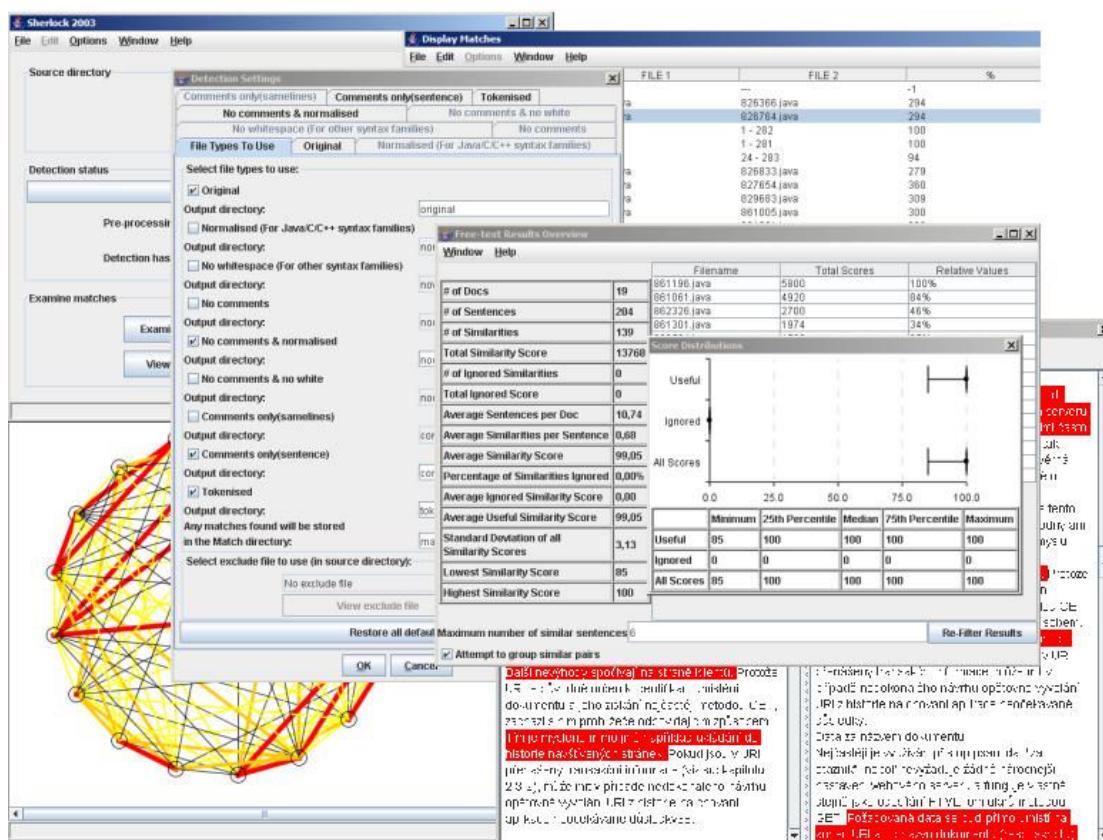
- původní originální dokument
- normalizovaný dokument (pro jazyky Java a C/C++)
- dokument bez whitespaces (pro ostatní jazyky)
- dokument bez komentářů
- dokument bez komentářů a normalizovaný
- dokument bez komentářů a bez whitespaces

<sup>167</sup> Existuje ještě jiný nástroj pro vyhledávání plagiátů se stejným jménem. Nachází se na adrese <http://www.cs.usyd.edu.au/~scilect/sherlock/>

- pouze komentáře (porovnávání řádků)
- pouze komentáře (porovnávání vět)
- tokenizovaný dokument

Je zřejmé, že takové porovnání není zrovna nejrychlejší, už proto, že se porovnává několikrát více dokumentů a předtím je nutné je z původních vytvořit, uložit na disk a pak znova načíst.

Podle [White2004] je detekce přirozeného textu v nástroji Sherlock nejnověji realizována tak, že za základní jednotky jsou brány věty bez ohledu na pořadí slov. Zřejmě je použit i nějaký mechanismus, který umožňuje ztotožnit i věty, které nemají všechna slova shodná. Praktické výsledky testů v českém jazyce ale ukázaly, že jsou jako shodné označovány někdy i zcela rozdílné věty, které obsahují několik stejných slov. Zajímavé ale bylo, že v některých případech šlo skutečně nejspíše o různé varianty překladů téhož textu z angličtiny. Pod Windows zobrazoval korektně pouze češtinu dle Windows-1250. S UTF-8 měl poměrně výrazné problémy.



Obrázek 27: Prostředí nástroje Sherlock

Výsledky jsou velmi detailní, je možné zobrazit podobný kruhový graf podobností velmi podobný tou z nástroje PRAISE<sup>168</sup>. Dále je možné si prohlédnout různé statistické údaje o korpusu a také je k dispozici detailní zobrazení nalezených shod vyznačených v jednotlivých dokumentech. Uživatel si také může zobrazit pohled, kde v levé části je grafické znázornění shod v podobě sloupců, a v pravé čtyři okna pro zobrazení dokumentu (dva původní podoba a dva tytéž ale po zvoleném

168 Ale nutno říci, že tento je díky použití různých druhů a barev přehlednější.

předzpracování). Celkově na nás ale prezentace výsledků tohoto nástroje nepůsobila subjektivně příliš přehledně a intuitivně<sup>169</sup>. Podobně jako nástroj Pl@giarism také ukládá své předzpracované dokumenty a konfigurační soubory přímo do původního adresáře s dokumenty.

*Tabulka 18: Charakteristika nástroje Sherlock*

<b>Název</b>		<b>Sherlock</b>
<b>Výrobce/autor/provozovatel</b>		Daniel White, Mike Joy, Russell Boyatt, Paul Isitt
<b>Adresa</b>		<a href="http://sourceforge.net/projects/cobalt/">http://sourceforge.net/projects/cobalt/</a> <sup>170</sup>
<b>Typ dokumentů</b>	<b>obsah</b>	zdrojové kódy, volný text
	<b>jazyk</b>	zdrojové kódy – Java, C++ volný text – nespecifikováno, s češtinou nemá problém
	<b>formát</b>	plaintext
<b>Způsoby detekce</b>		intrakorpální
<b>Dostupnost</b>	<b>licence</b>	GNU GPL, zcela zdarma včetně zdrojového kódu
	<b>připojení k síti</b>	nevýžadováno
	<b>zpracování</b>	lokální
	<b>použité metody</b>	k dispozici zdrojový kód (Java), vícestupňové předzpracování dokumentů a následné porovnání výstupů různého zpracování, u prostého textu základní jednotkou věta bez ohledu na pořadí slov
<b>Další</b>	<b>lokalizace</b>	ne, rozhraní anglicky, s češtinou nemá problém
	<b>prostředí</b>	Java, integrováno do BOSS

## 8.4 Shrnutí a doporučení

Byť výše uvedený výčet není zdaleka úplný, je z něho a z dosažených výsledků testů poměrně patrné, že trh nástrojů pro detekci plagiátů je dosud poměrně nevyspělý. V oblasti extrakorpální detekce převažují dva silní hráči TurnItIn.com a menší MyDropBox. Ani jejich řešení však nejsou

<sup>169</sup> Ani na již tak poměrně preplněný obrázek 27 se nám nevešly všechny dostupné obrazovky.

<sup>170</sup> Adresa systému BOSS. Její předchozí název byl zřejmě cobalt.

zdaleka dokonalá, byť jsou poskytovány za relativně velmi vysoké ceny. Nezdají se ani příliš vhodná pro české jazykové prostředí. Ještě horší je situace u ostatních extrakorpálních nástrojů. Jejich funkcionality je slabá a připravenost pro české prostředí mizivá.

Mnohem lepší je situace v oblasti intrakorpálních nástrojů. Mezi těmi pro lokální použití lze vybrat poměrně kvalitní a použitelné nástroje, které jsou navíc velmi často poskytovány zdarma. Vzhledem k větší angažovanosti vyučujících programování a přeci jen nižší obecné náročnosti zpracování zdrojových kódů jsou tedy nejvýspějšími nástroji právě ty, které je zpracovávají a to výhradně v intrakorpálním režimu. Podrobnější výsledky hlavních testů jsou shrnutы в tabulkách 19 a 20.

*Tabulka 19: Výsledky intrakorpálních testů*

Název nástroje/služby	Výsledek zpracování vzorku A	Doba zpracování vzorku B <sup>171</sup> 10/20/50/100/281 dokumentů
MyDropBox	dvě podobné dvojice identifikovány, další vůbec, relativně dost FP <sup>172</sup>	netestováno
WCopyFind	všechny podobné dvojice identifikovány (obsah nad 10 %) pouze jeden FP (několik shodných nesouvislých slovních spojení zejm. názvy produktů a technologií)	<1 s / <1 s / 1 s / 2 s / 7 s
Pl@giarism	všechny podobné dvojice identifikovány (obsah nad 10 %), ostatní pod 10 %	33 s / 33 s / 49 s / 97 s / 341 s
Ferret	všechny podobné dvojice identifikovány (podobnost nad 5 %), jeden FP (nad 5%) ne-související kratší shodné úseky <sup>173</sup>	<1 s / <1 s / 1 s / 2 s / 7 s
CISE (PRAISE)	kromě jedné všechny podobné dvojice identifikovány (podobnost nad 20 %), 1 FP (nesouvi-sející shodné části vět), poslední podobná dvojice s podobností nad 12 % (zde 4 FP)	2 s <sup>174</sup> / 4 s / 16 s / 64 s / cca 650 s

171 U lokálních nástrojů testy prováděny na stroji Intel Pentium D805 s 1GB RAM s Windows XP Professional SP2 bez jiné zátěže. Obě jádra procesoru využil pouze nástroj Pl@giarism, který je tvořen dvěma spustitelnými soubory.

172 Ale zřejmě zejména kvůli tomu, že zároveň probíhala extrakorpální detekce, která nezřídka nalezla ukázky zdrojových kódů v oficiálních dokumentacích. Jako „neobvyklé věty“ dostaly zřejmě jejich SMC poměrně velkou váhu. Při použití standardnějšího korpusu lze očekávat vyšší přesnost.

173 Kvůli tomu, že písmena s diakritikou používá jako oddělovače slov a proto jsou některé triplety velmi krátké.

174 Dobý zpracování jsou zde bez vizualizace a zarovnání obsahu dokumentů jednotlivých dvojic.

JPlag <sup>175</sup>	všechny podobné dvojice nalezeny (max. obsah nad 10 %) 1 FP – krátké nesouvisející shodné úseky, ostatní pod 10 %	55 s <sup>176</sup> / 28 s / 33 s / 63 s / 146 s <sup>177</sup>
Sherlock	nalezeny čtyři podobné dvojice (skóre nad 25 %) jedna nenalezena (skóre 5 %) ač má shodné plné tři odstavce, poměrně dost FP (zejména několik různých vět s podobnými slovy)	24 s / 38 s / 120 s / cca 430 s / 56 min

Podobné zdrojové kódy byly všemi zvolenými nástroji poměrně dobře rozpoznány a doby odezvy odpovídaly zhruba údajům získaným při porovnání volného textu (samořejmě s přihlédnutím k menší průměrné délce zdrojových kódů).

Tabulka 20: Výsledky extrakorpálních testů

Název nástroje/služby	Výsledek zpracování vzorku W								
	anglické dokumenty					české dokumenty			
	Money	Dr.Deer	Napoleon	ProQuest	Peníze	Peníze+	Matka	vzorek A	
CatchItFirst	ne	ano	ano	částečně <sup>178</sup>	ne	ne	část <sup>179</sup>	část <sup>179</sup>	
Eve 2	ano	ano	ano	ne	ne	ne	ne	ne	
Plagiarism Finder	ne	ne	ne <sup>180</sup>	ne	ne	ne	ne	ne <sup>181</sup>	
MyDropBox	ano	ano	ano	část <sup>179</sup>	část <sup>179</sup>	část <sup>179</sup>	ne	část <sup>179</sup>	

Případnému zájemci o nástroj pro detekci plagiátů tak na základě zmapování situace lze v současnosti doporučit následující.

- Intrakorpální nástroje pro zdrojové kódy jsou poměrně kvalitní a použitelné za témař nulové náklady. Pro lokální zpracování lze z testovaných nástrojů doporučit Ferret, v případě detajnějších požadavků nastavení a detekce potom Sherlock, pro distribuované zpracování nebo dokonce integraci do vlastního systému pak určitě JPlag.

<sup>175</sup> Test prováděn s připojením cca 2 Mbit/s download a cca 240 kbit/s upload (konektivita UPC). Na linkách s rychlejším uploadem případně lépe napojených na německou akademickou síť mohou být výsledky lepší.

<sup>176</sup> Z nějakého důvodu byl i při opakování pokusech čas na konkrétním použití korpusu B10 delší než u B20 a B50. Při použití jiného korpusu (B10a jako polovina B20) již byl čas okolo 20 s.

<sup>177</sup> Časy jsou včetně uploadu na server a případného čekání ve frontě. V tomto případě připadá zhruba 74 s na zabalení a upload požadavku (cca 1,6 MB), 13 s čekání ve frontě a cca 50 s samotné zpracování na serveru.

<sup>178</sup> Části textu byly nalezeny na webech questia a findarticles, tedy ne přímo v PQ.

<sup>179</sup> Některé či všechny zdroje nalezeny, ale jako nepůvodní označena pouze malá část textu.

<sup>180</sup> Odkazoval na nabídku knihy na serveru book.de. Bohužel nešlo o knihu o Bonapartovi, ale jakýsi špionážní román.

<sup>181</sup> Odkazoval na jakýsi vícejazyčný obrázkový návod v PDF na německý přístroj pro měření izolačních odporů.

- Extrakorpální nástroje pro volný text jsou téměř vždy placené a jejich výsledky a účinnost jsou minimálně v českém prostředí velmi sporné.
- Intrinsic nástroje pro volný text existují, ale jejich využitelnost je sporná a nepředstavují současný trend.
- Intrakorpální nástroje pro volný text existují zejména pro lokální použití. Z hlediska rychlosti zpracování lze nejspíše doporučit Ferret a zejména WCopyFind.
- Extrakorpální ani intrinsic nástroje pro zdrojové kódy prakticky neexistují<sup>182</sup>.
- Smíšené (extra- i intrakorpální) nástroje pro volný text jsou představovány poměrně velmi drahými službami, které v sobě zahrnují komplexní správu kurzů a odevzdávání dokumentů. Ani ty však nejsou zřejmě příliš vhodné pro české prostředí.

V intrakorpální zejména lokální detekci jsou tedy možnosti poměrně bohaté. Navíc i na současných kancelářských počítačích je taková detekce do několika stovek dokumentů otázkou maximálně desítek minut, spíše však méně. Větší problém může nastat v detekci extrakorpální. A to jak té, která pracuje s velkým množstvím (řádově od stovek) uchovaných dokumentů, tak zejména té, která porovnává vůči internetovým zdrojům.

Požaduje-li nějaká místní menší instituce extrakorpální detekci v češtině, tak při současných (a stále rostoucích) cenách poskytovaných služeb (které navíc nejsou pro české podmínky plně připraveny), může se jevit jako poměrně vhodné usilovat o vývoj vlastního řešení, případně ve spolupráci s jinými institucemi s podobnými potřebami. Lze přitom stavět na dostupných publikovaných materiálech a otevřených řešeních. Určitý návrh či základní koncept takového řešení předkládáme v následující kapitole.

---

<sup>182</sup> Zřejmě nejsou potřeba i když by bylo možné uvažovat o nástroji pracujícím se zdrojovými kódy dostupnými v rámci open source. V tom případě by ale zřejmě bylo nutné z hlediska kapacity uvažovat o nějaké variantě nekompletních reprezentací podobně jako u extrakorpální detekce volného textu.

## 9 Návrh systému detekce plagiátů

*Software gets slower faster than hardware gets faster.*

*Niklaus Wirth, švýcarský počítačový vědec (1934)*

*If you can't describe what you are doing as a process, you don't know what you are doing.*

*W. Edwards Deming, americký statistik (1900–1993)*

Jak plyne ze závěru předchozí kapitoly, v současnosti nejsou českým vzdělávacím institucím snadno dostupná řešení pro detekci plagiátů. Na rozdíl od lokálních intrakorpálních nástrojů, které mohou využívat jednotliví vyučující, extrakorpální nástroje s databází historických dokumentů a případně vyhledávání na Internetu jsou velmi nákladné a nepříliš funkční. V této kapitole se tedy pokusíme navrhnut koncept takového nástroje, který by mohl být k těmto účelům využíván lokálními institucemi. Nazveme ho ODPUST tj. Odhalování Plagiátů U STudentů.

### 9.1 Předpoklady návrhu

#### 9.1.1 Instituce

Jako cílového uživatele systému budeme předpokládat instituci typu střední či vyšší školy spíše střední velikosti. Tomu odpovídá kapacita zhruba 250–500 žáků a zhruba 30 vyučujících. Dále budeme předpokládat, že škola již má nějaký informační systém, pro správu kurzů a zejména příjem dokumentů v elektronické formě. Nebudeme předpokládat žádný takový konkrétní systém, ale pouze to, že akvizice dokumentů od studentů je prováděna mimo rámec námi navrhovaného systému a to včetně organizačních záležitostí jako povinnost všechny dokumenty odevzdat elektronicky v některém z předepsaných formátů a podobně<sup>183</sup>.

#### 9.1.2 Integrace do programu prevence plagiátů

Předpokládáme rovněž, že cílem instituce není pouze jen tak odhalovat plagiáty, ale různými metodami proti jejich výskytu všechno bojovat. K tomu by měl sloužit komplexnější integrovaný program PROPUST (PRevence a Odhalování Plagiátů U Studentů). Informační systém pro detekci je pouze jednou z jeho částí, vážící se k informačním technologiím. Ostatní části musí zahrnovat jak institucionálně organizační zázemí například podchycení problematiky v obecných předpisech jako je školní řád, kázeňský řád, interní směrnice a podobně, tak zázemí pedagogické. To by mělo být pokud možno jednotné.

Je zbytečné, aby každý vyučující ve svých předmětech zvlášť poučoval studenty o tom, jak mají psát své práce a jak správně pracovat se zdroji. Tyto dosud často poměrně zanedbávané základní obecné informační dovednosti a návyky<sup>184</sup> by měly být koncepčně zařazeny do výuky.

---

<sup>183</sup> Alternativně je možné předpokládat, že dokumenty mohou být do systému nahrány vyučujícím.

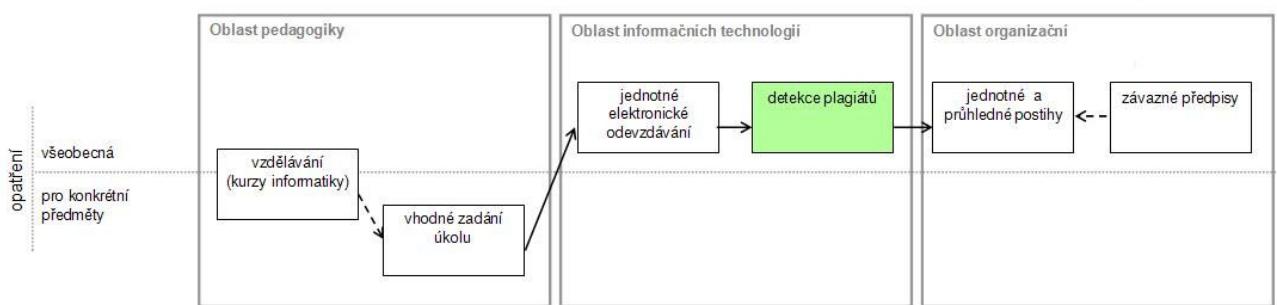
<sup>184</sup> Například, že i značná část toho, co je dostupné na Internetu má svého autora, a není možné to vydávat za svou práci. Ze „zdroj: Internet“ je nicneříkající označení. Ze cílem vyučujícího, který zadá vypracovat práci na dané téma není zdaleka pouze něco nového se osobně o daném tématu dozvědět (i když ani to není vyloučeno), ale

Jako velmi vhodné, a to hned z několika důvodů, se nám jeví zařazení této problematiky do kurzů informatiky. Kromě toho, že tato problematika je jim obsahově nesporně velmi blízká (zejména pokud nejde pouze o ovládání kancelářských aplikací, ale také vyhledávání informací, nejčastěji na Internetu), bývají tyto povinné obecné kurzy zařazovány na většině typů škol již do prvních ročníků. Navíc oproti ostatním předmětům či oborům je informatika stále poměrně mladá a nemá tak pevně zakotvené osnovy, které by zařazení nové problematiky mohly bránit. Objektivně poměrně rychlý vývoj oboru i tak vede k časté aktualizaci jejich obsahu.

Prevenci plagiátorství je vhodné podpořit také v ostatních předmětech, do nichž jsou úkoly vypracovávány. Ideální je volit taková téma, pro něž budou potenciální zdroje plagiátorství (tedy práce na podobné téma) velmi obtížně dostupné, ale přitom nebude tolik náročné sehnat potřebné zdroje. Tedy požadovat téma velmi aktuální, případně neobvyklá nebo v neobvyklých kombinacích. Je také možné u témat běžnějších mít některé specifické požadavky, které plagiátorství ztíží, jako například přiložit k práci také kopii několika zdrojů, provést vlastní analýzu zjištěných údajů a podobně<sup>185</sup>. Podrobněji k této problematice viz například [Haris2004].

Teprve potom nastupují represivní složky programu jako je samotný nástroj pro detekci a případné postupy pro usvědčené viníky. Pozici systému detekce plagiátů v námi navrhovaném komplexním programu jejich prevence ukazuje obrázek 28.

**PROPUST** (PRevence a Odhalování Plagiátů u STudentů)



Obrázek 28: Komplexní program prevence plagiátů a úloha nástroje pro detekci v něm

Každé z naznačených opatření přispívá k prevenci plagiátorství. Vzdělávání v kurzech informatiky poskytuje studentům základní informace o práci s přebíranými informacemi a vylučuje tak náhodné plagiátorství z neznalosti problematiky. Vhodné a specifické zadání konkrétního úkolu snižuje potenciální oblast zdrojů plagiátů. Jednotné elektronické odevzdávání<sup>186</sup> dokumentů<sup>187</sup> umožňuje podrobný centrální sběr odevzdávaných prací v elektronické formě. Ty jsou dále zpracovány nástrojem pro detekci plagiátů, jehož výsledky (po nezbytném manuálním zkontovalování) mohou být řešeny

naučit studenty pracovat s informacemi, hodnotit je v kontextu a formulovat na jejich základě vlastní myšlenky. Zatímco první, čistě informační, cíl by byl splněn i pouhým zkopirováním, ty druhé, výukové, a mnohem důležitější jistě ne.

185 Ovšem studenti by také měli obdržet odpovídající zpětnou vazbu. Rozhodně nelze považovat použití nástroje pro detekci plagiátů za náhradu kontroly vyučujícím. Vědomí, že vyučující stejně vypracované dokumenty neče, motivaci k plagiátorství naopak posiluje.

186 Oddělené od samotného detektoru plagiátů je zejména z organizačního hlediska, protože takový systém má své výhody i pokud je používán bez detektoru.

187 Domníváme se, že v dnešní době lze od studentů toto požadovat i vzhledem k tomu, že přístup k počítači ve volném čase mívají minimálně ve škole i ti studenti, kteří z nějakých důvodů domácí počítač nevlastní. Případně lze toto v konkrétních případech řešit organizačními úpravami např. vyčlenění některých kapacit právě pro tyto účely.

dle standardních a předem daných procedur včetně postihů. Kromě represivního účinku potom vědomí existence detektoru a reálnosti možných postihů (spolu s již takto uzavřenými případy) pak budou mít také další preventivní (tentokrát odrazující) efekt.

Cílem celého takového programu je poučit studenty o tom, co je to plagiátorství a následně jej od něj odradit tím, že minimalizujeme jeho výhodnost (časovou úsporu) a maximalizujeme jeho rizika (odhalení, postižení).

## 9.2 Požadavky na nástroj ODPUST

Při formulaci požadavků budeme vycházet z toho, že instituce plánuje použít nástroj ODPUST v konečném důsledku v rámci podobného programu jako je PROPUST, popisovaný výše. Není přitom nutné, aby veškeré požadavky byly splněny okamžitě, předpokládá se inkrementální vývoj a postupné zapojování jednotlivých modulů. Požadavky uvádíme v pořadí předpokládané důležitosti a termínu implementace.

- Nástroj bude schopen provádět intrakorpální detekci
- Nástroj bude schopen korektně porovnat dokumenty rozdílné délky (tj. bude pracovat minimálně se symetrizovanou asymetrickou metrikou)
- Nástroj bude korektně pracovat s dokumenty v českém jazyce.
- Nástroj bude přehledně zobrazovat výsledky porovnání včetně detailního zobrazení shodných částí
- Nástroj bude schopen provádět extrakorpální detekci vůči interní databázi uložených dokumentů.
- Nástroj bude schopen zpracovat<sup>188</sup> kromě čistě textových také dokumenty minimálně ve formátu DOC.
- Nástroj bude schopen získávat dokumenty s odpovídajícími metadaty<sup>189</sup> ze systému pro správu kurzů/odevzdávání elektronických dokumentů.
- Uložené dokumenty budou popsány metadaty aby bylo možné rychle vybrat vhodnou podmnožinu pro porovnání.
- Nástroj bude schopen provádět extrakorpální detekci vůči Internetovým zdrojům a to zejména českým.
- Nástroj bude mít přehledné rozhraní, s předdefinovanými parametry pro konkrétní typy úloh tak, aby je snadno zvládli vyučující.
- V případě nalezení podezřelého dokumentu bude nástroj schopen automatizovaně informovat nejen příslušného vyučujícího, ale i další pověřenou osobu<sup>190</sup>.

---

<sup>188</sup> Či automatizovaně převést na formu, kterou je schopen zpracovat.

<sup>189</sup> Např. autor, odpovědný vyučující, téma, ročník, ...

<sup>190</sup> Například výchovného poradce, zástupce ředitelky, garanta předmětu atp. Jde o opatření k tomu, aby bylo možno zamezit případnému „utulání“ záležitosti, pokud je prioritou školy právě boj proti plagiátům.

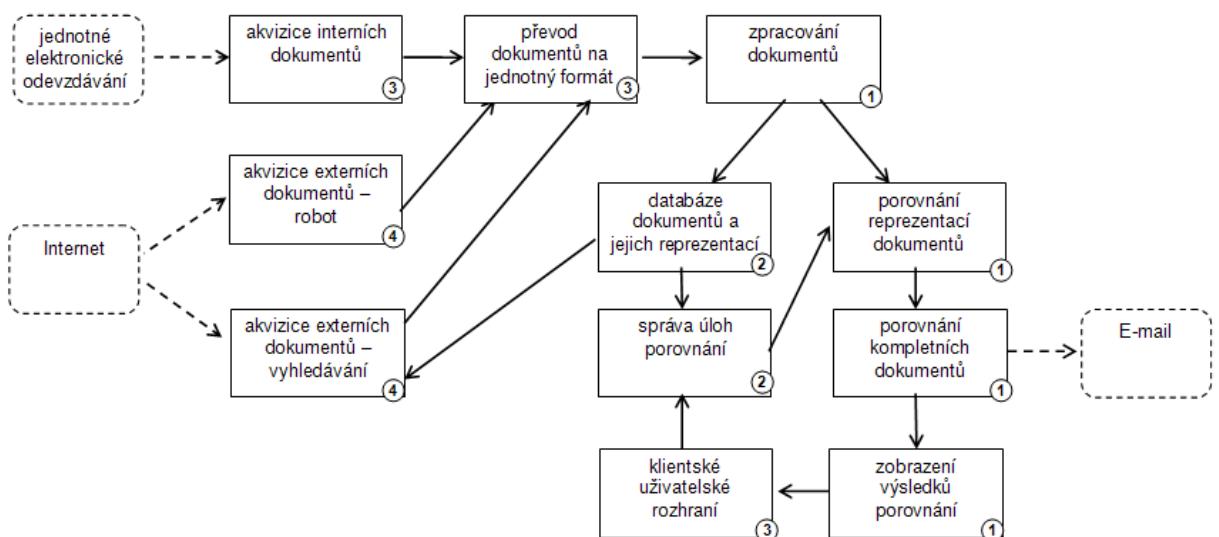
### 9.3 Koncept návrhu systému ODPUST

Předpokládáme, že systém bude koncipován jako modulární v konečné fázi fungující s klasickou třívrstvou architekturou oddělující funkci, data a rozhraní.

Vzhledem k univerzálnosti nasazení a úzké vazbě na internetové prostředí a také proto, že nevyžaduje žádné složité a interaktivní formulářové rozhraní, bychom navrhovali implementaci klientského rozhraní formou webové aplikace.

Vlastní transformační a porovnávací logika by se nám potom, zejména pokud je silný požadavek na rychlé zpracování velkého množství dat, jevila vhodnější implementovat samostatně v některém rychlém prostředí a spouštět dávkově na serveru<sup>191</sup>.

Na základě předchozích požadavků bychom navrhli systém rozčlenit na následující moduly. Jejich vzájemné vazby a vazby na externí systémy jsou znázorněny na obrázku 29. Čísla v pravých dolních rozích jednotlivých modulů na obrázku odpovídají zařazení funkcionality daného modulu jednotlivým předpokládaným vývojovým inkrementům.



Obrázek 29: Schéma rozdělení systému ODPUST na moduly a jejich vazby a vazby na okolí

#### 9.3.1 Jádro systému

Jádrem systému (na obrázku 29 označeno jako inkrement jedna) je porovnání dokumentů. Zde vzhledem k tomu, že se bude při extrakorpálním vyhledávání porovnávat vůči velkému množství dokumentů mimo korpus, doporučujeme pracovat primárně pouze s reprezentacemi dokumentů, jak to bylo zmiňováno v kapitole 5.2. Kvůli požadavku na přehledné zobrazení shodných částí ale bude toto porovnání sloužit pouze jako předvýběr<sup>192</sup>. Takto odhalené podezřelé dokumenty budou dále porovnány kompletně.

<sup>191</sup> Ideálně samozřejmě v C, případně s obětováním části výkonu ve prospěch rychlosti vývoje a snadnosti údržby i Java. Vše samozřejmě za předpokladu použití efektivních a optimalizovaných algoritmů.

<sup>192</sup> Jehož předvýběr zase bude proveden na základě metadat dokumentů.

Z toho plyne, že bude nutné uchovávat kromě krátké reprezentace pro první fázi porovnání také kompletní texty dokumentů. Alternativně je možné je pro následné porovnání kompletního obsahu načítat z externího zdroje (pokud jsou například uchovávány v systému pro jednotné elektronické odevzdávání). Vzhledem k tomu, že zkrácené reprezentace dokumentů nebudou představovat hlavní nárok na paměťový prostor, je možné zvýšit očekávanou úplnost případně bezpečnost a pracovat s trochu větším poměrem než doporučované 1 %. Alternativně lze uvažovat o práci se základními jednotkami proměnlivé délky.

Jako základní jednotku bychom vzhledem ke snadnosti implementace a v praxi poměrně častému použití volili V<sub>3-5a</sub> tedy překrývající se trojice až pětice slov. Před zpracováním je rovněž vhodné odstranit nebo převést minimálně některé interpunkční znaky (např. české uvozovky, případně čárky a podobně). Prozatím nebudeme předpokládat využití tezauru ani jiné lexikální transformace.

Pro práci s kompletním obsahem dokumentů se nabízí variace na stejné téma, tedy porovnání obsahů na základě stejných základních jednotek, tentokrát všech. Výhodou by jistě byla možnost zapojení již vytvořeného modulu pro zpracování. Navíc v případě, že by se nešetřilo místem a ukládaly se kromě obsahu dokumentu také jeho veškeré základní jednotky, bylo by z nich možné vybírat různými způsoby ty pro porovnání reprezentací a výrazně tak zvýšit bezpečnost<sup>193</sup> a škálovatelnost<sup>194</sup> systému.

Ve výsledcích porovnání předpokládáme, že budou zobrazeny dokumenty s podobností nad výstražnou mez (výchozí standardně 5–15 %) a to podle míry podobnosti. A to nejlépe tak, že kromě dvojic podobných dokumentů bude také možné zobrazit agregovanou podobnost<sup>195</sup>. V detailním zobrazení je pochopitelný požadavek na vyznačení shodných pasáží, nejlépe provázaných hypertextovými odkazy.

### **9.3.2 Databáze dokumentů a obecně datové základna**

Jak již bylo uvedeno výše, k tomu, aby bylo v rozumném čase možné zvládnout porovnání velmi velkého počtu dokumentů je potřeba kromě efektivního porovnávacího algoritmu a vhodné datové reprezentace také předem vyloučit ty dokumenty, které již z hlediska svého obsahu téměř nemají šanci být podobné, a tím výrazně snížit počet nutných porovnání. K tomu je potřeba mít dostatečně efektivní systém popisující dokumenty na základě jejich obsahu, zadání, data odevzdání a podobně. V námi navrhovaném systému počítáme s univerzálním systémem metadat pro popis dokumentů.

Univerzálně jej navrhujeme proto, že různé instituce mají různé potřeby a pracují s různými dokumenty. Nespecifikujeme proto konkrétní sadu metadat, ale spíše univerzální a pružný systém, který je snadno konfigurovatelný.

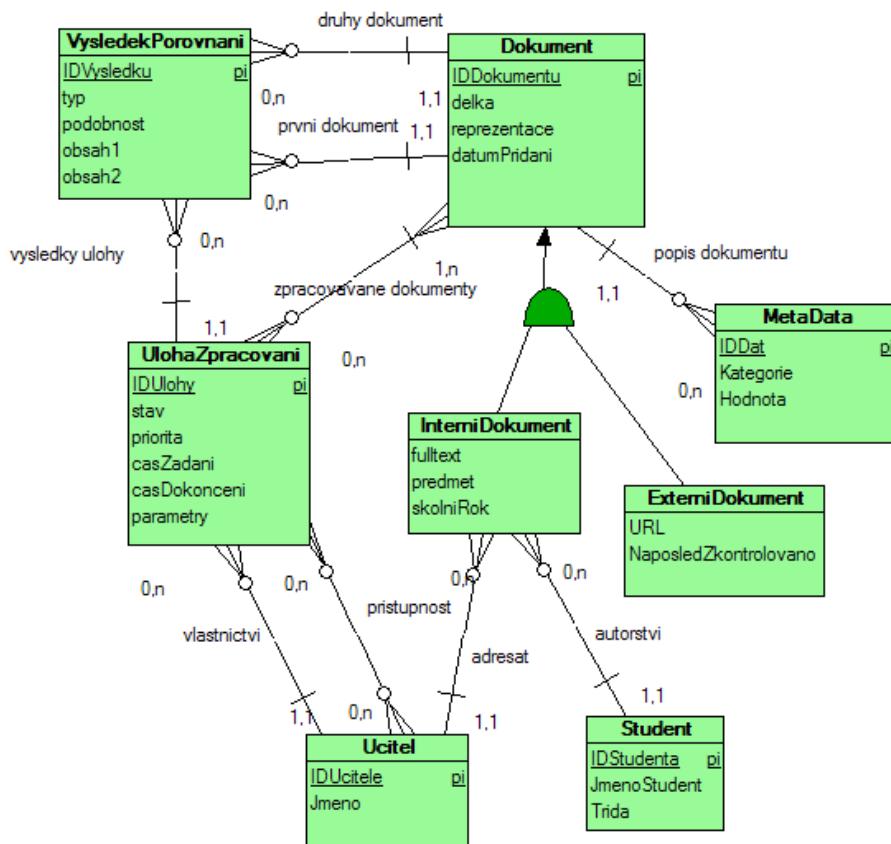
Na obrázku 30 je zjednodušený konceptuální datový model celého systému ODPUST. Pro přehlednost v něm nejsou zahrnuty veškeré (z hlediska funkcionality méně podstatné) vazby a entity. Chybí například ty, které se týkají pouze akvizice a převodu, případně číselníky různých typů metadat a také detailní parametry úlohy porovnání. Velmi zjednodušené nebo zcela vynechané jsou také

---

193 V Brinově pojednání.

194 Např. při porovnání většího počtu dokumentů zvolit méně základních jednotek jako reprezentaci, čímž sice můžeme dosáhnout nižší úplnosti, ale můžeme výrazně snížit výpočetní nároky.

195 Tak, aby např. komplát složený pouze z pěti odstavců, kde každý byl zkopirován z jednoho dokumentu byl označen jako 100 % podobný a nikoliv pouze jako pětkrát 20 % podobný.



Obrázek 30: Konceptuální datový model části systému ODPUST

ty entity, u kterých se předpokládá, že jsou již v informačním systému instituce zahrnuty jinde (zejména studenti, učitelé, třídy atp.). Předpokládáme také určitá další zjednodušení<sup>196</sup>.

### 9.3.3 Akvizice dokumentů

Úkolem modulu pro akvizici interních dokumentů je transparentně získávat dokumenty ze systému pro jednotné elektronické odevzdávání. Kromě samotného fyzického získání je musí opatřit pokud možno automaticky příslušnými metadaty tak, aby mohly být zařazeny do databáze a efektivně prohledávány.

Následuje předání dokumentů k převodu do jednotného (textového) formátu. Zde se předpokládá využití nástrojů třetích stran pro konverzi zejména z DOC a RTF. Výhledově je nutné počítat také s novým kancelářským formátem OpenXML a ODF. Předpokládáme také schopnost konvertovat HTML, která bude využita při získávání dokumentů z Internetu.

<sup>196</sup> Například, že primárně neukládáme plné texty externích dokumentů, že externí dokumenty pochází pouze z Internetu a nikoliv například z učebnic nebo jiných zdrojů, že žáci na svých dokumentech nepracují v týmech (autorem není více studentů) a dokumenty jsou vždy pro jeden konkrétní předmět a podobně. Samozřejmě je možné poměrně snadno odstranění těchto předpokladů zapracovat do návrhu. Přistoupili jsme tu na ně zejména proto, abychom model maximálně zpřehlednili a zaměřili se na podstatu problematiky a nikoliv na jednotlivé dílčí detaily.

Ty je možné získávat dvěma způsoby. Jeden předpokládá integraci webového robota, který bude autonomně procházet prostředí webu (zejména českého) a na základě stanovených pravidel<sup>197</sup> získávat vhodné dokumenty. U takto získávaných dokumentů by mohl nastat problém s přiřazením odpovídajících metadat. V takovém případě navrhujeme tyto dokumenty buď ručně alespoň zhruba označit a nebo je řadit neoznačené do společné skupiny (např. WEB) a následně umožnit porovnání buď i vůči této velké skupině, nebo bez ní.

Protože takto získaných dokumentů z internetu bude pravděpodobně velké velmi množství, navrhujeme do databáze ukládat pouze jejich reprezentace a odkaz na původní zdroj. V případě potřeby porovnat úplný obsah pak bude možné na vyžádání dokument znova získat a porovnání provést.

Druhou možností je získávat dokumenty na Internetu nikoliv preventivně a dopředu, ale až na základě studenty odevzdaných dokumentů s pomocí standardních vyhledávačů. Není možné efektivně vyhledávat všechny fráze všech dokumentů. V takovém případě zřejmě jako záchytné body nejlépe poslouží nějaká heuristická analýza stylu. Například velmi dlouhé věty, věty s neobvyklými slovy, neobvykle dlouhá slova atp. Na tomto základě je možné vybrat několik nejpodezřelejších frází v dokumentu a použít je pro vyhledání ve standardním fulltextovém vyhledávači, který dobře indexuje český web<sup>198</sup>. V takovém případě je práce s metadaty (pokud bude takto získaný dokument ukládán) jednodušší, můžeme částečně<sup>199</sup> použít ta, která jsou součástí onoho odevzdaného dokumentu, na jehož základě byl tento výsledek nalezen.

Možností je i kombinace obou přístupů po vyhledání možného zdroje vyslat na daný web robota aby jeho obsah převedl do databáze jako potenciální zdroje dalšího plagiátorství.

### **9.3.4 Hlavní funkce jednotlivých modulů**

Zde si pro přehlednost shrneme hlavní uvažovanou funkcionalitu celého systému a to, jak ji v tomto návrhu rozčleňujeme do jednotlivých modulů.

- Správa úloh
  - vytvořit úlohu zpracování
  - přidat dokumenty do úlohy zpracování na základě metadat
  - přidat dokumenty do úlohy přímo
  - nastavit parametry úlohy
  - správa a prioritizace úloh (administrátor)
  - automaticky spouštět naplánované úlohy
- Akvizice interních dokumentů
  - získat jednorázově dokumenty ze systému pro jednotné elektronické odevzdávání
  - zjišťovat pravidelně nově přidané dokumenty v systému pro jednotné elektronické odevzdávání

<sup>197</sup> Například podíl textu, různá klíčová slova i fráze (např. čtenářský deník), příslušnost k předem dané skupině domén a podobně.

<sup>198</sup> Nabízí se Google (vhodný navíc i pro zahraniční materiály), případně lépe skloňující Morfeo, Jyxo a nebo Seznam.

Z taktického hlediska se jeví jako ideální používat stejně vyhledávače, které při své práci používají i studenti.

<sup>199</sup> Samozřejmě zejména ta obsahová, nikoliv o autorovi nebo jeho vyučujícím.

- Akvizice externích dokumentů robot
  - nastavit pravidla pro robota
  - procházet web dle pravidel
  - získávat obsahu webu
- Akvizice externích dokumentů vyhledávání
  - nastavit parametry pro vyhledávání
  - extrahovat vhodné fráze z dokumentů
  - vyhledávat fráze s pomocí vyhledávače
  - získávat obsah nalezených výsledků
- Převod dokumentů na jednotný formát
  - automaticky zjistit formát dokumentu
  - převést DOC na plaintext
  - převést RTF na plaintext
  - automaticky zjistit použitou znakovou sadu
  - převést na výchozí znakovou sadu
  - převést HTML na plaintext
- Zpracování dokumentů
  - spravovat standardní třídy pravidel předzpracování
  - aplikovat pravidla předzpracování
  - získat základní jednotky porovnání pro dokument
  - vybrat základní jednotky pro reprezentaci
- Porovnání reprezentací dokumentů
  - spravovat standardní třídy pravidel porovnání reprezentací
  - zpracovat reprezentace zvolených dokumentů
  - vytvořit souhrnné výsledky
  - předat podezřelé dokumenty k úplnému porovnání
- Porovnání kompletních dokumentů
  - spravovat standardní třídy pravidel porovnání kompletních dokumentů
  - zpracovat obsah (resp. všech zákl. jednotky) zvolených dokumentů
  - vytvořit detailní výsledky (vyznačení přesných shod)
  - informovat pověřené osoby
- Zobrazení výsledků porovnání
  - zobrazení informací o úloze

- zobrazení souhrnných výsledků a provázání na detailní výsledky
- zobrazení detailních výsledků
- Databáze dokumentů a jejich reprezentací
  - uložit dokumenty (vč. reprezentace a metadat)
  - spravovat u dokumentů metadata hromadně i individuálně
  - spravovat číselníky a kategorií metadat
  - vybrat dokumenty podle metadat

#### 9.4 Shrnutí

Výše nastíněný návrh konceptu systému pro odhalování plagiátů u studentů ODPUST je pokusem o modulárně řešený integrovaný systém detekce plagiátů pro českou vzdělávací instituci střední velikosti. Od nejlepších existujících zdarma dostupných řešení se odlišuje zejména tím, že se jedná o extrakorpální centrální systém, který je tak možno nasadit a zapojit do procesu výuky v celé instituci. Od nejlepších komerčně dostupných služeb se odlišuje tím, že je navrhován tak, aby byl zcela zaměřen na detekci a zejména plně vyhověl požadavkům českého prostředí. Na to tyto služby často nejsou zacíleny a jejich nástroje nezřídka selhávají a nedosahují očekávaných výsledků. Přitom náklady na vývoj a pětiletý provoz takového systému<sup>200</sup> odhadujeme na nižší úrovni, než náklady na licenci u některé z předních zahraničních komerčních služeb. Přitom je potřeba mít na mysli skutečnost, že již existují velmi výkonné intrakorpální lokálně použitelné nástroje, které jsou dostupné zdarma případně i se zdrojovým kódem a další publikované informace, na kterých je možné detaily jádra systému založit.

Naším cílem bylo zejména navrhnout systém, který umožní efektivní extrakorpální detekci v rámci interní databáze a také práci s externími dokumenty a jejich získávání z Internetu. Rozhodně přitom doporučujeme případný takovýto nástroj zavádět pouze jako jednu ze součástí integrovaného komplexu systémových informatických i neinformatických opatření například tak, jak bylo ukázáno na příkladu modelového programu PROPUST.

<sup>200</sup> Zejména při, v tomto prostředí poměrně obvyklém, vývoji vlastními silami, případně spojenými silami více institucí s podobnými požadavky.

## Závěr

V této diplomové práci jsem čtenáře poměrně podrobně uvedl do problematiky automatické detekce plagiátů. Popsal jsem rozdíly mezi různými přístupy k ní a situace, pro které jsou ty které přístupy vhodné. Vysvětlil jsem rozdíly mezi různými obecnými typy nástrojů a jimi používaných metrik. Nastínil jsem také teoretické možnosti srovnání jejich výkonnosti. Představil jsem některé netradiční přístupy k detekci, které nejsou založeny na analýze obsahu dokumentů. Naznačil a popsal jsem několika autory sdílený princip fungování výkonného a úsporného nástroje pro detekci plagiátů dokumentů ve volném textu vhodného i pro extrakorpální použití. Uvedl jsme také některé výhrady proti němu a principy alternativního řešení a některé možné přístupy k optimalizaci.

V další části jsem podrobil testování a porovnání základních uživatelských parametrů poměrně velkou a reprezentativní část dostupných existujících nástrojů a to jak komerčních, tak zdarma dostupných. Z výsledků je vidět, že trh ještě není zcela konsolidovaný a mnoho nástrojů není příliš vhodných pro použití v českém prostředí.

V poslední kapitole jsem se pokusil navrhnout koncept nástroje pro detekci plagiátů ODPUST vhodného pro využití v menší až střední české vzdělávací instituci. Navrhovaný systém je rozdělen do modulů a ty společně tvoří skupiny vhodné pro postupnou implementaci v rámci inkrementálního vývoje. Jádro systému je koncipováno tak, aby bylo možné jako základ využít existující a osvědčené nástroje nebo alespoň principy popisované v předchozích kapitolách. Součástí návrhu je i jeho začlenění do komplexnějšího programu prevence plagiátů PROPUST, který má za cíl boj proti plagiátorství na více frontách a mimo jiné umožňuje snížit podíl plagiátů i s méně dokonalým (a tedy levnějším) softwarovým nástrojem a při zachování větší důvěry studentů.

Tato práce by měla být využitelná pro čtenáře, kteří vážně uvažují o zavedení nástroje pro detekci plagiátů a to jak některého z existujících, tak případně i o implementaci vlastního. K případnému dalšímu prohloubení a rozšíření zkoumané problematiky, mohu poskytnout několik námětů, kterými jsem se v této práci z různých důvodů nezabýval, ale mohly by přinést do této problematiky další zajímavé poznatky. Jedním z nich (jistě nepříliš snadným) by mohl být výzkum či odhad skutečné míry plagiátorství (případně povědomí o něm a o informační etice obecně) v nějaké konkrétní místní instituci. Alternativně by mohlo být zajímavé porovnat výsledky získané dotazováním a skutečnou analýzou reálných dokumentů. Jako další námět předkládám analýzu boje s plagiátorstvím v místních podmírkách v různých institucích (včetně např. popisu tam používaných vlastních nástrojů) a detailní návrh komplexního systému prevence (například rozšířením a prohloubením programu PROPUST). A konečně třetím okruhem by mohla být detailní analýza konkrétních optimalizovaných algoritmů pro detekci plagiátů případně spojená s detailním návrhem či dokonce implementací systému pro detekci plagiátů (například na bázi systému ODPUST).

## Rejstříky vložených tabulek a obrázků

### Rejstřík tabulek

Přehled kriterií pro klasifikaci nástrojů pro detekci plagiátů z pohledu uživatele.....	23
Shrnutí vlastností základních metrik.....	40
Doporučené základní jednotky porovnání u různých autorů.....	46
Vlastnosti různých možnosti volby základních jednotek porovnání (převzato a upraveno z [Brin1995]).....	48
Srovnání výsledků porovnání celých dokumentů a jejich reprezentace (převzato z [Heinze1996])..	55
True/false positives/negatives.....	71
Vybrané nástroje a služby a jejich základní charakteristika.....	79
Charakteristika služby CatchItFirst.....	81
Charakteristika nástroje Glatt Plagiarism Screening Program.....	82
Charakteristika nástroje CopyCatch Gold.....	84
Charakteristika nástroje EVE2.....	86
Charakteristika nástroje Plagiarism Finder.....	88
Vývoj ročních nákladů na službu TurnItIn.com pro University of Kansas (podle [Dye2006]).....	94
Charakteristiky služby TurnItIn.com.....	94
Charakteristika nástroje WCopyFind.....	97
Charakteristika nástrojů PRAISE a VAST.....	103
Charakteristika služby JPlag.....	105
Charakteristika nástroje Sherlock.....	108
Výsledky intrakorpálních testů.....	109
Výsledky extrakorpálních testů.....	110

### Rejstřík obrázků

Schéma fungování detekce plagiátů pomocí neviditelného značkování.....	25
Princip detekce a prevence plagiátorství v prostředí APE.....	28
Porovnání průběhu funkcí symetrických metrik.....	35
Porovnání průběhu funkcí symetrických a asymetrických metrik.....	37
Porovnání průběhu základních symetrických a symetrizovaných metrik.....	39
Schema přípravy reprezentace dokumentu.....	43
Vztah mezi prvkem, jednotkou a dokumentem v kontextu porovnání reprezentací dokumentů.....	45
Shoda počtu základních jednotek pro uložení pro 10 000 základních jednotek v souboru.....	51

Základní princip urychlení algoritmů pro vyhledávání řetězců.....	58
Rozdíl mezi nejdelším společným (pod)řetězcem a nejdelší společnou podsekvincí.....	59
Hrubá představa porovnání s pomocí komprese.....	60
Srovnání počtu porovnání 1:1 při využití metody "nových" a "starých" dokumentů a bez ní.....	62
Úrovně plagiátorství ve zdrojových kódech dle Faidhi a Robinson (převzato z [Noh2003]).....	65
Transformace plagiátů ve zdrojových kódech dle Jonesa (převzato z [Lancaster2005a]).....	65
grafické porovnání dvou dokumentů (převzato z [Ducasse1999]).....	68
Prostředí služby CatchItFirst.....	80
Prostředí nástroje CopyCatch Gold.....	83
Prostředí nástroje EVE2.....	85
Prostředí nástroje Plagiarism Finder.....	87
Prostředí služby MyDropBox.com.....	90
Prostředí služby TurnItIn.com.....	93
Prostředí nástroje WCopyFind.....	96
Prostředí nástroje Pl@giarism.....	98
Prostředí nástroje Ferret.....	100
Prostředí nástrojů PRAISE, VAST a FreeStyler.....	103
Prostředí služby JPlag.....	105
Prostředí nástroje Sherlock.....	107
Komplexní program prevence plagiátů a úloha nástroje pro detekci v něm.....	113
Schéma rozdělení systému ODPUST na moduly a jejich vazby a vazby na okolí.....	115
Konceptuální datový model části systému ODPUST.....	117

## Přehled literatury a použitých zdrojů

- [Aintegrity2006] Academic Integrity : Plagiarism and Related Issues [online]. 2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.hpcnet.org/peru/facultysenate/academicintegrity>>.
- [Arts2003] ARTS, Mark, GEUS, Bas: Plagiarism Software Evaluation for ELEUM [online]. 2003 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.fdewb.unimaas.nl/eleum/plagiarism/plagiarism.htm>>.
- [Auer2001] AUER, Nicole J., KRUPAR, Ellen M.: Mouse click plagiarism : The role of technology in plagiarism and the Librarian's role in combating it. *Library Trends*. 2001, 49, 3, s. 415.
- [wcopyfind] BLOOMFIELD, Lou.: The plagiarism resource site [online]. 2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://plagiarism.phys.virginia.edu/>>.
- [Braumoeller2001] BRAUOMELLER, Bear F., GAINES, Brian J.: Actions Do Speak Louder than Words : Deterring plagiarism with the use of plagiarism-detection software. *Political Science & Politics*. 2001, vol. 34, no. 4, s. 835.
- [Brin1995] BRIN, Sergey, DAVIS, James, GARCIA-MOLINA, Hector: Copy Detection Mechanism for Digital Documents. In *Proceedings of the ACM SIGMOD Conference*. [s.l.] : [s.n.], 1995. Dostupný z WWW: <[http://graphics.stanford.edu/~jedavis/projects/database/copy\\_detection.ps](http://graphics.stanford.edu/~jedavis/projects/database/copy_detection.ps)>.
- [Broder1997] BRODER, Andrei Z.: On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*. [s.l.] : [s.n.], 1997. s. 21. ISBN 0-8186-8132-2.
- [Burd2002] BURD, Elizabeth, BAILEY, John: Evaluating Clone Detection Tools for Use during Preventative Maintenance. In *Proceedings of the Second IEEE International Workshop on Source Code Analysis and Manipulation (SCAM'02)*. [s.l.] : [s.n.], 2002. s. 36.
- [Byrne2004] BYRNE, Siobhán, et al.: A technique for measuring plagiarism in automated & distance learning environments. In *Proceedings of the winter international symposium on Information and communication technologies*. Cancun, Mexico : [s.n.], 2004. s. 1.
- [eve2] CaNexus: EVE2 Plagiarism Detection System [online]. [2005] [cit. 2006-07-06]. Dostupný z WWW: <<http://www.canexus.com/>>.
- [cise] Centre for Interactive Systems Engineering - Plagiarism Prevention and Detection: *Tools* [online]. [2003] [cit. 2006-07-06]. Dostupný z WWW: <<http://cise.lsbu.ac.uk/tools.html>>.
- [copycatch] CFL Software Development: The CopyCatch Suite - 2006 [online]. 2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.copycatchgold.com/>>.
- [Charras1997] CHARRAS, Christian, LECROQ, Thierry: Exact string matching algorithms [online]. 1997 [cit. 2006-07-06]. Dostupný z WWW: <<http://www-igm.univ-mlv.fr/~lecroq/string/index.html>>.
- [Chen2003] CHEN, Xin, et al.: Shared Information and Program Plagiarism Detection. [online]. 2003 [cit. 2006-07-06]. Dostupný z WWW: <<http://monod.uwaterloo.ca/papers/04sid.pdf>>.

- [Clough2000] CLOUGH, Paul: Plagiarism in Natural and Programming Languages: An Overview of Current Tools and Technologies. *Research Memoranda*, CS-00-05, Department of Computer Science, University of Sheffield, UK, 2000. [online] <<http://ir.shef.ac.uk/cloughie/papers/plagiarism2000.pdf>>.
- [Collins2005] COLLINS, Mary Elizabeth, AMODEO, Maryann: Responding to plagiarism in schools of social work: Considerations and Recomendations. *Journal of Social Work Education*. 2005, vol. 41, is. 3, s. 527.
- [Culwin2001] CULWIN, Fintan, MACLEOD, Anna, LANCASTER, Thomas: Source code plagiarism in UK HE Computing Schools. In *2nd annual conference of the LTSN centre for Information and Computer Sciences*. [s.l.] : [s.n.], 2001. Dostupný z WWW: <<http://www.ics.heacademy.ac.uk/Events/conf2001/papers/Culwin.htm>>. ISBN 0-9541927-0-2.
- [Daly2005] DALY, Charlie, HORGAN, Jane: Patterns of plagiarism. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*. St. Louis, Missouri, USA : [s.n.], 2005. s. 383. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=1047473&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17231748&CFTOKEN=52573441](http://portal.acm.org/ft_gateway.cfm?id=1047473&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17231748&CFTOKEN=52573441)>. ISSN 0097-8418.
- [DEE] DEE, Hannah M.: *My, this page is all stolen!* [online]. - [cit. 2006-07-06]. Dostupný z WWW: <<http://www.comp.leeds.ac.uk/hannah/CandIT/naughty.html>>.
- [Donaldson1981] DONALDSON, John L., LANCASTER, Ann-Marie, SPOSATO, Paula H.: A plagiarism detection system. In *Proceedings of the 12th SIGCSE symposium on Computer science education*. [s.l.] : [s.n.], 1981. s. 21. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=800955&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17235946&CFTOKEN=71111834](http://portal.acm.org/ft_gateway.cfm?id=800955&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17235946&CFTOKEN=71111834)>. ISBN 0-89791-036-2.
- [Ducasse1999] DUCASSE, Stéphane, RIEGER, Matthias, DEMEYER, Serge: A Language Independent Approach for Detecting Duplicated Code. In *15th IEEE International Conference on Software Maintenance (ICSM'99)*. [s.l.] : [s.n.], 1999. s. 109.
- [Dye2006] DYE, Dustin: KU cancels Turnitin [online]. 2006 [cit. 2007-03-10]. Dostupný z WWW: <[http://reporting.journalism.ku.edu/fall06/bradford-noland/2006/09/ku\\_cancels\\_turnitin.html](http://reporting.journalism.ku.edu/fall06/bradford-noland/2006/09/ku_cancels_turnitin.html)>.
- [Finkel2002] FINKEL, Raphael A., et al.: Signature extraction for overlap detection in documents. In *Proceedings of the twenty-fifth Australasian conference on Computer science - Volume 4*. Melbourne, Victoria, Australia : [s.n.], 2002. s. 59. Dostupný z WWW: <[http://portal.acm.org/proceedings/ft\\_gateway.cfm?id=563809&type=pdf&coll=portal&dl=ACM&CFID=1452886&CFTOKEN=49967946](http://portal.acm.org/proceedings/ft_gateway.cfm?id=563809&type=pdf&coll=portal&dl=ACM&CFID=1452886&CFTOKEN=49967946)>. ISBN 1445-1336. ISSN 0-909925-82-8.
- [Gitchell1999] GITCHELL, David, TRAN, Nicholas.: Sim: A Utility For Detecting Similarity in Computer Programs. In *The proceedings of the thirtieth SIGCSE technical symposium on Computer science education*. New Orleans, Louisiana, United States : [s.n.], 1999. s. 266. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=299783&type=pdf&coll=ACM&dl=ACM&CFID=15151515&CFTOKEN=6184618](http://portal.acm.org/ft_gateway.cfm?id=299783&type=pdf&coll=ACM&dl=ACM&CFID=15151515&CFTOKEN=6184618)>. ISSN 0097-8418.
- [Glatt1999] Glatt Plagiarism Services, Inc.: *Welcome to Glatt Plagiarism Services* [online]. [cit. 2006-07-06]. Dostupný z WWW: <<http://www.plagiarism.com/>>.

- [Glod2006] GLOD, Maria: Students Rebel Against Database Designed to Thwart Plagiarists. *The Washington Post*. 2006-09-22, Page A01. Dostupný z WWW: <<http://www.washingtonpost.com/wp-dyn/content/article/2006/09/21/AR2006092101800.html>>.
- [Gregory2002] GREGORY, John, STRUKOV, Andrei: Ensuring Academic Integrity in the Age of the Internet : Evaluating a web-based analytic tool. In *Proceedings of the International Conference on Computers in Education (ICCE'02)*, 2002.
- [Haris2004] HARRIS, Robert: *Anti-Plagiarism Strategies for Research Papers* [online]. November 17, 2004 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.virtuallsalt.com/antiplag.htm>>.
- [Hayes2005] HAYES, Niall, INTRONA, Lucas: *Systems for the Production of Plagiarists: Developing countries and use of plagiarism detection systems in UK universities* [online]. 9.4.2005 [cit. 2006-07-06]. Prezentace. Dostupný z WWW: <<http://www.lums.lancs.ac.uk/files/sdaw/5708/download/>>.
- [Heinze1996] HEINTZE, Nevin: *Scalable Document Fingerprinting: Extended Abstract*. [online]. 1996 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.cs.cmu.edu/afs/cs/user/nch/www/koala/main.html>>.
- [Howard2006] HOWARD, R. M.: *Turnitin dollars* [online]. 2007 [cit. 2007-03-10]. Dostupný z WWW: <[http://www.rmhoward.net/2007/03/turnitin\\_dollars.html](http://www.rmhoward.net/2007/03/turnitin_dollars.html)>.
- [turnitin] IParadigms, LLC.: *TurnItIn.com* [online]. c2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.turnitin.com>>.
- [Jones2001] JONES, Edward L.: Metrics based plagiarism monitoring. In *Proceedings of the sixth annual CCSC northeastern conference on The journal of computing in small colleges*. Middlebury, Vermont, United States : [s.n.], 2001. s. 253. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=378727&type=pdf&coll=ACM&dl=ACM&CFID=15151515&CFTOKEN=6184618](http://portal.acm.org/ft_gateway.cfm?id=378727&type=pdf&coll=ACM&dl=ACM&CFID=15151515&CFTOKEN=6184618)>.
- [j plag] *JPlag - Detecting software plagiarism* [online]. [cit. 2006-07-06]. Dostupný z WWW: <<https://www.ipd.uni-karlsruhe.de/j plag>> nebo <<http://www.j plag.de>>.
- [Kang2006] KANG, NamOh, GELBUKH, Alexander, HAN, SangYong: PPChecker: Plagiarism Pattern Checker in Document Copy Detection. In *Lecture Notes in Computer Science*. [s.l.] : Springer Berlin / Heidelberg, 2006. s. 661. Volume 4188/2006. Dostupný z WWW: <<http://springerlink.metapress.com/content/y6171u806310234k>>. ISSN 0302-9743.
- [NLP] Laboratoř zpracování přirozeného jazyka: *Frekvence písmen, bigramů, trigramů, délka slov* [online]. 2004 [cit. 2006-07-06]. Dostupný z WWW: <[http://nlp.fi.muni.cz/nlp/NlpCz/Frekvence\\_pismen\\_bigramu\\_trigramu\\_delka\\_slov.html](http://nlp.fi.muni.cz/nlp/NlpCz/Frekvence_pismen_bigramu_trigramu_delka_slov.html)>.
- [Lancaster2005] LANCASTER, Thomas, CULWIN, Fintan: *Classifications of plagiarism detection engines: (internal second draft)*. [online] 2005 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.ics.heacademy.ac.uk/italics/Vol4-2/Plagiarism%20-%20revised%20paper.pdf>>.

- [Lancaster2001] LANCASTER, Thomas, CULWIN, Fintan: Towards an error free plagiarism detection process. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education*. Canterbury, United Kingdom : [s.n.], 2001. s. 57. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=377473&type=pdf&coll=portal&dl=ACM&CFID=1452881&CFTOKEN=30300288](http://portal.acm.org/ft_gateway.cfm?id=377473&type=pdf&coll=portal&dl=ACM&CFID=1452881&CFTOKEN=30300288)>. ISBN 1-58113-330-8.
- [Lancaster2005a] LANCASTER, Thomas, TETLOW, Mark. Does automated anti-plagiarism have to be complex? : Evaluating more appropriate software metrics for finding collusion. In *Asclilite 2005: Balance, Fidelity, Mobility: maintaining the momentum?*. 2005. s. 361. Dostupný z WWW: <[http://www.asclilite.org.au/conferences/brisbane05/blogs/proceedings/42\\_Lancaster.pdf](http://www.asclilite.org.au/conferences/brisbane05/blogs/proceedings/42_Lancaster.pdf)>
- [Lyon2004] LYON, Caroline, BARRETT, Ruth, MALCOLM, James: A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector. Plagiarism: *Prevention, Practice and Policies Conference* [online]. 2004 [cit. 2006-07-06]. Dostupný z WWW: <<http://homepages.feis.herts.ac.uk/~comrcml/LyonPaperFerret.pdf>>.
- [Lyon2006] LYON, Caroline. M., BARRET, R., MALCOLM, J. A.: Plagiarism is Easy, but also Easy To Detect. *Plagiary.org* [online]. 2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.plagiary.org/Plagiarism-Is-Easy-To-Detect.pdf>>. ISSN 1559-3096.
- [Malkin2005] MALKIN, Michael, VENKATESAN, Ramarathnam: Comparison of texts streams in the presence of mild adversaries. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research - Volume 44*. Newcastle, New South Wales, Australia : [s.n.], 2005. s. 179. ISBN 1445-1336. ISSN 1-920-68226-0.
- [Mares2005] MAREŠ, Jiří: Elektronické podvádění ve škole. In *Informační a komunikační technologie ve vzdělávání (ICTE2005)*. 2005. Dostupný z WWW: <<http://www.osu.cz/ictc/2005/31.doc>>.
- [plagiarismfinder] Mediaphor AG.: *Plagiarism-Finder* [online]. [2006] [cit. 2006-07-06]. Dostupný z WWW: <<http://www.m4-software.de>>.
- [MeyerZE2006] MEYER ZU EISSEN, Sven, STEIN, Benno: Intrinsic Plagiarism Detection. In *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006. s. 565. Volume 3936/2006. Dostupný z WWW: <<http://springerlink.metapress.com/content/x7x483u1k3970863>>. ISBN 3-540-33347-.
- [Monostori2002a] MONOSTORI, Krisztian: *Efficient computational approach to identifying overlapping documents in large digital collections*. 2002. 227 s. School of Computer Science and Software Engineering, Monash University. PhD thesis. Dostupný z WWW: <<http://www.csse.monash.edu.au/projects/MDR/papers/monostori-thesis2002.pdf>>.
- [Monostori2002] MONOSTORI, Krisztián, et al.: Comparison of Overlap Detection Techniques. In *Proceedings of The 2002 International Conference on Computational Science*. Amsterdam, The Netherlands : [s.n.], 2002. s. 51. Dostupný z WWW: <<http://www.csse.monash.edu.au/projects/MDR/papers/ICCS2002-monostori.pdf>>.
- [Monostori2001] MONOSTORI, Krisztián, ZASLAVSKY, Arkady, SCHMIDT, Heinz: Efficiency of Data Structures for Detecting Overlap in Digital Documents. In *Proceedings of the 24th Australasian Computer Science Conference (ACSC'01)*. 2001. Dostupný z WWW: <<http://csdl.computer.org/dl/proceedings/acsc/2001/0963/00/09630140.pdf>>.

- [Moussiades2005] MOUSSIADES, Lefteris, VAKALI, Athena: PDetect : A Clustering Approach for Detecting Plagiarism in Source Code Datasets. *The Computer Journal*. 2006, vol. 48, no. 6, s. 651.
- [Mulcahy2004] MULCAHY, Sue, GOODACRE, Christine: Opening Pandora's box of academic integrity: Using plagiarism detection software. In *ATKINSON, Roger , et al. Proceedings of the 21st ASCILITE Conference*. 2004. Dostupný z WWW: <<http://www.ascilite.org/conferences/perth04/procs/mulcahy.html>>.
- [Niezgoda2006] NIEZGODA , Sebastian, WAY, Thomas P.: SNITCH: a software tool for detecting cut and paste plagiarism. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. Houston, Texas, USA . 2006. s. 51. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=1121359&type=pdf&coll=GUIDE&dl=GUIDE&CFID=20831081&CFTOKEN=62414774](http://portal.acm.org/ft_gateway.cfm?id=1121359&type=pdf&coll=GUIDE&dl=GUIDE&CFID=20831081&CFTOKEN=62414774)>. ISSN 0097-8418.
- [Noh2003] NOH, Seo-Young, GAIDA, Shashi K.: An XML Plagiarism Detection Model for Procedural Programming Languages. In *Technical Report 03-14, Computer Science*. Iowa State University. : 2003. Dostupný z WWW: <[http://archives.cs.iastate.edu/documents/disk0/00/00/03/32/00000332-00/xpdec\\_03-14.pdf](http://archives.cs.iastate.edu/documents/disk0/00/00/03/32/00000332-00/xpdec_03-14.pdf)>.
- [Ottenstein1976] OTTENSTEIN, K. J.: An algorithmic approach to the detection and prevention of plagiarism. In *ACM SIGCSE Bulletin*. 1976. s. 30. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=382462&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17236600&CFTOKEN=60986640](http://portal.acm.org/ft_gateway.cfm?id=382462&type=pdf&coll=GUIDE&dl=GUIDE&CFID=17236600&CFTOKEN=60986640)>. ISSN 0097-8418.
- [Oxford] Oxford University Press: *The English Language - Frequently Asked Questions : How many words are there in the English language?* [online]. c2007 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.askoxford.com/asktheexperts/faq/aboutenglish/numberwords>>.
- [ferret] Plagiarism Detection Research Group University of Hertfordshire: *Ferret* [online]. 2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://homepages.feis.herts.ac.uk/~pdgroup/>>.
- [Prechelt2002] PRECHELT, Lutz, MALPOHL, Guido, PHILIPPSEN, Michael: Finding Plagiarism among a Set of Programs with JPlag. In *Journal of Universal Computer Science*, Vol. 8, no. 11. 2002. s. 1016.
- [Schleimer2003] SCHLEIMER, Saul, WILKERSON, Daniel S., AIKEN, Alex: Winnowing: Local Algorithms for Document Fingerprinting. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 2003. s. 76. Dostupný z WWW: <<http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf>>.
- [mydropbox] Sciworth Inc.: *MyDropBox* [online]. c2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.mydropbox.com>>.
- [Shivakumar1998] SHIVAKUMAR, Narayanan, GARCIA-MOLINA, Hector: Finding near-replicas of documents on the web. In *Proceedings of Workshop on Web Databases (WebDB'98)*. 1998. Dostupný z WWW: <<http://dbpubs.stanford.edu/pub/showDoc.Fulltext?lang=en&doc=1998-31&format=pdf&compression=&name=1998-31.pdf>>.
- [plagiarismtk] SPAN, Georges: *Pl@giarism: a free plagiarism detection tool* [online]. 2006 [cit. 2007-03-10]. Dostupný z WWW: <<http://www.plagiarism.tk>>.

- [Stiffler2003] STIFFLER, Jason, et al.: Examining Anti-Plagiarism Software: Choosing the Right Tool. In *Presented at EDUCAUSE Annual Conferences*. 2003. Dostupný z WWW: <<http://www.educause.edu/ir/library/pdf/EDU03168.pdf>>.
- [sherlock] *The BOSS Online Submission System* [online]. [2004] [cit. 2006-07-06]. Dostupný z WWW: <<http://sourceforge.net/projects/cobalt/>>.
- [UKansas2007] University of Kansas: *About the University* [online]. c2007 [cit. 2007-03-10]. Dostupný z WWW: <<http://www.ku.edu/about/facts.shtml>>.
- [UniSydneyReport2003] University of Sydney academic board: *Plagiarism detection software, its use by universities, and student attitudes to cheating: a report for the University of Sydney Teaching and Learning Committee*. c2003. Dostupný z WWW: <[http://www.usyd.edu.au/su/ab/docs/2003/ABAgaug03\\_attach\\_13.2.3.pdf](http://www.usyd.edu.au/su/ab/docs/2003/ABAgaug03_attach_13.2.3.pdf)>.
- [UJC] Ústav pro jazyk český Akademie věd ČR: *Jazyková poradna : Na co se nás často ptáte* [online]. [cit. 2006-07-06]. Dostupný z WWW: <<http://www.ujc.cas.cz/poradna/pořadna.htm>>.
- [Vamplew2005] VAMPLEW, Peter, DERMOUDY, Julian: An anti-plagiarism editor for software development courses. In *Proceedings of the 7th Australasian conference on Computing education - Volume 42*. Newcastle, New South Wales, Australia, 2005. s. 83. Dostupný z WWW: <[http://portal.acm.org/proceedings/ft\\_gateway.cfm?id=1082435&type=pdf&coll=portal&dl=ACM&CFID=1452886&CFTOKEN=49967946](http://portal.acm.org/proceedings/ft_gateway.cfm?id=1082435&type=pdf&coll=portal&dl=ACM&CFID=1452886&CFTOKEN=49967946)>. ISBN 1445-1336. ISSN 1-920682-24-4.
- [catchitfirst] Vancouver Software Labs: *CatchItFirst.com* [online]. c2006 [cit. 2007-03-10]. Dostupný z WWW: <<http://www.catchitfirst.com/>>.
- [White2004] WHITE, Daniel R., JOY, Mike S.: Sentence-based natural language plagiarism detection. Journal on *Educational Resources in Computing (JERIC)*. 2004, vol. 4, is. 4, Article No. 2. Dostupný z WWW: <[http://portal.acm.org/ft\\_gateway.cfm?id=1086341&type=pdf&coll=GUIDE&dl=GUIDE&CFID=20831717&CFTOKEN=80645880](http://portal.acm.org/ft_gateway.cfm?id=1086341&type=pdf&coll=GUIDE&dl=GUIDE&CFID=20831717&CFTOKEN=80645880)>. ISSN:1531-4278.
- [LCSwiki2006] Wikipedia.org: *Longest common subsequence* [online]. c2006 [cit. 2006-07-06]. Dostupný z WWW: <[http://en.wikipedia.org/wiki/Longest\\_common\\_subsequence](http://en.wikipedia.org/wiki/Longest_common_subsequence)>.
- [wikiplag] Wikipedia.org: *Plagiarism* [online]. c2006 [cit. 2006-07-06]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Plagiarism>>.
- [Wise1996] WISE, Michael J.: YAP3: improved detection of similarities in computer program and other texts. In *Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education*. Philadelphia, Pennsylvania, United States, 1996. s. 130. ISSN 0097-8418.
- [Young2002] YOUNG, Jeffrey R.: Anti-Plagiarism Experts Raise Questions About Services With Links to Sites Selling Papers. *The Chronicle of Higher Education*. 2003, no. 03, Dostupný z WWW: <<http://chronicle.com/free/2002/03/2002031201t.htm>>.
- [Zeidman2004] ZEIDMAN, Bob: Detecting Source-Code Plagiarism: Tools and algorithms for finding plagiarism in source code. *Dr. Dobb's Journal* [online]. 2004, vol. VI, no. 01 [cit. 2006-07-06]. Dostupný z WWW: <<http://www.ddj.com/184405734>>.
- [Zelený2006] ZELENÝ, Jindřich. *IT\_380 - Seminární práce* [online]. [1999-2006] [cit. 2006-07-06]. Dostupný z WWW: <<http://nb.vse.cz/~zelenyj/it380/eseje/vse.htm>>.

## Terminologický slovník

### Přejaté termíny ze slovníku ČSSI

Algoritmus	Algoritmus obecně je každý přesný předpis jednoznačně určující postup řešení úlohy pomocí definované soustavy operací. Algoritmus pro počítačové zpracování musí po konečném počtu kroků dospět k požadovaným výsledkům (rezultativnost) a musí dojít ke správným výsledkům pro všechny vstupní hodnoty z množiny definované v zadání úlohy (hromadnost). Pro danou úlohu lze sestavit zpravidla více algoritmů, na základě časové a paměťové náročnosti můžeme obvykle vybrat nevhodnější. Algoritmus je možné vyjádřit slovním popisem, grafickými prostředky (například strukturní diagramy, vývojové diagramy) nebo programovacím jazykem.
Aplikace	Využívaný software uživatelem, jím zpracovávaná data, poskytované funkce a podporované procesy a současně s aplikací spojené potřebné technologie.
Autentizace	Autentizace znamená ověřování proklamované identity subjektu. Autentizace znamená ověřování pravosti, autentický znamená původní, pravý, hodnověrný. Autentizace patří k bezpečnostním opatřením a zajišťuje ochranu před falšováním identity (angl. Impersonation, maskarade), kdy se subjekt vydává za někoho, kým není. Rozlišujeme autentizaci entity (osoby, programu) a autentizaci zprávy.
Databáze	zde ve smyslu datová základna – integrovaná počítačově zpracovávaná množina dat
Doba odezvy	Doba odezvy informačního systému představuje čas, který uplyne mezi okamžikem zadání požadavku na zpracování a okamžikem přijetí výsledku zpracování zadavatelem.
Dotazovací jazyk	Umožňuje uživateli neprogramátorovi formulovat dotazy či získávat informace z databáze formulací požadavku v běžném jazyce např. angličtině.
Entita	Označení obecného jasně definovaného prvku, příklady entit: třída, objekt, proces.
HTML	Jazyk, který vychází z normy ISO8879 (Standard Generalized Markup Language). HTML vznikl v souvislosti s rozvojem služby WWW. Je založen na principu označování (mark-up) částí textu pomocí předem známé množiny značek. Značky specifikují význam textu (např. určitá část textu je nadpisem) nebo umožňují vkládat do textu odkazy na jiné objekty.
Informatika	Multidisciplinární obor, jehož předmětem je vývoj a užití informačních systémů v organizacích a společenstvích, a to na bázi informačních a komunikačních technologií. Multidisciplinarita v tomto případě znamená,

	že zkoumání předmětu zahrnuje technické, ekonomické, sociální, psychologické, právní a další aspekty.
Internet	Globální celosvětová počítačová síť propojující regionální a rozsáhlé počítačové sítě, které používají TCP/IP jako síťový protokol.
Internetová aplikace	Druh aplikace, jejíž uživatelské rozhraní je zobrazováno prohlížečem (též webová aplikace).
Java	Objektově orientovaný programovací jazyk.
Klient/server	Architektura softwarových systémů, kde jeden program (proces) vystupuje v roli klienta a druhý program (proces) v roli serveru. Úkolem klienta je umožnit zadání požadavku a následné zobrazení výsledků. Úkolem serveru je požadavky klienta přijímat, provést zpracování a formulovat odpověď, kterou zašle klientovi. Jde o speciální případ vrstvené architektury (prezentační vrstva, aplikační vrstva, datová vrstva). Podle rozdílení těchto vrstev do programu klienta a serveru rozlišujeme dvouúrovňovou, tříúrovňovou a n-úrovňovou architekturu klient/server.
Komprese dat	Komprese dat je proces, zajišťující snížení nároků souborů dat na paměťový prostor, nutný k jejich uložení. Komprese spočívá v redukci počtu bitů, potřebných pro digitální vyjádření "předmětu komprese" (např. souboru).
Model	Dílčí pohled na vytvářený systém, souhrn všech pohledů na celý systém.
Modul	Subsystém, část modelu na dané úrovni podrobnosti již dále nerozkládaná, s relativně samostatnou funkčností.
Počítačová síť	Představuje obecně systém vzájemně propojených počítačů, terminálů a periferních zařízení, komunikujících prostřednictvím komunikačního subsystému sítě, přenosových médií a aktivních komunikačních prvků.
Program	1) Předpis posloupnosti činností pro podřízené či výkonné elementy (program zájezdu, televizní program, program rozvoje firmy). 2) Program ve výpočetní technice – předpis činnosti počítače. Velké programy = programové systémy, např. operační systém, databázový systém.
Programovací jazyk	Souhrn pravidel, která popisují způsob zápisu příkazů pro počítač. Programovací jazyk může být blízký počítači nebo blízký člověku, pak ale musí existovat převodník mezi souborem zapsaným ve vyšším programovacím jazyce.
Software	Programy, procedury a pravidla pro zpracování konkrétní úlohy na počítači neboli pokyny počítači, jak má danou úlohu řešit. Program je napsán v programovacím jazyku (např. Java, C, Pascal, Cobol, assembler). Software se dělí na základní (operační systém, databázový systém, komunikační systém) a aplikační.
UTF8	Pravděpodobně nejrozšířenější způsob zápisu/přepisu/přenosu znaků v Unicode pomocí "běžných" znaků, standard v rámci Internetu.

WWW prohlížeč Je aplikační program služby WWW. Uživateli zpřístupňuje a zobrazuje informace (textové, grafické, multimediální), které jsou uspořádány do stránek (dokumentů) a umožňuje mu, aby mezi stránkami (nebo v jejich rámci) přecházel prostřednictvím poklepání na odkaz.

## ***Nové a nově vymezené termíny***

DOC	Formát souborů používaný produkty Microsoft Word.
Extrakorpální nástroj	Typ → nástroje pro detekci plagiátů, který umožňuje odhalení → plagiátu jehož zdroj se nachází mimo korpus.
Intrakorpální nástroj	Typ → nástroje pro detekci plagiátů, který umožňuje odhalení → plagiátu pouze pokud se jeho zdroj vyskytuje v tomtéž → korpusu.
Intrinsic nástroj	Typ → nástroje pro detekci plagiátů, který pracuje pouze na úrovni jediného dokumentu a neprovádí žádná porovnání s jinými dokumenty.
Korpus	Množina dokumentů poskytovaná → nástroji pro detekci plagiátů najednou jako celek ke zpracování (např. množina letošních odevzdaných seminárních prací z konkrétního předmětu).
Metadata	Strukturovaná data o datech. Data popisující obsah či význam jiných dat.
Nástroj pro detekci pl.	Automatizovaný nejčastěji softwarový nástroj podporující rozlišení → plagiátů od děl původních a originálních. Pracují na různých principech, nejčastěji zřejmě porovnávání částí obsahu mnoha dokumentů.
PDF	Portable Document Format, formát souborů vytvořený v roce 1993 společností Adobe Systems pro přesnou reprezentaci vzhledu dokumentů.
Plagiát	Dílo nebo jiný výtvar vzniklý procesem → plagiátorství na základě jiného díla (zdroj plagiátu).
Plagiátorství	Vydávání cizího díla nebo jeho části za dílo vlastní; tvorba vlastního díla založená z podstatné části na díle cizím aniž je původní autor uveden.
RTF	Rich Text Format, formát souborů vytvořený v roce 1987 společností Microsoft pro ukládání jednoduše formátovaných dokumentů.
Smíšený nástroj	Typ → nástroje pro detekci plagiátů, který kombinuje funkcionality → intrakorpálního a → extrakorpálního nástroje.
Zákl. jednotka porovnání	Nejmenší část obsahu dokumentu (např. slovo, tři slova, věta) používaná k samostatnému porovnání s obsahem jiných dokumentů v nástrojích pro detekci plagiátů pracujících na principu porovnání. Může být složena z menších částí → základních prvků.
Základní prvek	Nejmenší, z hlediska zpracování dále nedělitelná část obsahu dokumentu používaná v nástrojích pro detekci plagiátů pracujících na principu porovnání. Neporovnává se samostatně ale ve skupině s ostatními tvoří → základní jednotku porovnání.