VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

Fakulta informatiky a statistiky

Metody zpracování informací

II - Ukládání a vyhledávání

RNDr. Jan Rauch, CSc.
1995

Toto je elektronická verse skript určená pro studenty předmětu 4IZ210 daná k dispozici dne 28. 10. 2011. Od původní verse se liší opravou známých tiskových chyb. Chybí zde obrázky 10.1, 12.1 a 12.2, další informace jsou přímo u těchto obrázků.

Obsah

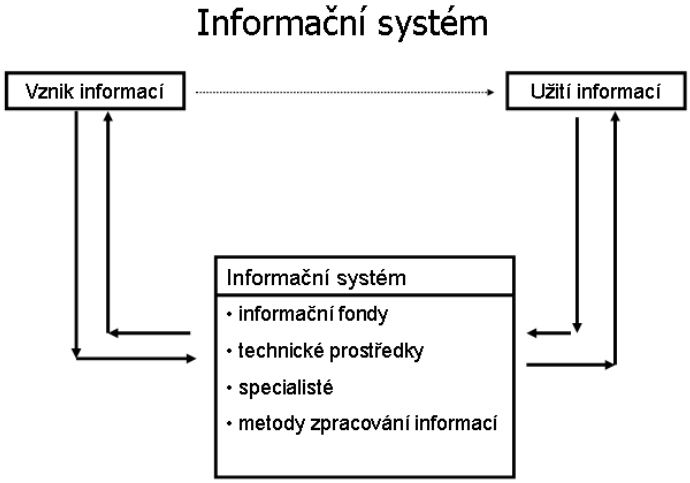
Kapitola 1

Úvod

Potřeba přenosu informací v prostoru a čase sahá k počátkům lidské civilizace, stejně jako vývoj odpovídajících prostředků a metod. Zásadními impulsy v tomto směru byly zejména počátek používání písma, vynález knihtisku, průmyslová revoluce a v posledních desetiletích stále se zrychlující vývoj výpočetní a komunikační techniky. Důsledkem potřeby přenosu informací jsou permanentně probíhající informační procesy.

Informačním procesem rozumíme přenos informace od jejího vzniku k uživateli. Předmětem zájmu jsou zejména informační procesy ve společnosti a v organizaci, ale i individuální informační proces. Na cestě informace od vzniku k užití je mnoho různých překážek, k nejzávažnějším patří prostor, čas, struktura informací a odbornost uživatelů.

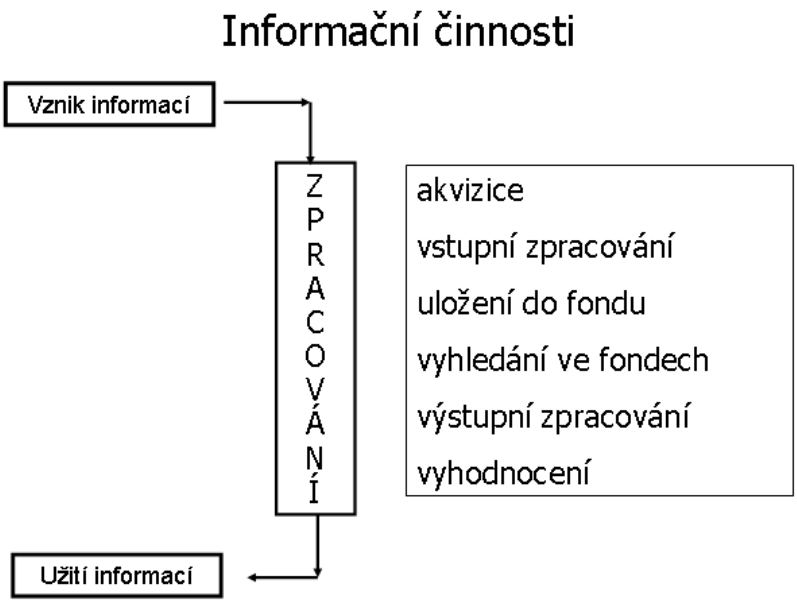
Pro přenos informací slouží informační systémy, přenos přímou komunikací mezi tvůrcem a uživatelem informace probíhá spíše vyjímečně, viz obr. 1.1.



Obrázek 1.1: Informační proces

Informace jsou uchovávány v informačních fondech. Přenos informací od vzniku k užití je zprostředkováván informačními činnostmi, viz obr. 1.2.

Podrobněji je o informačním procesu a o souvisejících pojmech pojednáno v [RAUCH 94]. Zde se budeme zabývat metodami ukládání a vyhledávání informací používanými při vstupním zpracování, ukládání a vyhledávání informací v informačních fondech. Metody používané při vstupním zpracování zařazujeme k metodám ukládání a vyhledávání proto, že způsob uložení a tedy i způsob vyhledávání podstatně závisí na výsledku vstupního zpracování.



Obrázek 1.2: Informační činnosti

Způsob ukládání a vyhledávání souvisí s vlastnostmi ukládaných a vyhledávaných informaci. Jiná je situace v případě, kdy ukládané informace jsou ve formě vhodných kódů (např. numerických nebo alfanumerických) a jiná v případě ukládání a vyhledávání informací ve formě volného textu, případně obrazu nebo multimediální informace.

V prvním případě je úloha snažší, neboť lze původní, netransformované informace pomocí počítačů (případně i mechanicky) třídit a separovat. Typickým příkladem jsou informace uchovávané v běžných databázových systémech a poskytované prostřednictvím SQL. Jednotkou informace je jeden rekord, rekordy lze třídit, atd. V tomto případě lze i přesně vyjádřit informační požadavek. Příkladem je úloha vyhledat všechny zaměstnance expedičního oddělení, jejichž měsíční plat je vyšší než 12 000 Kč.

S výběrem vhodných kódů i s jejich používáním je spojeno mnoho problémů. Příkladem je situace s oborovými a výrobkovými kódy popisovaná v [BABKA 94]. Podle SIC (Standard Industrial Classification), používané mimo jiné v databázích Dun & Bradstreet, jsou kuličková ložiska označena kódem 3562, v InformKatalogu kód 3562 znamená všechna ložiska, podle Registru ČSÚ patří kuličková ložiska do položky číslo 28 750 - výroba ostatního kovového zboží, ještě jinak je tomu v německých bázích dat, atd. Problémy kódování se zde nebudeme zabývat. Ukládání a vyhledávání informace vyjádřené pomocí kódů se však týkají teoretické partie skript v kapitole 13.

K situacím, kdy je ukládáná a vyhledáváná informace ve formě volného textu patří např. práce s knihami a časopisy v knihovnách, vyhledávání odborných článků v textových databázích nebo práce se zákony a soudními rozhodnutími ať už v papírové nebo elektronické formě. Knihy, časopisy, články, diplomové práce, disertace, zákony, rozsudky atd. budeme souhrnně nazývat **textovými dokumenty**. Lze hovořit i o obrazových, zvukových, případně multimediálních dokumentech. Předmětem našeho zájmu budou pouze textové dokumenty. Místo termínu "textový dokument" budeme proto většinou používat pouze termín "dokument".

Požadavek na vyhledání dokumentů může být vyjádřen pomocí formálních údajů. Příkladem je úloha vyhledat všechny články daného autora vydané v daném období. V podstatě se potom jedná o výše zmíněnou práci s kódovanou informací. Nový okruh problémů začíná, pokud vyhledáváme dokumenty podle jejich obsahu. Dosud vyvinuté vyhledávací systémy neumožňují tuto úlohu řešit zcela uspokojivě. Problémy vyhledávání dokumentů jsou v obecné rovině diskutovány v kapitole 2.

K ukládání a vyhledávání textových dokumentů lze přistoupit dvěma způsoby. První spočívá v tom, že při vstupním zpracování vytvoříme **popis dokumentu** a dále pracujeme pouze s tímto popisem. Jedná se o přístup realizovatelný jak zcela bez počítačů, tak i s různým stupněm jejich využití. V této sovislosti se hovoří o **sekundárních** a **primárních dokumentech** nebo **pramenech**, a o **sekundárních** a **primárních fondech**. Je jim věnována kapitola 3.

Přesto, že výpočetní a komunikační technika hrají stále dominantnější roli v informačním procesu, má smysl se zabývat metodami a prostředky pro ukládání a vyhledávání dokumentů vyvinutými v době před jejím nástupem. Důvodem je mimo jiné fakt, že řada informačních fondů je stále dostupná pouze prostřednictvím klasických metod. Dalším důvodem je, že při automatizaci informačních fondů pomocí výpočetní techniky jsou klasické metody a prostředky podstatným způsobem využívány. Často jsou však dále doplňovány a rozvíjeny způsobem bez výpočetní techniky nerealizovatelným.

Při popisu dokumentu je nejobtížnější dostupnými prostředky **charakterizovat obsah**. V kapitole 4 se budeme zabývat prostředky pro tyto účely vyvinutými před zaváděním výpočetní techniky. Kapitola 5 ukazuje několik přístupů, jak lze k těmto účelům využít výpočetní techniku.

První softwarové systémy pro vyhledávání textových dokumentů vznikaly aplikací počítačů v práci s bibliografickými záznamy (t.j. sekundárními dokumenty) připravenými podle dříve vyvinutých metodik. Principem bylo vyhledávání dokumentů splňující danou podmínku vyjádřenou pomocí booleovských logických spojek. Hovoříme proto o **booleovském modelu**, kterému je věnována kapitola 6.

Počítače umožnily, aby dílčí podmínky kombinované pomocí logických spojek postihovaly i různé gramatické tvary slov a vzájemnou pozici slov v textu (např. v názvu dokumentu). Kombinace klasických metod s novými možnostmi počítačů činí z dnešních softwarových systémů pro vyhledávání textových dokumentů založených na booleovském modelu silné nástroje. Pro jejich efektivní využívání je nutno znát i principy nových možností, které přinesla výpočetní technika. Nejdůležitější z nich jsou také popsány v kapitole 6.

Druhý z přístupů k práci s textovými dokumenty spočívá v tom, že uživatel má stále k dispozici plný text dokumentu v elektronické formě. Při vyhledávání se obvykle aplikují prostředky používané i v rámci booleovského modelu, dostupnost plného textu však vyhledávání relevantních dokumentů činí efektivnějším. Systémům pro práci s plnými texty je věnována kapitola 7.

Přes své nesporné úspěchy a značné rozšíření má boolevský model podstatné nevýhody. Již delší dobu jsou proto hledány cesty, jak nahradit nebo rozšířit booleovskou logiku tak aby tyto nevýhody byly odstraněny. Některé základní principy v tomto směru jsou ukázány v kapitole 8.

Krom booleovského modelu existují i další přístupy k automatizaci vyhledávání textových dokumentů založené na poměrně jednoduchých matematických metodách a pojmech. Dva z nich jsou naznačeny v kapitole 9.

Relativně novým, ale velmi rychle do praxe se rozšiřujícím přístupem k vyhledávání textových dokumentů, je pojmové vyhledávání, kterému je věnována kapitola 10.

Probíhá však také rozsáhlý výzkum a vývoj nových systémů pro ukládání a vyhledávání dokumentů, tyto systémy využívají jak sílu současné výpočetní techniky tak i řadu hlubokých teoretických výsledků včetně principů umělé inteligence. Základní informace o umělé inteligenci je v kapitole 11. Příklad použití prostředků umělé inteligence při vyhledávání textových dokumentů je v kapitole 12.

V kapitole 13 jsou naznačeny některé teoretické otázky související s algoritmy a datovými strukturami pro ukládání a vyhledávání kódovaných informací. Kapitola 14 se zabývá invertovaným souborem, který je specifickým nástrojem pro vyhledávání textových dokumentů.

Kapitoly 13 a 14 je však třeba chápat pouze jako stručný úvod do rozsáhlé problematiky. Jak již bylo naznačeno v předmluvě, budou jí věnována další skripta.

Do oblasti ukládání a vyhledávání informací patří i hypertextové systémy. Nejsou zde však zpracovány vzhledem k tomu, že na toto tema se připravují samostatná skripta.

Kapitola 2

Vyhledávání dokumentů

2.1 Základní pojmy

Cílem této kapitoly je stručně informovat o základních pojmech a faktech souvisejících s vyhledáváním dokumentů. Na začátku celého procesu je **informační potřeba**. Tento pojem budeme pokládat za intuitivně dostatečně jasný a nebudeme jej dále definovat. Příkladem je potřeba informací k výzkumné, vývojové, výrobní nebo obchodní činnosti, potřeba informací k vypracování semestrální nebo diplomové práce, atd. Podrobněji je pojem informační potřeby diskutován např. v [VICKERY].

Informační potřeba se vyjadřuje pomocí informačního požadavku, například:

- články týkající se privatizace cukrovarů uveřejněné v denním tisku,
- odborné články zabývající se přepisovatelnými optickými disky uveřejněné v odborných časopisech,
- učebnice predikátového počtu.

Prvním krokem při uspokojování informačního požadavku je vytipování vhodných informačních zdrojů. Ve druhém kroku jsou vyhledávány dokumenty v jednotlivých zdrojích. Pro vyhledávání dokumentů se používá pojem **rešerše**. Hovoří se o například o retrospektivní rešerši nebo o přírůstkové rešerši, viz např. [ČERNÁ 92], stručně i [RAUCH 94]. Rešerší se označuje jak vlastní proces vyhledávání, tak i jeho výsledek. Používá se i pojem **rešeršní strategie**, několik zásad rešeršní strategie je zmíněno v kapitole 12 v souvislosti s aplikacemi umělé inteligence. Vyhledávání dokumentů v konkrétním informačním fondu (neboli rešerši do daného fondu) je z obecného pohledu věnován odstavec 2.2.

Přirozeným cílem je získat vhodný počet dokumentů vyhovujících informačnímu požadavku. Co je vhodný počet záleží na konkrétní situaci. Pokud chceme investovat v oblasti cukrovarů, pak je vhodné získat všechny články týkající se privatizace cukrovarů. Pro přípravu na zkoušku z matematické logiky však pravděpodobně nebude mít smysl opatřit si všechny učebnice predikátového počtu.

V této souvislosti se zavádí **osobní** (**individuální**) **relevance dokumentu**. Pokud nebude nebezpečí nedorozumění, budeme hovořit pouze o relevanci dokumentu, viz též [Salton nebo Vickery]. Dokument je relevantní, jestliže vyhovuje informačnímu požadavku.

Podstatným faktem je, že ve většině případů prakticky nelze získat všechny relevantní dokumenty a naopak, že ne všechny získané dokumenty budou vždy relevantní. Jednou z příčin

je, že se musíme omezit jenom na konečný, většinou malý počet informačních zdrojů, přičemž nelze spoléhat na to, že ve vybraných zdrojích jsou dostupné všechny relevantní dokumenty. Druhou podstatnou příčinou je, že vyhledávácí prostředky konkrétního zdroje většinou nejen že neumožní vyhledat všechny relevantní dokumenty ve zdroji uložené, ale naopak vydají i dokumenty irelevantní. S tím souvisí i pojmy **přesnost** a **úplnost vyhledávání**, kterým je věnován odstavec 2.3.

V případě informačního požadavku na odborné články zabývající se optickými přepisovatelnými disky uveřejněné v odborných časopisech, je vhodné použít např. CD-ROM Computer Select, dostupný např. ve Státní technické knihovně. Dále je možno použít anotace z časopisů PC WORLD a COMPUTER WORLD vystavené v počítačové síti VŠE. Je také možno pátrat v časopisech dostupných v různých knihovnách, např. v Knihovně akademie věd ČR nebo v knihovnách různých odborných ústavů, případně ve firemních knihovnách. Žádný ze zmíněných zdrojů jistě nebude obsahovat všechny časopisy zabývající se danou tematikou.

To že v daném informačním zdroji nejsou dostupné všechny relevantní dokumenty k danému informačnímu požadavku nemusí nutně znamenat nedostatek informačního zdroje. CD-ROM Computer Select se zabývá pouze časopisy vycházejícími v angličtině, anotace z časopisů PC WORLD a COMPUTER WORLD zase naopak pouze českým odborným tiskem. V knihovně firmy zabývající se výrobou optických disků lze očekávat rozsáhlý fond s dokumenty týkajícími se technologie výroby, méně však aplikací. Naopak, firma zabývající se projektováním a budováním podnikových informačních systémů bude uchovávat spíše dokumenty zabývající se aplikacemi optických disků než detaily technologie jejich výroby.

Pro kvalitu informačního zdroje je podstatné zejména co nejpřesnější vymezení relevantních informací pro zdroj a průběžné získávání co nejvíce relevantních a co nejméně irelevantních informací.

Obecně lze říci, že při určování informací relevantních pro daný zdroj nejsou brány v úvahu informační potřeby pouze jednoho uživatele, ale předpokládaný souhrn informačních potřeb všech potencionálních uživatelů. Vymezení a získávání relevantních informací závisí i na typu zdroje, jiné je pro knihovnu a jiné pro databázi ekonomických ukazatelů. Z relevantních informací se ve většině případů nepodaří získat všechny. Naopak, z různých důvodů se mohou do zdroje dostat i informace ve skutečnosti pro zdroj irelevantní. Jednou z možností je dar zařazený do knihovního fondu.

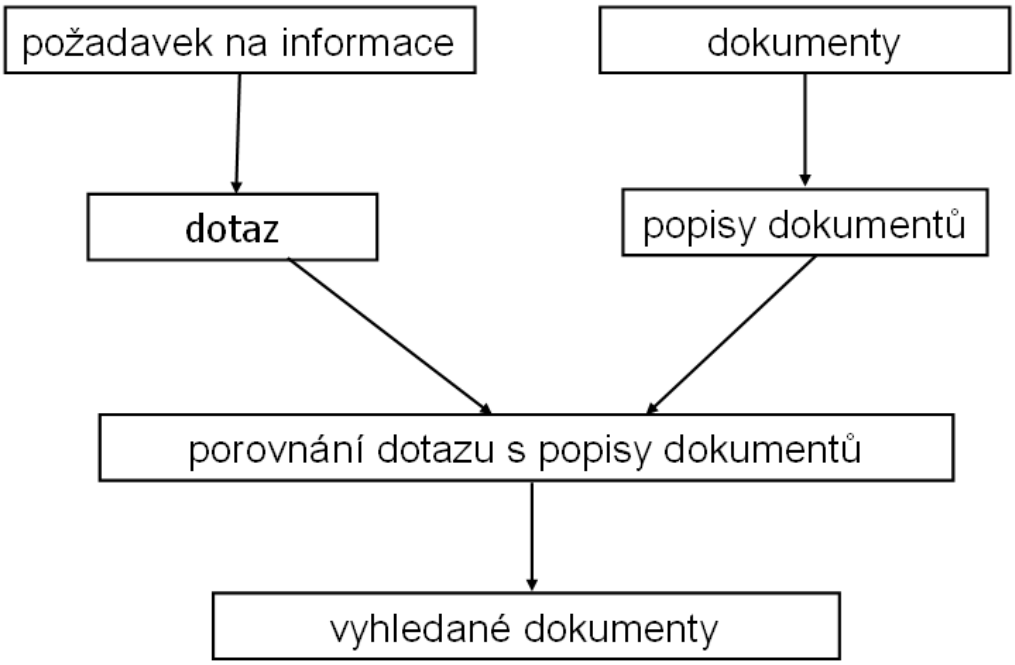
Výsledné chování zdroje i jeho uživatele je ovlivněno řadou faktorů, například cenou informací, časovými omezeními, některé informace nelze získat bez porušení zákona, atd. Těmito problémy se zde však nebudeme zabývat.

2.2 Rešerše do daného fondu

Dále se budeme věnovat vyhledávání textových dokumentů v konkrétním informačním fondu. Celou situaci lze schematicky znázornit dle obr. 2.1.

Každý dokument je v rámci vstupního zpracování před uložením do informačního fondu popsán prostředky danými tímto fondem, výsledkem je **popis dokumentu**. Pro tuto činnost se používá i termín "indexování dokumentů". Popis dokumentu je, v závislosti na konkrétním informačním fondu, tvořen několika položkami. Obvykle mezi ně patří název dokumentu, autor nebo autoři, datum vydání, nakladatelství, charakteristika obsahu dokumentu, atd. V případě informačního fondu uchovávajícího plné texty dokumentů je do popisu dokumentu třeba zahrnout i každé slovo z textu dokumentu, viz též kapitolu 7.

Informační požadavek se vyjádří pomocí dotazovacího jazyka, výsledkem je **dotaz**. Dotaz určuje nějakou podmnožinu dokumentů uchovávaných v informačním fondu. Různé dotazovací jazyky se často velmi liší jak vyjadřovacími možnostmi, tak i syntaxí. Vyhledávání dokumentů



Obrázek 2.1: Ukládání a vyhledávání dokumentů

je možno chápat jako porovnávání popisů dokumentů s dotazem. Pokud popis dokumentu vyhovuje dotazu, je dokument považován za vybraný a je zařazen do výstupu. Vybrané dokumenty se nazývají **hity**. Z důvodů dále rozvedených je třeba velmi pečlivě rozlišovat mezi informačním požadavkem a dotazem.

Schema vyhledávání uvedené v obr 2.1 budeme diskutovat vzhledem k vyhledávacím systémům využívajícím výpočetní techniku, které nás především zajímají. I když lze toto schema vztahnout i k procesu vyhledávání pomocí klasických knihovních katalogů, nebudeme se v dalším na případné i když pravděpodobně nevelké problémy z takovýchto zobecnění ohlížet.

Popis dokumentu má obvykle dvě části. První část, nazývaná identifikační nebo jmenný popis, obsahuje formální údaje, např. jméno autora, rok vydání nebo nakladatele. Druhá část obsahuje nějakou charakteristiku obsahu dokumentu. Jak už bylo naznačeno v úvodu, vyhledávání dokumentů podle formálních údajů nepřináší principiálně nové problémy. Pokud požadujeme například všechny články daného autora vydané v daném roce a uchovávané v daném informačním fondu, není problém provést vyhledání tak, abychom získali všechny dokumenty vyhovující takovému požadavku a již žádné jiné. Údaje potřebné pro vyhledání mohou být beze zbytku vyjádřeny jak v popisu dokumentu, tak i v dotazu. Pro každý dokument lze jednoznačně určit, zda danému informačnímu požadavku vyhovuje nebo ne.

2.3 Přesnost a úplnost vyhledávání

Jiná je situace, pokud vyhledáváme dokumenty podle obsahu. Z různých příčin, zejména díky tomu, že dostupné prostředky neumožňují jednoznačně a beze zbytku vyjádřit obsah dokumentu, dochází při popisu dokumentu i při formulaci dotazu ke značnému zjednodušení a často i ke zkreslení obsahu. V důsledku toho se jednak do výstupu dostávají dokumenty, které

nevyhovují informačnímu požadavku a jednak se do výstupu nedostanou všechny dokumenty informačnímu požadavku vyhovující. Pokud použijeme pojem relevantní dokument zavedený v odstavci 2.1, lze říci, že při vyhledávání dokumentů podle obsahu dochází k tomu že některé relevantní dokumenty nejsou vybrány a naopak že jsou vybrány irelevantní dokumenty.

Na základě rozboru konkrétních případů vyhledávání se ukázalo, že k uvedeným odchylkám dochází v nezanedbatelném množství. Tento problém byl studován zejména u booleovského modelu vyhledávání dokumentů, některé podrobnosti jsou uvedeny v kapitole 6.

Pro určení míry úspěšnosti vyhledávání textových dokumentů se používají různé koeficienty definované na základě počtů dokumentů dle tabulky 2.1.

	vybráno	nevybráno	
relevantní	A	B	$A+B$
irelevantní	C	D	$C+D$
	$A+C$	$B+D$	

Tabulka 2.1: Počty dokumentů pro hodnocení úspěšnosti vyhledávání

Počet vybraných dokumentů relevantních k informačnímu požadavku je označen A. Písmeno B značí počet dokumentů sice relevantních k informačnímu požadavku, ale při vyhledávání nevybraných. Počet dokumentů irelevantních k informačnímu požadavku ale přesto vybraných je označen C a počet irelevantních nevybraných dokumentů je označen D.

Používají se především dva ukazatele, **koeficient přesnosti** (**Precision**) a **koeficient úplnosti** (**Recall**).

Koeficient přesnosti, obvykle značený P, říká jaká je šance, že vybraný dokument je relevantní. Je definován takto:

$$P = \frac{\text{počet vybraných relevantních}}{\text{počet všech vybraných}} = \frac{A}{A+C}$$

Koeficient úplnosti, obvykle značený R, udává jaká je šance, že relevantní dokument bude vybrán. Je definován takto:

$$R = \frac{\text{počet vybraných relevantních}}{\text{počet všech relevantních}} = \frac{A}{A+B}$$

Ideální požadavek je, aby $R = 1$ i $P = 1$. Obvykle je však mezi přesností a úplností vztah nepřímé úměrnosti, větší úplnost lze získat jen za cenu menší přesnosti a naopak.

Uvedené koeficienty přesnosti a úplnosti jsou založeny na ostré hranici mezi relevantním a irelevantním dokumentem. Takovýto předpoklad však není vždy reálný. Předpokládejme, že máme vypracovat studii o ekologické problematice dané oblasti. Odborný článek podrobně se zabývající současným stavem a prognózami ekologických aktivit v deseti největších podnicích dané oblasti bude téměř jistě víc relevantní než článek poukazující, že úspěšně proběhla rekultivace odlehlé skládky jen málo toxických odpadů umístěná na okraji hodnocené oblasti. Lze však očekávat, že oba články budou na základě vhodně formulovaného dotazu vybrány jako relevantní.

I z tohoto důvodu je jedním ze současných trendů snaha číselně vyjádřit míru relevance dokumentu k požadavku. Míru relevance dokumentu k požadavku vyjádřenému jedním klíčovým slovem lze například zvýšit, pokud se slovo vyskytuje v nadpisu. Jiným východiskem k výpočtu stupně relevance může být frekvence jednotlivých slov v plném textu dokumentu. Více k těmto přístupům je uvedeno v kapitolách 8 až 10.

V této souvislosti je vhodné zmínit tři složky informace popsané např. v [RAUCH 94]. Jedná se o

- sdělení (message),
- data (elements),
- nosič (medium).

Sdělením se rozumí obsah informace pro příjemce. **Data** slouží k vyjádření informace v komunikačním jazyce, mohou to být čísla, diagramy, slova, atd. **Nosič** je fyzický prostředek uložení a přenosu informace.

Současné komerční rešeršní systémy umožňující práci s plnými texty dokumentů využívají elektronických nosičů informace. Ve smyslu výše uvedených tří složek informace však tyto systémy pracují pouze s daty, ne přímo se sdělením. Používané prostředky, např. frekvence slov, nemohou plně vyjádřit obsah informace pro příjemce. Pokud se podaří uspokojivým způsobem vyřešit automatizaci práce s obsahem informace v textových dokumentech, bude možno automatizovat řešení například i následujících úloh:

- vyhledání všech dokumentů z fondu, které se týkají daného problému vyjádřeného volným textem,
- vyhledání všech dokumentů z fondu, ve kterých se daný problém posuzuje ze zadaného hlediska,
- vyhledání všech dokumentů z fondu, ve kterých se daný problém řeší stejně, jako v zadaném dokumentu,
- nalezení řešení daného problému na základě studia dokumentů v daném fondu.

Kapitola 3

Popis dokumentů

3.1 Primární a sekundární prameny

Předpokladem uchování a manipulace s dokumentem je jeho popis obsahující všechny potřebné údaje. Jaké údaje jsou zapotřebí je dáno způsobem práce s dokumenty. Nejstarší zkušenosti s manipulací s dokumenty souvisí se vznikem měst [VICKERY 94]. Práce s textovými dokumenty se v souvislosti s rozvojem a růstem průmyslové výroby, výzkumu, vývoje vzdělávání i dalších aktivit stala součástí každodenního života. V dnešní době dochází v této oblasti k rychlému vývoji vzhledem k novým možnostem manipulace s dokumenty v elektronické formě.

Zde si budeme všímat popisu textových dokumentů který vychází zejména z potřeb práce s knihovními fondy a z potřeb šíření textových informací prostřednictvím časopisů, sborníků, diplomových a disertačních prací, výzkumných zpráv, studií atd. Předpokládáme šíření dokumentů jak v klasické, tak v elektronické formě. Dále popsané metody a prostředky jsou však v různé míře používány i jinde, např. v archivech nebo i při práci s dokumenty ve firmě.

Činnosti, které je třeba hlavně zajišťovat jsou:

- vyhledávání dokumentů podle obsahu,
- vyhledávání dokumentů podle formálních znaků,
- informování uživatelů o nových dokumentech, zejména o jejich obsahu,
- výpůjčky dokumentů z vlastního knihovního fondu,
- výpůjčky dokumentů z fondů jiných knihoven.

Podstatným rysem manipulace s dokumenty v této oblasti je nutnost práce se zhuštěně vyjádřeným obsahem dokumentu. Proto je součástí popisu dokumentu i charakteristika obsahu.

Místo termínu "dokument" se často používá termín "**informační pramen**" [RAUCH 94]. Popis informačního pramene se nazývá **sekundární informační pramen**, původní dokument se nazývá **primární informační pramen**. Primární informační prameny se uchovávají v **primárních informačních fondech**, sekundární informační prameny v **sekundárních informačních fondech**.

Typickým příkladem primárního informačního fondu je knihovní fond v knihovně, jednotlivé knihy jsou primárními informačními prameny. Příkladem sekundárního informačního fondu je autorský katalog, jednotlivé katalogové lístky jsou sekundární informační prameny. Smyslem

sekundárních informačních pramenů je vhodným způsobem zastupovat primární prameny při manipulaci s nimi.

Sekundární informační prameny jsou zpravidla nějakým způsobem odvozeny od bibliografického záznamu, kterému je věnován následující odstavec. Pro popis dokumentů se používají selekční jazyky, které jsou zmíněny v odstavci 3.3.

3.2 Bibliografický záznam

Bibliografický záznam obsahuje údaje nutné pro to, aby mohl potřebným způsobem zastupovat primární dokument. Je výsledkem analýzy dokumentu, která se zabývá jak formálními znaky dokumentu, tak i jeho obsahem. Protože je bibliografický záznam určen pro komunikaci informací, nutně musí vyhovovat příslušným národním i mezinárodním normám. Zde pouze naznačíme strukturu a obsah bibliografického záznamu, které jsou do určité míry na normách nezávislé.

Vyjdeme z ČSN 01 0195 - Bibliografický a katalogizační záznam, která platí od 1. 10. 1992 a je určena pro knihovny, informační střediska a další instituce zabývající se uchováváním a šířením textových dokumentů. Struktura bibliografického záznamu je následující:

Soupisné údaje

- autor
- název dokumentu
- notační znak selekčního jazyka klasifikačního typu
- výraz předmětového selekčního jazyka

Vyčleněné údaje

- datum schválení (např. pro výroční zprávy)
- datum obhajoby (např. pro diplomové práce)
- datum účinnosti (např. pro normy nebo zákony)

...

Lokační údaje

- signatura (t.j. adresa dokumentu v primárním fondu)
- sigla (t.j. zkratka instituce vlastnící dokument)

...

Popisné údaje

- další údaje o autorech
- nakladatelství
- vydání (pořadí, datum,...)
- rozsah (počet stran)
- ISBN, ISSN

...

Charakteristika obsahu

- anotace

- referát

Služební údaje

- přírůstkové číslo
- kód zpracovatele

...

Tučně jsou vyznačeny jednotlivé části bibliografického záznamu, pomlčkou jsou označeny údaje patřící do jednotlivých částí. Tři tečky znamenají, že do příslušné části patří i další údaje. Bibliografický záznam musí zahrnovat potřebné údaje pro různé typy dokumentů, např. pro knihy, články v časopisech, diplomové a disertační práce, zákony, atd. Ne všechny údaje je tedy možno vždy uvést. Existují také formy (typy) bibliografických záznamů pro různé účely, existuje např. rozsahem největší záznam pro národní bibliografii, méně údajů se uvádí v základním záznamu, používají se i záznamy pro různé specielní účely.

Význam většiny údajů bibliografického záznamu je zřejmý, u některých se podrobněji zastavíme. Selekčním jazykům zmíněným v části soupisných údajů a anotaci a referátu uváděným u charakteristiky obsahu je věnována následující kapitola.

Signaturu, patřící do lokačních údajů jako adresa dokumentu v primárním fondu, je třeba chápat v souvislosti s klasickou, tedy ne elektronickou formou dokumentů. Sigla jakožto zkratka instituce vlastnící dokument se používá např. pro meziknihovní výpujční službu.

Mezi další údaje o autorech patřící mezi popisné údaje může patřit např. sestavovatel (u sborníku), překladatel, ilustrátor nebo přepracovatel.

ISBN (International Standard Book Number) je číslo, přiřazované knize podle mezinárodně dohodnutých pravidel, které ji jednoznačně identifikuje. Vztahuje se k němu také ČSN 01 0189 - Mezinárodní standardní číslování knih. ISBN je tvořeno čtyřmi skupinami cifer oddělených pomlčkami, například ISBN 80-85623-22-6.

První skupina cifer určuje skupinu vydavatelů, např.:

0	-	publikace v angličtině (USA, UK, Kanada, ...)
2	-	publikace ve francouzštině (včetně kanadských)
3	-	publikace v němčině
80	-	česky nebo slovensky
92	-	publikace mezinárodních organizací

Druhá skupina cifer určuje vydavatele v rámci skupiny vydavatelů, třetí skupina jenoznačně identifikuje dokumentu v rámci vydavatele. Poslední cifra je kontrolní číslo.

Podobně, jako ISBN jednoznačně určuje dokument, tak ISSN (International Standard Serial Number) jednoznačně určuje časopisy a další periodika. Vztahuje se k němu ČSN 01 0187 - Mezinárodní standardní číslování seriálových publikací.

3.3 Selekční jazyky

Údaje uváděné v bibliografickém záznamu mají až na vyjímky (např. abstrakt a referát) povahu kódů v tom smyslu jak jsme použili v úvodu. V této souvislosti se hovoří o selekčních jazycích a rozumí se jimi souhrn více či méně formalizovaných výrazů které se používají při popisu dokumentů s tím, že význam jednotlivých výrazů je vždy dostatečně specifikován. Do selekčních jazyků se nezahrnuje abstrakt a referát. Selekčními se tyto jazyky nazývají proto, že jejich pomocí se provádí výběr (selekce) dokumentů.

Je třeba poznamenat, že kódem zde rozumíme, stejně jako v úvodu, řetězce alfanumerických znaků (jména osob, názvy dokumentů, čísla, datumy, ale i například klíčová slova, atd.), které lze pomocí počítačů třídit a separovat bez ztráty informace. K již několikrát zmíněné ztrátě informace ohledně obsahu dokumentu dochází proto, že obsah vyjádříme pomocí selekčního jazyka. S výrazy selekčního jazyka potom v počítači zacházíme stejně, jako s kódy ve zde uvedeném smyslu.

Rozlišujeme **identifikační** a **věcné selekční jazyky**. Identifikační selekční jazyky se týkají údajů sloužících pro identifikaci dokumentů, vyjadřují se jimi údaje jako autor, název, ISBN, nakladatel, atd. Věcné selekční jazyky slouží pro vyjádření obsahu dokumentů. Je jim věnována většina následující kapitoly.

Kapitola 4

Charakteristika obsahu dokumentů

4.1 Prostředky pro charakteristiku obsahu

Obsahem dokumentu zde rozumíme téma, o kterém pojednává. Dokument se může týkat více témat, v tom případě je třeba rozlišit hlavní téma od vedlejších. Obsah dokumentu lze charakterizovat v zásadě dvojím způsobem. Prvním je popis obsahu souvislým textem v přirozeném jazyce. Obvyklé formy jsou anotace a referát, kterým je věnován odstavec 4.2.

Druhý způsob vychází z použití jednotlivých výrazů přirozeného jazyka, případně čísel nebo alfanumerických řetězců. Jedná se potom o věcný selekční jazyk zmíněný v odstavci 3.3.

Věcné selekční jazyky se vyvíjely na dvou principech. První spočívá v pokrytí celé oblasti vědění, které se dokumenty týkají, hierarchicky uspořádaným systémem tříd. Třídy na stejné hierarchické úrovni jsou vzájemně disjunktní. Hovoříme potom o **systematických selekčních jazycích**, podrobněji o nich pojednává odstavec 4.3. Druhý přístup k věcným selekčním jazykům vychází z popisu obsahu dokumentu pomocí jednotlivých výrazů přirozeného jazyka. Potom se jedná o **předmětové selekční jazyky**.

Selekční jazyky se dělí také na **prekoordinované** a **postkoordinované**. Výrazy prekoordinovaného jazyka jsou předem pevně dány s tím, že je nelze spolu kombinovat. Typickým představitelem prekoordinovaných selekčních jazyků jsou předmětová hesla ve stručnosti zmíněná v odstavci 4.4. Typickým příkladem postkoordinovaných selekčních jazyků jsou klíčová slova, kterým je věnován odstavec 4.5. Předmětová hesla i klíčová slova patří mezi předmětové selekční jazyky.

S používáním předmětových selekčních jazyků je spojeno mnoho problémů souvisících s nejednoznačností přirozeného jazyka. Významnou pomůckou k jejich odstranění je **tezaurus** jako slovník deskriptorů. Základní informace o deskriptorech a tezaurech je v odstavci 4.6.

Systematické selekční jazyky se obvykle zařazují mezi prekoordinované selekční jazyky, i když například Mezinárodní desetinné třídění jako typický systematický selekční jazyk má prostředky i pro vyjádření kombinací jednotlivých tříd. Mezi systematické selekční jazyky patří i Ranganathanovo dvojtečkové třídění, které je postkoordinovaným selekčním jazykem, viz odstavec 4.3.

4.2 Anotace a referát

Anotace je stručná charakteristika dokumentu z hlediska jeho obsahu, určení, formy a jiných rysů. Co se týče obsahu, říká anotace pouze o čem dokument vypovídá. Má vysvětlující nebo doporučující ráz, její délka je zpravidla do 500 znaků. Uvádí se v sekundárních dokumentech, může být součástí primárního dokumentu.

Referát, na rozdíl od anotace neuvádí pouze tema dokumentu, ale i základní informace o jeho obsahu, používá se osnova:

- téma, předmět, charakter a cíl práce,
- použité metody,
- výsledky
- závěry
- oblast využití.

Maximální délka referátu je 2500 znaků, obvyklá délka je cca 1000 znaků. Vycházejí referátové časopisy a sborníky. K nejznámějším patří Current Contents, který vydává Institut for Scientific Information.

Pravidla pro tvorbu anotací a referátů jsou dána v ČSN 01 0194 - Referát a anotace.

4.3 Systematické selekční jazyky

Jak už bylo uvedeno, princip systematických selekčních jazyků spočívá v tom, že se celá oblast vědění, které se popisované dokumenty týkají, pokryje hierarchicky uspořádaným systémem tříd. Třídy na stejné hierarchické úrovni nemají žádné společné prvky.

Tento způsob charakteristiky obsahu je historicky starší než předmětové selekční jazyky a souvisí s potřebou vhodným způsobem uspořádat knihy v knihovně. Přirozeným řešením bylo rovnat knihy podle oborů. Obory byly systematicky pořádány, pořádání knih ve fondech tedy úzce souvisí s klasifikací věd a lidského poznání vůbec.

Při tomto přístupu byly používány systematické katalogy, které odrážely uspořádání knih. Takto byly uspořádány knihy v Alexandrijské knihovně. Uvádí se [KOVÁŘ 84], že její katalog měl 120 svazků, ve kterých bylo zaznamenáno veškeré písemnictví té doby. Byl členěn na poezii, právo, filozofii, historii, rétoriku a různé.

Jedním z nejdůležitějších systematických selekčních jazyků je Deweyovo Desetinné třídění (DDT, DDC), které vytvořil významný americký knihovník M. Dewey. První vydání DDC vyšlo v roce 1876, již ve dvacátých letech našeho století bylo používáno v 95% knihoven USA [KOVÁŘ 84]. Jeho základem je 10 hlavních tříd pokrývajících lidské vědění, tyto třídy se dále hierarchicky dělí, každá na deset skupin, atd. Významnou pomůckou pro informační pracovníky i další uživatele DDT byl tzv. relativní index, který obsahoval abecedně uspořádané názvy pojmů spolu s číselnými znaky tříd, kam byly jednotlivé pojmy zařazeny. Dalším významným systematickým selekčním jazykem je klasifikační jazyk používaný v Library of Congress USA [UNESCO 91].

Velmi známým představitelem systematických selekčních jazyků je také **Mezinárodní desetinné třídění** (MDT, UDC). Vzniklo z Deweyovo Desetinného třídění na počátku 20. století a prošlo postupným vývojem. Jedná se o univerzální, hierarchicky uspořádanou třídící soustavu používající desetinnou notaci. Zahrnuje všechny oblasti lidského vědění a používá se pro

všechny druhy dokumentů. Vztahuje se k němu také ČSN 01 0180 - Mezinárodní desetinné třídění.

MDT používá deset hlavních tříd:

0 Obecnosti.
1 Filosofie.
2 Náboženství.
3 Společenské vědy.
4 v současnosti volná
5 Matematika. Přírodní vědy.
6 Užité vědy. Lékařství. Technika.
7 Umění. Sport. Hry.
8 Jazyky. Jazykověda. Krásná literatura.
9 Zeměpis. Životopisy. Dějiny.

Každá hlavní třída, stejně jako každá třída vzniklá postupným dělením, může být rozdělena na dalších deset nebo méně tříd. Znak pro novou třídu se vytvoří připojením jedné cifry ke znaku pro původní třídu. Trojice cifer se oddělují tečkami.

Příklady tříd vzniklých dělením hlavní třídy 6:

6	Užité vědy. Technika
66	Chemické inženýrství. Chemický průmysl. Příbuzná průmyslová odvětví
669	Hutnictví
666.1	Hutnictví železa
666.18	Výroba oceli

Krom takto vzniklých hlavních znaků má MDT různé možnosti pro vyjádření i složitějších témat dokumentů, například [KOVÁŘ 84]:

622 + 669	znamená hornictví a hutnictví
[622 + 669] (485)	znamená hornictví a hutnictví ve Švédsku
17 : 7	znamená Etika a umění. Etika ve vztahu k umění
7 : 17	znamená Umění a etika. Umění ve vztahu k etice

Lze vyjádřit i formu dokumentu (např. knihy, periodika, sborníky) etnika, národy, čas a další.

Zajímavé, i když málo rozšířené, je Ranganathanovo dvojtečkové třídění. Ranganathan (1892-1972) byl jedním z nejvýznamnějších světových knihovníků, působil v univerzitní knihovně v Madrasu v Indii. Ranganathan vystupoval proti enumerativním (výčtovým) klasifikacím, jejichž cílem bylo explicite zahrnout každé možné téma. Ve svém dvojtečkovém třídění vyšel z 26 hlavních tříd (A - věda všeobecně, B - matematika, ... Z - právo). Místo následného explicitního hierarchického členění těchto tříd se však pracuje s klasifikačními charakteristikami, jejichž uplatněním vznikají různé třídy (nazývají se fasety). Fasety obsahují jednotlivé termíny, označené symboly. Pro označení tématu dokumentu je rozhodující jednak výběr správných faset a jednak správné určení jejich pořadí. Při vytváření výsledného označení tématu dokumentu se původně významným způsobem používala dvojtečka. Ranganathanovo dvojtečkové třídění je příkladem postkoordinovaného systematického selekčního jazyka [KOVÁŘ 84].

4.4 Předmětová hesla

Příklady předmětových hesel jsou [KOVÁŘ 84]:

Traktory - zemědělství - příručky
Polsko - dějiny - přehledy
Děti městské - výchova rodinná - doba předškolní
Zámky (budovy)
Zámky (zařízení zavírací)

Podstatné je, že pro strukturu předmětových hesel platí pevná pravidla, která navíc závisí na národním jazyku. Jiná jsou pro angličtinu, jiná pro češtinu, atd. Důležité je například pořadí substantiva a adjektiva. Velkou nevýhodou je, že tato, někdy složitá pravidla je třeba respektovat i při formulaci dotazu. To prakticky znamená, že při formulaci dotazu musí asistovat školený pracovník. Další nevýhodou předmětových hesel je nemožnost práce se složenými tématy. Tvorbě předmětových hesel je věnována ČSN 01 0188.

4.5 Klíčová slova

Uvedenou nevýhodu předmětových hesel odstraňují **postkooordinované předmětové selekční jazyky**. Jejich principem je vyjádření tématu pomocí volné kombinace klíčových slov. Klíčová slova jsou volena tak, aby co nejvýstižněji charakterizovala obsah dokumentu. Mohou to být jednotlivá slova i krátká slovní spojení. Obsah dokumentu se vyjadřuje několika klíčovými slovy.

Informační požadavek se vyjadřuje vhodnou podmínkou na přítomnost nebo nepřítomnost klíčových slov v dokumentu. V tomto směru je nejrozšířenější booleovský model, kterému je věnována kapitola 6. Jsou však i jiné možnosti, například vektorový model, viz odstavec 9. 1.

Vyhledávání dokumentů relevantních k dotazu formulovanému pomocí klíčových slov se dnes provádí pomocí výpočetní techniky. K tomuto účelu byly používány i mechanické pomůcky, například průhledové karty patentované roku 1915 nebo Zato Coding System patentovaný roku 1947 americkým matematikem C. Mooersem [KOVÁŘ 84].

Velmi složité je stanovení vhodných klíčových slov. Určují se na základě analýzy dokumentu. Pro odbornou literaturu je vhodné vybírat klíčová slova z názvu dokumentu nebo z názvu kapitol a odstavců. Lze také využít abstraktu nebo referátu, případně úvodu nebo závěru dokumentu.

Mnoho problémů přináší mnohotvárnost přirozeného jazyka. Některé z nich lze řešit pomocí desriptorů - předepsaných klíčových slov. Deskriptory jsou spolu s dalšími informacemi uváděny v tezaurech.

4.6 Deskriptory a tezaury

K úspěšnému vyhledávání relevantních dokumentů je nezbytné, aby téma dokumentu i požadavku bylo formulováno pomocí stejného slovníku. To je však značně komplikováno výskytem synonymních výrazů. Tento problém je možno odstranit použitím řízeného slovníku, tedy slovníku podléhajícího jistým pravidlům.

Příkladem řízeného slovníku je **tezaurus**. Je založen na výběru jednoho z možných synonym jako závazného klíčového slova, kterým budou ostatní synonyma nahrazována. Takováto klíčová slova se nazývají **deskriptory**, zbývající synonyma jsou **nedeskriptory**. Tezaurus je slovník, který obsahuje deskriptory a nedeskriptory. Dále jsou v tezauru zachyceny i vztahy

mezi deskriptory. Jsou to jednak hierarchické vztahy podřízenosti a nadřazenosti pojmů a jednak vztahy vzájemné asociace.

Přehled vztahů zaznamenaných v tezauru je v tab. 4.1.

odkaz	zkratka	
	česky	anglicky
z nedeskriptoru na deskriptor	viz	USE
z deskriptoru na nedeskriptor	ekv	UF
na nadřazený deskriptor	nd	BT
na podřazený deskriptor	pd	NT
na asociovaný deskriptor	ad	RT

Tabulka 4.1: Vztahy mezi deskriptory a nedeskriptory v tezauru

Odkaz ad se používá zejména pro vyjádření vztahů:

celek - část celku
předmět - užití
předmět - vlastnost
proces - prostředek
příčina - účinek

Příklad z tezauru INSPEC:

Deskriptor: INFORMATION ANALYSIS

ekvivalentní UF Citation Analysis

podřazený NT Abstracting
Cataloguing
Classification
Indexing

nadřazený BT Information Science

asociovaný RT Information Centres
Information Retrieval
Information Services
Information Storage
Information Use

Kapitola 5

Automatická charakteristika obsahu

5.1 Cíle a principy

Cílem automatické charakteristiky obsahu je automatické přiřazování vhodné charakteristiky obsahu textovým dokumentům umožňující vyhledávání dokumentů na kvalitativně srovnatelné nebo lepší úrovni než to umožňuje intelektuální indexování obsahu.

Jsou dva principiální důvody pro tuto automatizaci. Prvním je snaha o vyloučení subjektivních vlivů při indexování. Při intelektuální charakteristice obsahu tento vliv nelze vyloučit. Stává se dokonce, že tentýž dokument je při opakovaném indexování stejnou osobou prováděném s jistým časovýn odstupem indexován jiným způsobem než při prvním zpracování [CHEN 92].

Druhým důvodem je prudce rostoucí počet textových dokumentů dostupných v elektronické podobě. Krom různých odborných dokumentů jsou v této formě k dispozici i některé české noviny. Není prakticky možné všechny dostupné dokumenty indexovat intelektuálně a už vůbec ne v dostatečně krátké době.

Základním problémem automatické charakteristiky obsahu je (alespoň v současné době) praktická nemožnost napodobit postupy intelektuální indexace počítačem. Potíž je v tom, že intelektuální indexace je založena na porozumění smyslu textu a jeho stručném vyjádření. Automatizace porozumění smyslu textu není prozatím na dostatečné úrovni zvládnuta.

Automatická indexace vychází z předpokladu, že jestliže se některé slovo vyskytuje v textu v dostatečné frekvenci, pak se dokument týká pojmu odpovídajícímu tomuto slovu. V odstavci 5.2 je popsána jednoduchá indexovací metoda, která je jednoduchou aplikací tohoto předpokladu. Ukazuje se [SALTON 89], že zmíněný princip je správný. Pro praktické použití však má jistá omezení, která lze různými způsoby a v různé míře odstranit. Jedna z těchto možností je podrobněji ukázána v odstavci 5.3, další jsou naznačeny v odstavci 5.4.

5.2 Jednoduchá indexovací metoda

Jak už bylo řečeno, jednoduchá indexovací metoda vychází z předpokladu, že jestliže se některé slovo vyskytuje v textu v dostatečné frekvenci, pak se dokument týká pojmu odpovídajícímu tomuto slovu. Dostatečná frekvence se zde definuje jako prahová hodnota pro frekvenci slova, nezávislá na jednotlivých dokumentech. Jedná se tedy o konstantu pro celý fond indexovaných

dokumentů. V dalším ji budeme značit P. Dané slovo se tedy stane klíčovým slovem pro dokument, jestliže jeho frekvence v tomto dokumentu je nejméně P.

Při indexování se však bere ohled na to, že existují slova, kterým neodpovídají žádné pojmy, podle kterých je obvykle zapotřebí vyhledávat. Jedná se například o spojky, předložky nebo slovesa. Příklady konkrétních slov jsou: *a, i, nebo, před, je, byla, souvisí, vyžaduje*. Tato slova se nazývají **stop slova** a jsou z procesu indexování vyloučena. Seznam stop slov se nazývá **stop-list**.

Budeme předpokládat, že máme N dokumentů D_1, D_2,... D_N. Je-li vytvořen seznam stop slov, probíhá indexování ve třech krocích:

1. Z indexovaných dokumentů se vynechají stop slova.
2. Spočtou se frekvence ostatních slov S_1, S_2,..., S_K pro všechny dokumenty. Počet výskytů slova S_j v dokumentu D_i budeme značit $F_{i,j}$.
3. Zvolí se prahová hodnota P, dokumentu D_i se jako klíčová slova přiřadí všechna slova S_j taková, že $F_{i,j} > P$.

Ze zkušeností vyplývá, že častý výskyt slova v dokumentu znamená, že dokument se týká odpovídajícího pojmu. Naopak, pokud se dokument týká pojmu odpovídajícímu nějakému slovu, pak se toto slovo v dokumentu vyskytuje s velkou frekvencí. Pokud tedy použijeme slova s vysokou frekvencí jako klíče, dostaneme velmi pravděpodobně všechny relevantní dokumenty. Jinými slovy, použití slov s vysokou frekvencí jako klíčů podporuje úplnost vyhledávání.

Zásadní problém však může nastat s přesností. Předpokládejme, že máme fond dokumentů specializovaný na nějakou oblast. Příkladem může být fond článků týkajících se databází. Lze očekávat, že v každém článku se bude s velkou frekvencí vyskytovat slovo *databáze*. Slovo *databáze* se tedy stane klíčovým slovem pro všechny (nebo téměř všechny) články. Použijeme-li toto klíčové slovo v dotazu nebude nám to však příliš platné. Pokud bude v konjunkci např. *databáze AND CD-ROM*, bude tato konjunkce ekvivalentní (nebo téměř ekvivalentní) jednodušímu dotazu *CD-ROM*. Na druhé straně, odpovědí na dotaz *databáze OR CD-ROM* budou všechny (nebo téměř všechny) uchovávané články. Problém je v tom, že klíčové slovo *databáze* vyskytující se u všech článků z fondu nepomáhá jednotlivé články od sebe oddělit, neumožňuje tedy zvýšit přesnost vyhledávání. (Dotazy *databáze AND CD-ROM* a *databáze OR CD-ROM* jsou příklady booleovských dotazů, o kterých je pojednáno v odstavci 6.2)

Tento problém jednoduché indexovací metody lze částečně odstranit její modifikací, které je věnován následující odstavec.

5.3 Modifikace jednoduché indexovací metody

Cílem popisované modifikace jednoduché indexovací metody je nalézt taková klíčová slova, která by jednak charakterizovala obsah dokumentu a jednak pomáhala oddělit od sebe jednotlivé dokumenty ve fondu. Jedná se o dva různé požadavky na klíčové slovo. Princip modifikace spočívá v těchto zásadách:

a) Číselně vyjádříme stupeň, ve kterém slovo charakterizuje obsah dokumentu.

b) Číselně vyjádříme také schopnost slova oddělit od sebe jednotlivé dokumenty ve fondu.

c) Z takto získaných dvou čísel vytvoříme vhodným způsobem jedno číslo tak, aby jeho velikost přímo závisela na velikosti každého z nich.

d) Dané slovo prohlásíme za klíčové slovo, jestliže výsledek dle bodu **c)** bude větší než zvolená prahová hodnota.

Konkrétních realizací tohoto postupu existuje samozřejmě více, ukážeme jednu z nich převzatou ze [SALTON 89].

Pro vyjádření stupně, ve kterém slovo charakterizuje obsah dokumentu použijeme, stejně jako v jednoduché indexovací metodě, počet výskytů slova v dokumentu. Zachováme-li stejné značení, jako v předchozím odstavci, pak je to $F_{i,j}$ pro slovo S_j v dokumentu D_i.

Pro vyjádření schopnosti slova oddělit od sebe jednotlivé dokumenty ve fondu vyjdeme z následující úvahy: Nejhorší pro tento účel bude slovo vyskytující se ve všech dokumentech. S klesajícím počtem dokumentů, ve kterých se slovo vyskytuje, roste schopnost slova oddělit dokumenty od sebe. Pokud se však slovo bude vyskytovat jen v jednom nebo v několika málo dokumentech, vzniknou zase jiné potíže, ale těmi se teď nebudeme zabývat.

Z uvedené úvahy plyne, že schopnost daného slova oddělit od sebe jednotlivé dokumenty ve fondu lze vyjádřit funkcí, kde nezávisle proměnnou bude počet dokumentů fondu, ve kterých se slovo vyskytuje a jejíž hodnota bude klesat, pokud počet dokumentů poroste. Funkcí tohoto typu je např.

$$\log(\frac{N}{DF_j})$$

kde N je počet dokumentů ve fondu a DF_j je počet dokumentů, ve kterých se vyskytuje slovo S_j.

Dle bodu **c)** je zapotřebí z hodnot $F_{i,j}$ a $\log(\frac{N}{DF_j})$ vytvořit jedno číslo tak, aby jeho velikost přímo závisela na velikosti obou hodnot, $F_{i,j}$ i $\log(\frac{N}{DF_j})$. Zde se obvykle používá součin, v našem případě to tedy bude

$$W_{i,j} = F_{i,j} * \log(\frac{N}{DF_j}).$$

Tento součin se nazývá váhou slova S_j v dokumentu D_i, budeme ho značit $W_{i,j}$.

Shrneme-li výše uvedené, probíhá indexování pomocí modifikované indexovací metody opět ve třech krocích.

1. Z indexovaných dokumentů se vynechají stop slova.
2. Spočtou se váhy $W_{i,j}$ ostatních slov pro všechny dokumenty.
3. Zvolí se prahová hodnota P, dokumentu D_i se jako klíčová slova přiřadí všechna slova S_j taková, že $W_{i,j} > P$.

Podle zkušeností dává tato metoda lepší výsledky než jednoduchá indexovací metoda [SALTON 89]. Je však možné a vhodné výsledky takto získané výsledky dále vylepšit.

5.4 Další přístupy

V tomto odstavci naznačíme některé další přístupy používané při automatickém indexování.

A) Nejprve si všimněme, že počet výskytů slova S_j v dokumentu D_i označený $F_{i,j}$ nebere v úvahu délku dokumentu D_i ani celkovou frekvenci slova S_j v celém fondu. Jedná se však o údaje, které mohou vypovídat o vhodnosti použít slovo S_j jako klíčové. Různé možnosti jak vzít tyto údaje v úvahu jsou uvedeny v [STROSSA 94].

B) Jednotlivé výskyty určitého slova nemusí mít stejný význam pro charakteristiku obsahu dokumentu. Větší význam bude mít výskyt slova v popisu obrázku, v závěrečném shrnutí, v

nadpisu odstavce případně v nadpisu celého dokumentu. Tyto skutečnosti spolu s řadou dalších bere v úvahu metoda MOZAIKA viz např. [STROSSA 94].

C) Slova s výskytem v mnoha dokumentech je vhodné spojit s dalšími slovy tak, aby výsledné spojení (fráze) mělo užší význam a tedy větší rozlišovací schopnost [SALTON 89]. Například ze slova "informace" je možno vytvořit frázi "obchodní informace". Také tento proces lze automatizovat, je však třeba využít lingvistických metod. Například je vhodné umět ve větě automaticky rozlišit, zda se jedná o spojení slov týkající se podmětu ("obchodní informace") nebo slovesa ("jsou nezbytné") atd.

D) Slova s výskytem ve velmi málo dokumentech je možno nahradit ekvivalentními tezaurovými deskriptory tak, aby se rozšířil jejich význam. (Podle tezauru INSPEC je tedy nedeskriptorový termín "Citation Analysis" nahrazen deskriptorem "INFORMATION ANALYSIS").

E) V případě modifikace jednoduché indexovací metody jsme ukázali, jak lze k danému slovu přiřadit jeho váhu $W_{i,j}$ vyjadřující stupeň v jakém slovo charakterizuje obsah dokumentu. Stejný význam má i frekvence $F_{i,j}$ ze které se vychází při jednoduché indexovací metodě. Přístupy zmíněné v bodech **B)** a **C)** vedou také k výpočtu jisté váhy slova. Doposud jsme uvedli pouze takové metody, ve kterých se získaná váha slova porovná s nějakou prahovou hodnotou a je-li větší, tak se slovo zařadí mezi klíčová slova. Principiálně jiný přístup spočívá v tom, že slovo zařadíme mezi klíčová slova a zapamatujeme si i jeho váhu, kterou použijeme k výpočtu stupně relevantnosti dokumentu k dotazu. Podrobně je tento přístup rozveden v kapitole 8.

Kapitola 6

Booleovský model vyhledávání dokumentů

6.1 Přehledný popis

Prvním významným impulsem pro využití počítačů při ukládání a vyhledávání textových dokumentů byla snaha automatizovat knihovní katalogy a bibliografie. Relativně velmi snadným se stalo vyhledávání všech bibliografických záznamů splňujících danou i poměrně složitou podmínku. Podmínky, které musí požadované bibliografické záznamy splňovat se většinou vyjadřují aplikací booleovských spojek AND, OR a NOT na údaje bibliografického záznamu. Hovoří se proto o booleovském modelu vyhledávání dokumentů, jeho principu je věnován odstavec 6.2.

Zkušenosti s booleovským modelem ukazují, že kvalita vyhledávání dokumentů měřená pomocí úplnosti a přesnosti není příliš vysoká. Podrobněji je těmto otázkám věnován odstavec 6.3. Existuje více způsobů, jak lze v rámci booleovského modelu dosáhnout alespoň dílčího zlepšení přesnosti a úplnosti. Tyto způsoby se stále zdokonalují. Jeden konkrétní vyhledávací systém však zpravidla nemá implementovány všechny možnosti. V odstavcích 6.4 až 6.8 je popsáno několik způsobů, které patří k nejrozšířenějším. Je třeba poznamenat, že některé z nich se používají zejména v plnotextových systémech. Jsou však používány v rámci nějaké booleovské podmínky a proto je uvádíme v kapitole o booleovském modelu.

6.2 Princip booleovského modelu

Booleovský model vyhledávání dokumentů byl vyvinut pro práci s bibliografickými respektive katalogovými databázemi, tedy s databázemi obsahujícími záznamy dokumentů. Oproti běžným relačním databázím mají bibliografické databáze některá specifika. Jedním z nich je velký důraz na práci s textovými údaji. Hlavní odlišnost však spočívá v přístupu k práci s bibliografickými údaji majícími více výskytů v jednom bibliografickém záznamu. Příkladem takového údaje jsou zejména klíčová slova, pro jeden dokument se vždy uvádí několik klíčových slov. Dalším takovým důležitým údajem jsou autoři, kterých je v případě spoluautorství také více.

Je třeba poznamenat, že pomocí relačního modelu by jistě šlo vyjádřit příslušné vztahy mezi bibliografickým záznamem a opakujícími se údaji, tento přístup se však obvykle nepoužívá. Jedním důvodem je, že v době vývoje prvních bibliografických databází nebyl relační ani jiný databázový model na potřebné úrovni k dispozici. Druhým, podstatnějším důvodem je fakt, že byly vyvinuty speciální metody pro rychlé vyhledávání v bibliografických databázích, které budou asi vždy rychlejší, než metody přístupu k datům používané v běžných relačních databázích. Jedná se zejména o invertovaný soubor, který je popsán v kapitole 14.

Podstata booleovského modelu spočívá ve vyjádření požadavku na dokumenty pomocí logických spojek AND, OR a NOT. Příkladem takového zápisu požadavku je výraz

$$počítač\ AND\ síť$$

kde *počítač* a *síť* jsou klíčová slova. Na základě takto formulovaného dotazu budou vyhledány všechny dokumenty (přesněji jejich bibliografické záznamy), mezi jejichž klíčovými slovy je zároveň *počítač* i *síť*.

Na tomto místě je třeba připomenout rozdíl mezi informačním požadavkem a dotazem zdůrazněný v kapitole 2. Výraz "*počítač AND síť*" je dotaz. Odpovídající informační požadavek může znít např.: "Vyhledejte všechny články týkající se počítačových sítí".

Logické spojky se obvykle aplikují na výroky, tedy na tvrzení, o kterých lze rozhodnout, zda jsou či nejsou pravdivá. Rozlišujeme **základní výroky**, tedy samotná tvrzení bez logických spojek a **složené výroky**, které vznikají ze základních výroků aplikací logických spojek.

Z tohoto pohledu je třeba výraz

$$počítač\ AND\ síť$$

chápat jako složený výrok "*počítač* je klíčovým slovem pro daný dokument" AND "*síť* je klíčovým slovem pro daný dokument". Je důležité, že je uveden konkrétní dokument, neboť jenom tak lze rozhodnout zda základní výroky "*počítač* je klíčovým slovem pro daný dokument" a "*síť* je klíčovým slovem pro daný dokument" jsou či nejsou pravdivé.

Jinými slovy řečeno, jsou na základě dotazu

$$počítač\ AND\ síť$$

poskytnuty všechny dokumenty, pro které je výše uvedený složený výrok pravdivý.

Logická spojka AND se v bibliografických databázích chová způsobem odpovídajícím výrokové logice. Totéž platí pro logickou spojku OR. Například na základě dotazu

$$počítač\ OR\ síť$$

získáme všechny dokumenty, mezi jejichž klíčovýni slovy je *počítač* nebo *síť*.

Pozor je třeba dát na spojku NOT, která je v bibliografických databázích často chápána jako zkratka za výraz $AND\ NOT$. Dotazem

$$počítač\ NOT\ síť$$

získáme všechny dokumenty, mezi jejichž klíčovými slovy je počítač ale není síť. Pokud bychom chtěli i v případě spojky NOT dodržet analogii s výrokovou logikou, bylo by třeba tento dotaz zapsat takto:

$$počítač\ AND\ NOT\ síť$$

Dále je třeba zdůraznit, že klíčové slovo nemusí být slovem v gramatickém smyslu. Může se jednat o víceslovný termín , např. "*počítačové sítě*" nebo "*sítě počítačové*". Každý z těchto dvouslovných termínů by byl v našem příkladě asi vhodnější z hlediska charakteristiky obsahu než dvě samostatná slova "*počítač*" a "*síť*". Místo "klíčové slovo" se proto někdy používá "klíčový termín".

Způsob vyjadřování informačních požadavků na vyhledání dokumentů pomocí booleovského modelu jsme ukázali na příkladě klíčových slov. V konkrétních vyhledávacích systémech je možno vždy pracovat i s dalšími bibliografickými údaji. Jedná se zejména o autora, datum vydání, název časopisu, nakladatele, atd. Nemusí se však nutně jednat o všechny údaje z bibliografického záznamu.

6.3 Úplnost a přesnost v booleovském modelu

Zkušenosti a experimenty ukazují, že booleovský model není příliš dokonalý co se týče přesnosti a úplnosti vyhledávání. Připomeňme, že definice přesnosti a úplnosti vychází z tabulky 2.1, kterou zde pro rychlejší orientaci uvádíme ještě jednou jako tabulku 6.1:

	vybráno	nevybráno	
relevantní	A	B	$A+B$
irelevantní	C	D	$C+D$
	$A+C$	$B+D$	

Tabulka 6.1: Počty dokumentů pro hodnocení úspěšnosti vyhledávání

Zde A značí počet vybraných dokumentů relevantních k informačnímu požadavku, atd.

Na zjednodušených příkladech ukážeme mechanismy které způsobují, že relevantní dokument není vyhledán nebo naopak, že je vyhledán irelevantní dokument. Kořeny těchto potíží jsou v:

a) Nedokonalém vyjádření obsahu klíčovými slovy, nedokonalost přitom může pramenit jak v nedokonalých vyjadřovacích schopnostech klíčových slov, tak i v nedokonale provedeném indexování.

b) Rozdílném vyjádření jednoho tematu indexátorem a uživatelem.

c) Principiálních možnostech vyjádřit pomocí spojek AND, OR a NOT informační požadavek.

Příklad 6.1:

Předpokládejme že potřebujeme odborné články zabývající se datovými strukturami nebo algoritmy pro ukládání dat na CD-ROM, které má svá specifika. Máme k dispozici informační fond obsahující články z oblasti výpočetní techniky, jejichž obsah je indexován pomocí klíčových slov. Požadavek na vyhledání článků vyjádříme dotazem

$$(\textit{datová struktura}\ OR\ \textit{algoritmus})\ AND\ \textit{CD-ROM}.$$

To znamená, že budou vyhledány všechny články, mezi jejichž klíčovými slovy je *CD-ROM* a alespoň jeden z termínů *datová struktura* nebo *algoritmus*.

Ve fondu může existovat článek, který se podrobně zabývá rozmanitými typy algoritmů a jejich vlastnostmi, mezi jiným i specifickým algoritmem obzvlášť vhodným pro ukládání dat na CD-ROM. Tento článek tedy je relevantní vzhledem k našemu informačnímu požadavku. Fakt, že se jedná o algoritmy právě pro CD-ROM však může být z hlediska článku okrajový a proto se termín CD-ROM mezi klíčová slova nedostane. Znamená to však také, že tento, pro nás relevantní článek nebude na základě výše uvedené podmínky vybrán.

Na druhé straně může být ve fondu článek, který popisuje CD-ROM obsahující knihovnu algoritmů pro numerická řešení diferenciálních rovnic popisujících dynamiku stavebních konstrukcí. Mezi klíčovými slovy tohoto článku patrně budou termíny CD-ROM a algoritmus. Článek tedy bude na základě výše uvedené podmínky vybrán, i když ho v žádném případě nelze považovat za relevantní vzhledem k našemu informačnímu požadavku.

Oba uvedené nedostatky lze přičíst na vrub *nedokonalým vyjadřovacím schopnostem klíčových slov.*

Příklad 6.2:

Předpokládejme, že nás zajímají zprávy ČTK týkající se počítačových sítích. V některých případech se však stává, že zpráva zřetelně se týkající počítačových sítí je indexována obecným termínem "elektrotechnika". Příčinou může být, že indexátor nemá hlubší znalosti z příslušné oblasti a proto použije obecný termín. Za tímto problémem lze vidět *nedokonalé indexování.* Na druhé straně však nelze očekávat, že indexátor bude odborníkem ve všech oborech, jichž se jednotlivé zprávy týkají.

Pro indexátora - odborníka na výpočetní techniku nebude těžké rozpoznat, že se zpráva týká počítačových sítí a termín "počítačová síť" bude použit jako klíčový. Problém může však vzniknout v tom, že uživatel použije jako klíčový termín "síť počítačů" místo "počítačová síť". Je myslitelný i dotaz počítač *AND* síť. Problém spočívá v *rozdílném vyjádření jednoho tématu indexátorem a uživatelem.*

Tento problém lze principiálně odstranit použitím tezauru, který však nebývá vždy k dispozici. S tezaurem je mimo jiné spojen problém jeho aktualizace pro rychle se vyvíjející obory, ve kterých však nebývá ustálená terminologie.

Příklad 6.3:

Předpokládejme, že máme dokumenty týkající se mimo jiné počítačových a telekomunikačních sítí. Dále předpokládejme, že k indexování jsou použity jednoslovné výrazy, např. "počítač", "síť", nebo "telekomunikace". To může být výsledkem méně kvalifikované indexace při které nejsou použity termíny jako "počítačová síť" nebo "telekomunikační síť". Může to být i výsledkem automatizovaně provedené indexace založené na frekvencích slov.

Pokud pro vyhledání dokumentů týkajících se počítačových sítí použijeme dotaz

počítač AND síť,

uniknou nám mimo jiné dokumenty, které nemají mezi klíčovými slovy termín "síť". Takový dokument však přesto může být relevantní, že "síť" není mezi jeho klíčovými slovy může být důsledkem těsně podprahové hodnoty jeho frekvence.

Pokud použijeme dotaz

počítač OR síť,

určitě získáme řadu irelevantních dokumentů týkajících se např. telekomunikačních, rybářských, houpacích, špionážních nebo jiných sítí.

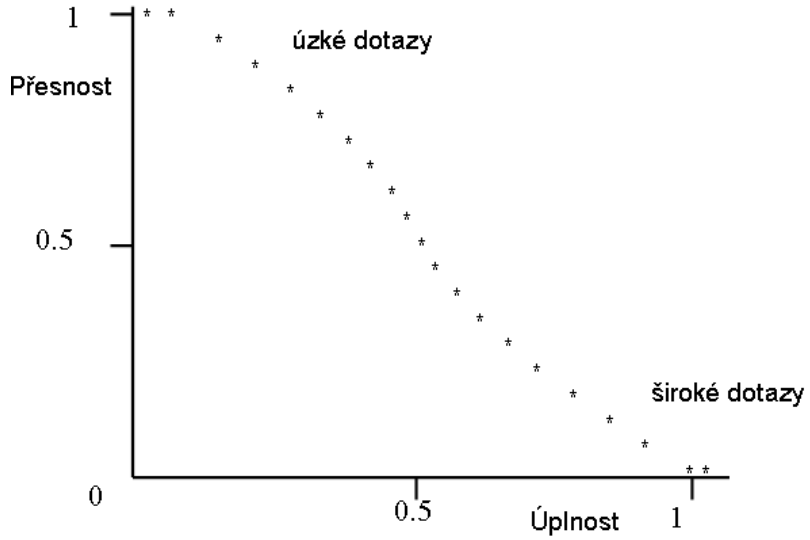
Problém spočívá v *"ostrosti" booleovských operací*, kterými lze dokumenty rozdělit pouze na relevantní a irelevantní, bez žádného mezistupně.

O možnostech booleovského modelu vypovídá i tzv. experiment se STAIRS [ŽBIRKA 93]. Cílem tohoto experimentu bylo ověřit kvalitu vyhledávání na základě booleovského modelu. Vyhledávání se týkalo databáze právnických textů zahrnující soudní případy a jejich dokumentaci, např. protokoly z výslechů. Databáze obsahovala okolo 40 000 právnických dokumentů, celkem 350 000 stran textu.

Experiment spočíval v určení přesnosti a úplnosti vyhledání relevantních dokumentů pro 51 informačních požadavků. Jednalo se o podklady pro soudní případy. Bylo proto důležité, aby bylo dosaženo vysoké úplnosti. Pro každý informační požadavek byly podklady vyhledány počítačem na základě booleovského modelu a poté byla určena přesnost a úplnost. Přáním bylo dosažení úplnosti okolo 75%.

Výsledek byl však podstatně horší. Bylo dosaženo úplnosti asi 20%, přičemž přesnost byla asi 80%. Tyto značně neuspokojivé výsledky byly jedním z podnětů dalšího vývoje systémů pro vyhledávání textových dokumentů.

Pro booleovský model je typický vztah nepřímé úměrnosti mezi přesností a úplností vyhledávání. Na základě experimentů [SALTON 83] bylo ověřeno, že vztah přesnosti a úplnosti se dá vyjádřit křivkou dle obr. 6.1.



Obrázek 6.1: Vztah mezi úplností (R) a přesností (P)

Úzkým dotazem se rozumí dotaz na jehož základě získáme relativně malý počet dokumentů. Obvykle se toho dosáhne použitím spojky *AND*, příkladem je výše uvedený dotaz

počítač AND síť.

Širokým dotazem rozumíme dotaz, kterým získáme relativně velký počet dokumentů. Typickým širokým dotazem je dotaz využívající spojku *OR*, např.

počítač OR síť.

6.4 Rozšiřování slov

Nízká úplnost vyhledávání souvisí mimo jiné s tím, že slova se vyskytují v různých gramatických tvarech. Podstatné jméno *informace* se může vyskytovat i ve tvarech

informaci, informací, informacích, informacím, informacemi.

Předpokládejme, že podle slov v názvu vyhledáváme dokumenty týkající se informací. Pokud v dotazu uvedeme pouze termín informace, uniknou nám všechny dokumenty v jejichž názvu se sice slovo informace nevyskytuje, ale vyskytuje se tam některý z dalších možných tvarů - informaci, informací, informacích, informacím nebo informacemi. To bude mít velmi pravděpodobně za následek značné snížení úplnosti.

Jedna z možností, jak tento problém řešit je uvést v dotazu všechny přípustné tvary slova informace spojené logickou spojkou *OR*, tedy

informace OR informaci OR informací OR informacích OR informacím OR informacemi.

Většina vyhledávacích systémů však umožňuje tento zápis provést podstatně stručněji pomocí **pravostranného rozšiřování slov** (**suffix**). To spočívá v tom, že na konci výrazu uvedeme vyhrazený znak (velmi často znak "*") a systém krom uvedeného výrazu automaticky pracuje i se všemi jeho pravostrannými rozšířeními.

Zmíněný problém více tvarů slova *informace* v názvech dokumentů lze tedy řešit zápisem výrazu

*informac** .

Je třeba zdůraznit, že se jedná o mechanické rozšiřování, bez přihlédnutí ke gramatickým pravidlům nebo k významovým souvislostem. Budeme-li podle slov v názvu vyhledávat dokumenty o Praze a použijeme zápis

*Prah** ,

unikne nám např. dokument s názvem "O Praze historické". Pokud použijeme zápis *Pra** ,

můžeme dostat i dokument "Pravěké osídlení Brněnska".

Analogicky k pravostrannému rozšíření lze zavést **levostranné rozšíření** (**prefix**) nebo **infixové rozšíření**. Příkladem zápisu levostranného rozšíření je výraz

**ie*,

na jehož základě jsou vyhledávány záznamy obsahující např. slova *filosofie, filozofie, filologie, filharmonie, historie, chemie* nebo *Žofie*. Příkladem zápisu infixového rozšíření je výraz

*fil*ie*,

na jehož základě jsou vyhledávány dokumenty obsahující např. výrazy *filosofie, filozofie, filologie* nebo *filharmonie*, slova *historie, chemie* a *Žofie* nebudou brána v úvahu.

Pravostranné rozšíření je používáno podstatně častěji než ostatní rozšíření. Důvodem je jeho snadná implementace pomocí invertovaného souboru, podrobněji viz kapitolu 14.

Někdy bývá k dispozici i možnost vyjádřit, že na určité pozici může být libovolný znak, ne však již řetězec znaků. Pozice bývá také vyznačena specielním znakem, používá se např. "?". Na základě výrazu

filo?o?ie

jsou tedy vyhledány dokumenty obsahující slova *filosofie, filozofie* nebo *filologie*.

Poznamenejme, že stejný způsob zápisu je možný v rámci operačního systému MS DOS.

6.5 Stemování

Rozšiřování slov zmíněné v předchozím odstavci řeší problém výskytu slov v různých gramatických tvarech jen nedokonale. Krom již uvedených příkladů je možno poukázat i na problémy se slovem "věda". Pokud použijeme pravostranné rozšíření a pro vyhledávání zadáme termín "*věd**", systém si bude všímat i slov *vědma, vědátor, vědomost*. Přesnější řešení nabízí *stemování*. Principem je převedení slova na jeho kmen (anglicky stem).

Převedení se automaticky provádí při indexování dokumentů i při zpracování dotazu. Při zjišťování relevance dokumentu k dotazu je potom rozhodující shodnost gramatických kmenů porovnávaných výrazů v dokumentu a dotazu. Výsledkem zpravidla je zpřesnění vyhledávání oproti pravostrannému rozšiřování.

Pro stemování se používá i výraz "lematizace". Je s ním spojena řada zejména lingvistických problémů, jejichž studium přesahuje rozsah těchto skript. Podrobnější zpracování lze nalézt v [STROSSA 94].

6.6 Proximitní operátory

Proximitní operátory slouží pro vyjádření omezení vzájemné pozice vyhledávaných termínů. Příklady takových omezení jsou:

- termíny se musí vyskytovat ve stejné větě,

- mezi termíny nesmí být více než zadaný počet slov a na jejich pořadí přitom nezáleží,
- mezi termíny nesmí být více než zadaný počet slov přičemž pořadí termínů je pevně dáno.

Příklad 6.4:

Předpokládejme, že máme k dispozici plnotextovou databázi článků týkajících se strojírenského průmyslu, automaticky indexovanou na základě frekvencí jednotlivých slov. Dále předpokládejme, že nás zajímají články o řízení kvality. Kvalita i řízení jsou v této oblasti často používané pojmy, jistě tedy bude řada článků mezi jejichž klíčovými slovy bude jak "řízení", tak i "kvalita". Ne všechny takové články se však nutně musí týkat řízení kvality. Jednoduchým protipříkladem budou články zabývající se kvalitou řízení. Takových irelevantních článků se zbavíme pokud budeme požadovat, aby termíny "řízení" a "kvalita" následovaly bezprostředně po sobě a v uvedeném pořadí.

Příklad 6.5:

Předpokládejme, že máme stejnou databázi jako v předchozím příkladu, a že nás zajímají články o využívání počítačů při řízení kvality. Počítač je také často používaný pojem, jistě tedy bude řada článků mezi jejichž klíčovými slovy bude jak "kvalita", tak i "počítač". Ne všechny takové články se však nutně musí týkat použití počítačů při řízení jakosti. Například je možné, že článek obecně popisuje podnik včetně jeho bohatého vybavení počítačovou technikou a v následující části se podrobněji zabývá chemickou podstatou povrchových úprav používaných materiálů, což s kvalitou výrobků úzce souvisí. Takový článek se nemusí vůbec týkat využívání počítačů při řízení kvality.

Pokud budeme požadovat, aby se oba termíny vyskytovaly v jedné větě zvýšíme pravděpodobnost že získáme relevantní článek. Této podmínce vyhovuje např. následující věta: "*Při řízení kvality jsou podstatným způsobem používány počítače.*"

Pokud však v článku bude uvedeno "*Ve všech fázích vývoje i výroby jsou používány počítače. Zejména při řízení kvality hrají podstatnou roli.*", podmínka na výskyt obou termínů v jedné větě nepomůže. Pomůže však požadavek, aby oba termíny nebyly od sebe vzdáleny o více než o 5 slov.

Ve všech uvedených příkladech proximitních omezení je obzvláště v případě češtiny vhodné, aby byly automaticky brány v úvahu i různé gramatické tvary slov. Tedy, budeme-li hledat bezprostředně po sobě následující termíny "řízení" a "kvalita", bude za relevantní dokument považován i článek ve kterém se vyskytuje výraz "řízením kvality".

6.7 Využití tezauru

Metody naznačené v odstavcích 6.4 až 6.6 využívají pro zvýšení přesnosti a úplnosti vyhledávání operace s různými tvary slov nebo požadavky na jejich vzdálenost. Jedná se tedy o formální operace, bez přihlédnutí k významu slov. Tímto způsobem však nelze vyřešit problémy spojené s významem slov. Takový problému například vznikne, pokud je v článcích zmíněných v příkladech 6.5 a 6.6 použit termín "jakost" místo "kvalita".

Synonymita slov spolu s jejich dalšími významovými vztahy jsou zachyceny v tezauru (viz odstavec 4.6). Pokud je k dispozici tezaurus v elektronické formě, může vyhledávací systém automaticky doplňovat synonyma z tezauru. Je-li tedy například použito klíčové slovo jakost, systém automaticky vytvoří výraz jakost *OR* kvalita a dále s ním pracuje. Možnosti automatické práce s tezaurem v rámci rozsáhlejšího systému založeného na využívání umělé inteligence jsou popsány v kapitole 12.

6.8 SOUNDEX

SOUNDEX je metoda pro vyhledávání podobně znějících slov. Byla vytvořena pro vyhledávání v situacích, kdy známe pouze nepřesné znění vyhledávaného slova. Lze ji použít např. při hledání v telefonním seznamu na základě nepřesného znění jména.

Její princip spočívá v tom, že se slovům přiřazují podle jistých pravidel alfanumerické kódy. Pravidla jsou vytvořena tak, aby stejně znějícím slovům byly přiřazeny stejné kódy. Pravidla byla vytvořena pro angličtinu, pro češtinu je jejich použití omezené i když ne zcela irelevantní. Problémy souvisí mimo jiné s bohatějším českým tvaroslovím. Další informace včetně popisu příslušného algoritmu lze získat v [STROSSA 94].

Kapitola 7

Systémy pro práci s plnými texty

7.1 Charakteristické rysy

Zatímco různé bibliografické vyhledávací systémy vznikly před rozšířením počítačů, existence systémů pro práci s plnými texty je užíváním počítačů podmíněna. Jednak proto, že elektronická forma textu na počítači vzniká a jednak proto, že manipulace s texty v rozsahu ve kterém to tyto systémy dělají není bez počítače možná. Plnotextové systémy kopírují některé funkce klasických automatizovaných bibliografických systémů, využívají však dostupnosti plného textu k rozšíření vyhledávacích možností a ke spolupráci s textovými editory.

Kopírování funkcí bibliografických systémů spočívá zejména v umožnění vyhledávání podle předem stanovených částí textu. Lze tedy vyhledávat například podle autora, vydavatele, slov z názvu, roku vydání atd. Součástí vyhledávacího systému bývá i možnost stanovit pevné části textu, podle kterých se bude vyhledávat.

Relativně jednoduchým rozšířením vyhledávacích možností založeným na dostupnosti plného textu je vyhledávání podle každého slova z textu, přičemž je možno klást podmínky na přítomnost slov pomocí booleovských spojek AND, OR a NOT. Je však možno brát v úvahu i frekvence slov, případně ve vztahu k délce textu. Některé z možností jsou ukázány v kapitole 8.

7.2 Příklady komerčních systémů

Cílem tohoto odstavce je stručně upozornit na tři plnotextové systémy dostupné v ČR. V žádném případě není záměrem podat úplný přehled dostupných plnotextových systémů, pro uvedení v tomto přehledu nebyla rozhodující ani celková kvalita produktu. Důvodem jejich uvedení je pouze to, že zahrnují některé další možnosti, které nejsou u jiných systémů vždy implementovány. Neznamená to však, že tyto systémy jsou jediné, které uvedené možnosti mají.

Nejprve se zmíníme o systému **TOPIC**. Jedná se o produkt americké firmy VERITY, distributorem v ČR je firma TOVEK. TOPIC umožňuje vyhledávání relevantních dokumentů nejen pomocí běžných booleovských podmínek, ale i pomocí uživatelem definovaných pojmů. Pojmy mohou být hierarchicky strukturovány, při definici nového pojmu lze využít i pojmy dříve definované. Další podrobnosti jsou v kapitole 10. Charakteristickým rysem systému TOPIC je možnost spolupráce s širokou škálou různých textových editorů. Zajímavá je i možnost exportu údajů o frekvencích slov.

Dalším zajímavým systémem je **Super Text** firmy 5D software. Uvádíme jej zde zejména proto, že spojuje výhody plnotextového systému s možnostmi hypertextu. To přináší v někte-

rých případech podstatnou výhodu. Tvorba hypertextových aplikací je zřejmě prvotním cílem tohoto systému. Systém umožňuje i práci s obrázky.

Nakonec se zmíníme o systému **ByllBase** firmy Byll software. Je to pro jeho schopnost práce s homonymi, která zvyšuje přesnost vyhledávání. Firemní materiály uvádějí jako příklad vyhledávání dokumentů týkajících se slova hnát ve smyslu "utíkat". Systém ví, že slovo hnát se používá i ve smyslu "končetina". Díky vlastní metodě zpracování je při vyhledávání dokumentů obsahujících slovo hnát ve smyslu "utíkat" možno vyloučit dokumenty obsahující toto slovo v jiném smyslu.

Kapitola 8

Rozšiřování booleovského modelu

8.1 Cíle rozšiřování

Booleovský model vyhledávání dokumentů je založen na využití booleovských spojek AND, OR a NOT. To znamená, že nemůže být respektován přirozený a podstatný fakt, že některá klíčová slova charakterizují obsah dokumentu více a jiná méně. Jak lze tento fakt numericky vyjádřit pomocí váhy je ukázáno v kapitole 5 o automatickém indexování. Booleovské spojky umožňují pracovat pouze s vahami 1 (jedná se o klíčové slovo) a 0 (nejedná se o klíčové slovo).

Stejně tak booleovský model neumožňuje vyjádřit fakt, že některý termín použitý v dotazu je důležitější než jiný. Může se např. stát, že se zajímáme o textové databáze jak pro WINDOWS tak i pro DOS, přičemž nás zajímají hlavně databáze pro WINDOWS, pro DOS spíše okrajově. Tento fakt by jistě bylo možno vyjádřit vhodnými vahami u jednotlivých termínů v dotazu, např.

$$[\textit{textové databáze}, 1.0]\ AND\ ([\textit{WINDOWS}, 0.9]\ OR\ [\textit{DOS}, 0.2])$$

kde termínu *textové databáze* je přiřazena váha 1.0, termínu *WINDOWS* je přiřazena váha 0.9 a termínu *DOS* váha 0.2. Logické spojky AND a OR však neumožňují tyto váhy zpracovat.

Pokud by bylo možno vzít v úvahu váhy jednotlivých termínů, pak přirozeným důsledkem vyhodnocení dotazu pro každý dokument by byl stupeň relevance dokumentu k dotazu pomocí těchto vah vypočítaný. Výsledkem dotazu by tedy mohl být seznam dokumentů seřazených podle stupně relevance. Tento typ výstupu je v rámci booleovského modelu nerealizovatelný.

Jistým nedostatkem booleovského modelu je i "tvrdost" logické spojky AND. Například pro dotaz

$$\textit{USA}\ AND\ (\textit{Gorbačov}\ OR\ \textit{Stalin})$$

je dokument, který nemá jako klíčové slovo USA irelevantní. Je však myslitelné, že takový dokument mající mezi klíčovými slovy jak *Gorbačov* tak *Stalin* by přesto mohl být zajímavý i když třeba s podstatně menším stupněm relevance.

Na základě výše uvedeného rozboru je zřejmé, že má smysl zabývat se hledáním takového rozšíření booleovského modelu, které by umožňovalo:

1. rozlišení důležitosti klíčových slov v dokumentu;
2. rozlišení důležitosti klíčových slov v dotazu;

3. řazení vybraných dokumentů podle stupně relevance k dotazu;

4. odstranění tvrdosti konjunkce.

Ukážeme dva přístupy jak těchto cílů dosáhnout. Jsou převzaty ze [SALTON 83], kde jsou uvedeny i další možnosti. V závěru kapitoly jsou porovnány výsledky dosažené pomocí booleovského modelu s výsledky získanými pomocí dále popsaných rozšíření booleovského modelu.

8.2 Rozšíření pomocí fuzzy logiky

Fuzzy logika je rozšířením booleovské logiky v tom smyslu, že připouští různé úrovně pravdivosti výroků. Jednalo se původně o ryze teoretickou záležitost, která se však mimo jiné díky snadným možnostem práce s plynule se měnící úrovní pravdivosti výroků dočkala široké škály aplikací. Teoretické základy fuzzy logiky jsou např. v [NOVÁK 90].

Poznamenejme, že anglické slovo *fuzzy* se do češtiny překládá mimo jiné jako *matný, mlhavý, neostrý, ale i chomáčovitý, chmýřivý, kučeravý, zakalený* nebo *nalíznutý.*

V booleovské logice je každý výrok buď pravdivý nebo nepravdivý. Místo logických hodnot *pravda* a *nepravda* se často používají číselné hodnoty 1 (místo logické hodnoty *pravda*) a 0 (místo logické hodnoty *nepravda*). Někdy se říká, že pravdivost výroku je 1 nebo že pravdivost výroku je 0. Ve fuzzy logice je pravdivost výroku číslo z intervalu $< 0, 1 >$.

Pravdivost výroku U budeme značit $\Pr(U)$. Ve fuzzy logice tedy může mít např. výrok "Míč je veliký" pravdivost 0.7, což symbolicky zapisujeme

$$\mathbf{Pr}(\text{"Míč je veliký"}) = 0.7.$$

Podrobnější interpretace faktu, že nějaký výrok má pravdivost 0.7 přesahuje rozsah těchto skript.

Pro účely rozšíření booleovského modelu budeme různé úrovně pravdivosti výroků využívat pro rozlišení důležitosti klíčových slov v dokumentech. Budeme tedy stejně jako dříve pracovat s výroky typu "S je klíčové slovo pro dokument D.", ale s tím rozdílem, že pravdivost tohoto výroku bude číslo z intervalu $< 0, 1 >$. Jestliže bude například

$$\mathbf{Pr}(\text{"}\textit{Počítač}\text{ je klíčové slovo pro dokument } D\text{"}) = 0.9$$

budeme tomu rozumět tak, že klíčové slovo *počítač* má v dokumentu D důležitost 0.9. Většinou se říká, že *počítač* má v dokumentu D váhu 0.9, tuto formulaci budeme používat i v dalším. Budeme vždy předpokládat, že váha slova je číslo z intervalu $< 0, 1 >$.

V booleovském modelu je dotazem výrok (obvykle složený) týkající se přítomnosti klíčových slov v dokumentu. Podobně je tomu i při použití fuzzy logiky. Při vyhodnocení dotazu pro jednotlivé dokumenty se však berou v úvahu různé stupně pravdivosti příslušných základních výroků dané vahami jednotlivých klíčových slov. Pro pravdivost výroků vytvořených pomocí logických spojek OR, AND a NOT platí následující pravidla (U a V jsou výroky):

$\mathbf{Pr}(U\ OR\ V) = \max(\ \mathbf{Pr}(U), \Pr(V))$

$\mathbf{Pr}(U\ AND\ V) = \min(\ \mathbf{Pr}(U), \Pr(V))$

$\mathbf{Pr}(\text{NON } U) = 1 - \mathbf{Pr}(U)$

V tabulce 8.1. jsou ukázány příklady hodnot výroků $U\ OR\ V$, $U\ AND\ V$, a NON U pro různé hodnoty $\mathbf{Pr}(U)$ a $\mathbf{Pr}(V)$.

$\mathbf{Pr}(U)$	$\mathbf{Pr}(V)$	$\mathbf{Pr}(U\ OR\ V)$	$\mathbf{Pr}(U\ AND\ V)$	$\mathbf{Pr}(\text{NON}\ U)$
1	1	1	1	0 *
1	0.7	1	0.7	0
1	0.1	1	0.1	0
1	0	1	0	0 *
0.9	1	1	0.9	0.1
0.2	1	1	0.2	0.8
0	1	1	0	1 *
0.7	0.9	0.9	0.7	0.3
0.9	0.1	0.9	0.1	0.1
0.1	0.3	0.3	0.1	0.7

Tabulka 8.1: Příklady hodnot složených výroků ve fuzzy logice

Hvězdičkami jsou označeny případy, kdy oba výroky U i V nabývají booleovských hodnot 1 a 0. Všimněme si podstatného faktu, že v tomto případě jsou hodnoty složených výroků stejné jako v případě booleovského modelu.

Vyhodnocování dotazu ukážeme na příkladu . Budeme pracovat s dotazem *textové databáze* AND (*WINDOWS OR DOS*) který označíme Q. V tabulce 8.2 jsou uvedeny váhy klíčových slov *textové databáze*, *WINDOWS* a *DOS*) pro dokumenty X, Y a Z. Výraz **Pr**(Q,dokument) značí hodnotu dotazu Q pro daný dokument.

	váha klíčových termínů			
dokument	*textové databáze*	*WINDOWS*	*DOS*	**Pr**(Q,dokument)
X	0.9	0.1	0.8	0.8
Y	0.0	0.7	0.7	0.0
Z	0.7	0.9	0.8	0.7

Tabulka 8.2: Příklady hodnot dotazu ve fuzzy logice

Vyhodnocení je založeno na použití výše uvedených pravidel pro výpočet pravdivosti výroků vytvořených pomocí logických spojek OR, AND a NOT. Pravdivost základních výroků je dána vahami uvedenými v tabulce 8.2.

Formální postup výpočtu naznačíme pro dokument X. Při vyhodnocování dotazu

textové databáze AND (*WINDOWS OR DOS*)

pro dokument X počítáme vlastně pravdivost výroku

(”*textové databáze* je klíčové slovo pro dokument X”) AND ((”*WINDOWS* je klíčové slovo pro dokument X”) OR (”*DOS* je klíčové slovo pro dokument X”))

kde

Pr(”*textové databáze* je klíčové slovo pro X”) = 0.9

Pr(”*WINDOWS* je klíčové slovo pro dokument X”) = 0.1

Pr(”*DOS* je klíčové slovo pro dokument X”) = 0.8.

Aplikací výše uvedených pravidel dostaneme **Pr**(Q, X) = 0.8. Hodnotu **Pr**(Q, X) je možno považovat za stupeň relevance dokumentu X k dotazu Q. Dokumenty lze tedy seřadit podle stupně relevance k dotazu, v případě dokumentů X, Y a Z z tab. 8.2 je pořadí X - Z - Y.

Aplikací fuzzy logiky na váhy klíčových slov v dokumentech se z cílů stanovených na konci odstavce 8.1 tedy podařilo splnit cíle

1) rozlišení důležitosti klíčových slov v dokumentu;

3) řazení vybraných dokumentů podle stupně relevance k dotazu.

V následujícím odstavci ukážeme, jak lze pomocí fuzzy logiky splnit i druhý cíl:

2) rozlišení důležitosti klíčových slov v dotazu.

8.3 Váhy klíčových slov v dotazu

V odstavci 8.1 jsme zmínili možnost vyjádřit pomocí vhodných vah důležitost jednotlivých termínů v dotazu. Jako příklad jsme uvedli dotaz

$$[\textit{textové databáze},1.0]\ AND\ ([\textit{WINDOWS}, 0.9]\ OR\ [\textit{DOS}, 0.2])$$

kde termínu *textové databáze* je přiřazena váha 1.0, termínu *WINDOWS* je přiřazena 0.9 a termínu *DOS* váha 0.2. V dalším budeme tento dotaz označovat Q_1.

Přiřazení váhy je subjektivní záležitostí uživatele. Otázkou je, jak takový dotaz vyhodnotit, pokud jsou váhy klíčových slov také u dokumentů. Způsob vyhodnocení vysvětlíme na příkladu dokumentu X z tabulky 8.2.

Klíčové slovo *WINDOWS* má v dotazu váhu 0.9, v dokumentu má váhu 0.1. Jeho celkový význam jakým přispívá k relevanci dotazu a dokumentu by měl přímo záviset na obou vahách. Celkový význam slova lze tedy stanovit jako součin obou vah, tedy 0.09. Vzhledem k tomu, že pracujeme s vahami z intervalu $< 0, 1 >$, bude i jejich součin vždy v intervalu $< 0, 1 >$. Na tento součin se lze dívat také jako na pravdivostní hodnotu výroku "*WINDOWS* je relevantní k dotazu Q_1 v dokumentu X". Platí tedy

Pr("*WINDOWS* je relevantní k dotazu Q_1 v dokumentu X") = 0.09

Obdobně lze určit:

Pr("DOS je relevantní k dotazu Q_1 v dokumentu X") = 0.16
Pr("textové databáze je relevantní k dotazu Q_1 v X") = 0.9

Stupeň relevance dotazu Q_1 tedy můžeme určit jako pravdivostní hodnotu složeného výroku

("*textové databáze* je relevantní k dotazu Q_1 v X") AND ("*WINDOWS* je relevantní k dotazu Q_1 v dokumentu X") OR (" DOS je relevantní k dotazu Q_1 v dokumentu X").

s výše uvedenými hodnotami základních výroků. Výsledek je 0.16.

Tímto způsobem lze dosáhnout všechny cíle uvedené na konci odstavce 8.1 krom cíle

4) odstranění tvrdosti konjunkce.

Příčina je ve způsobu výpočtu pravdivostní hodnoty pro konjunkci, kde platí

$$\mathbf{Pr}(U\ AND\ V) = \min(\mathbf{Pr}(U), \mathbf{Pr}(V)).$$

To souvisí s další výhradou proti výpočtu relevance dotazu a dokumentu pomocí fuzzy logiky. Výhrada se týká toho, že na výslednou hodnotu mají vliv převážně základní výroky nabývající extremních hodnot. Jestliže totiž počítáme hodnotu libovolně dlouhé konjunkce je výsledek vždy roven minimální pravdivostní hodnotě ze všech členů konjunkce, ostatní členy konjunkce výsledek nijak neovlivní. Analogicky se chová maximum pro disjunkci.

Tento nedostatek je možno odstranit například pomocí geometrického rozšíření booleovského modelu, kterému je věnován následující odstavec.

8.4 Geometrické rozšíření

Podstata geometrického rozšíření booleovského modelu je podobná jako u rozšíření pomocí fuzzy logiky. Spočívá v tom, že se připouští váhy vyjadřující důležitost klíčových slov v dokumentu případně i v dotazu a že je dán předpis jak spočítat váhu složených výroků tvořených pomocí spojek AND, OR a NOT. Rozdíl spočívá v různých předpisech pro výpočet vah složených výroků.

V případě geometrického rozšíření je tento předpis odvozen z geometrické představy dokumentu jako bodu v prostoru. Počet rozměrů prostoru závisí na počtu klíčových slov. Protože nám jde pouze o ukázání principu, budeme pracovat pouze se dvěma klíčovými slovy. Následující úvahy se tedy budou odehrávat v rovině.

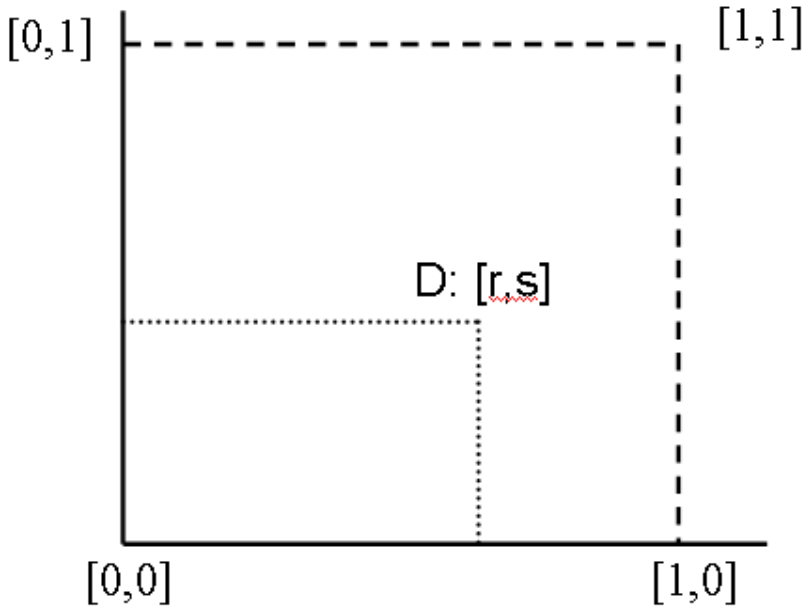
Předpokládejme, že máme dokument **D** se dvěma klíčovými slovy, která budeme dále značit U a V. Dále předpokládejme, že U má v dokumentu **D** váhu **r** a V má váhu **s**. V případě booleovského modelu je $\mathbf{r} = 1$ nebo $\mathbf{r} = 0$, stejně tak pro **s**. Pokud chceme dokument chápat jako bod v prostoru, je možno ztotožnit váhy jednotlivých klíčových slov dokumentu s eukleidovskými souřadnicemi bodu. Poloha takto určeného bodu v prostoru potom vlastně vyjadřuje výsledek indexování. V případě dokumentu **D** s klíčovými slovy U a V bude odpovídající bod [**r,s**] ležet ve čtverci s vrcholy [0,0], [0,1], [1,0] a [1,1], viz obr. 8.1. Váhu klíčového slova U vynášíme na osu x, váhu klíčového slova V na osu y.

Ukážeme úvahy, která vedou ke stanovení předpisu pro výpočet hodnoty složeného výroku $U\ OR\ V$. V případě booleovského modelu, kdy váhy jsou 0 nebo 1, odpovídá dokumentu některý z bodů [0,0], [0,1], [1,0] nebo [1,1]. Výrok $U\ OR\ V$ nabývá v booleovské modelu hodnotu 1 pro každý z těchto bodů krom [0,0]. Dá se říct, že je-li vzdálenost bodu odpovídajícího dokumentu (tedy některého z bodů [0,0], [0,1], [1,0] nebo [1,1]) od bodu [0,0] větší než 0, pak hodnota výroku $U\ OR\ V$ je 1, jinak je 0.

Všimněme si, že vzdálenost bodu $[1, 1]$ od bodu $[0, 0]$ je větší, než vzdálenost bodů $[0, 1]$ a $[1, 0]$ od bodu $[0, 0]$. Jestliže dokumentu odpovídá bod $[1, 1]$, pak obsahuje obě klíčová slova, U i V. Takový dokument je jistě lepší než dokument odpovídající některému z bodů $[0, 1]$ nebo $[1, 0]$, který obsahuje pouze jedno z obou klíčových slov. Toto pozorování lze zobecnit tak, že čím dál je bod odpovídající dokumentu od bodu [0,0], tím lépe dokument vyhovuje výroku $U\ OR\ V$. Je tedy přirozené stanovit hodnotu výroku $U\ OR\ V$ tak, aby byla přímo úměrná vzdálenosti bodu odpovídajícího dokumentu od bodu $[0, 0]$. Je však třeba dodržet konvenci, že váha každého výroku je z intevalu $< 0, 1 >$.

Maximální váhu bude mít dokument odpovídající bodu $[1, 1]$, tento dokument je z hlediska výroku $U\ OR\ V$ nejlepší. Hodnotu výroku $U\ OR\ V$ pro dokument D je tedy vhodné stanovit jako podíl vzdálenosti bodu $[r, s]$ od bodu $[0, 0]$ a vzdálenosti bodu $[1, 1]$ od bodu $[0, 0]$. Použitím Pythagorovy věty dostaneme vzorec

$$\text{hodnota}(U\ OR\ V) = \sqrt{\frac{r^2+s^2}{2}}$$



Obrázek 8.1: Geometrická představa dokumentu

váha		
U	V	hodnota U OR V
1.0	1.0	1.00 *
1.0	0.0	0.71 *
0.0	1.0	0.71 *
0.3	0.8	0.60
0.1	0.9	0.64
0.1	0.2	0.22
0.1	0.1	0.10
0.0	0.0	0.00 *

Tabulka 8.3: Příklady hodnot U OR V

V tab. 8.3 jsou příklady hodnoty $U\ OR\ V$ pro několik různých hodnot vah r a s.

Hvězdičkami jsou označeny případy, kdy výroky U a V nabývají booleovských hodnot 1 a 0. Všimněme si, že pro geometrické rozšíření nejsou hodnoty výroku U OR V stejné jako u booleovského modelu.

V případě výroku $U\ OR\ V$ jsme jeho hodnotu odvozovali od vzdálenosti bodu $[r, s]$ od bodu $[0, 0]$. Jedná se o jediný bod, pro který výrok $U\ OR\ V$ nabývá hodnotu 0, pro ostatní tři body je booleovská hodnota tohoto výroku 1. Booleovská hodnota 0 výroku $U\ OR\ V$ tedy jednoznačně určuje bod $[0, 0]$. Pro výrok $U\ AND\ V$ se takto chová bod $[1, 1]$, který je jediným bodem, pro který booleovský výraz $U\ AND\ V$ nabývá hodnotu 1. Je tedy možné vyjít při definici hodnoty výroku $U\ AND\ V$ ze vzdálenosti bodu $[r, s]$ od bodu $[1, 1]$. Je však nutno použít nepřímou úměrnost, neboť čím je vzdálenost bodu $[r, s]$ od bodu $[1, 1]$ menší, tím lepší je dokument vzhledem k výrazu $U\ AND\ V$. Na základě této úvahy dostaneme použitím Pythagorovy věty vzorec

$$\text{hodnota}(U\ AND\ V) = 1\text{-}\ \sqrt{\frac{(1-r)^2+(1-s)^2}{2}}$$

Následující tabulka 8.4 je doplněním tabulky 8.3 o hodnoty $U\ AND\ V$.

váha			
U	V	hodnota $U\ OR\ V$	hodnota $U\ AND\ V$
1.0	1.0	1.00	1.00 *
1.0	0.0	0.71	0.29 *
0.0	1.0	0.71	0.29 *
0.3	0.8	0.60	0.49
0.1	0.9	0.64	0.32
0.1	0.2	0.22	0.15
0.1	0.1	0.10	0.10
0.0	0.0	0.00	0.00 *

Tabulka 8.4: Příklady hodnot $U\ OR\ V$ a $U\ AND\ V$

Hvězdičkami jsou opět označeny případy, kdy výroky U a V nabývají booleovských hodnot 1 a 0. Všimněme si, že pro geometrické rozšíření nejsou v těchto případech hodnoty výroku $U\ AND\ V$ stejné jako u booleovského modelu. Dále je vhodné si všimnout, že hodnoty výroku $U\ AND\ V$ nejsou větší než hodnoty $U\ OR\ V$.

Hodnota $NOT\ U$ se definuje jako 1 - r.

Ze vzorce pro výpočet hodnoty $U\ AND\ V$ i z numerických příkladů je zřejmé, že takto je splněn požadavek číslo 4) na odstranění tvrdosti konjunkce stanovený v odstavci 8.1.

Výše uvedené vzorce lze rozšířit i na situaci, kdy budou použity váhy klíčových slov také v dotazu. Předpokládejme např. dotaz

$$[U, a]\ OR\ [V, b]$$

ve kterém má klíčové slovo U váhu a a klíčové slovo V váhu b. Jestliže vyhodnocujeme tento dotaz pro dokument ve kterém má klíčové slovo U váhu r a klíčové slovo V váhu s, použijeme vzorec

$$\text{hodnota}\ ([U, a]\ OR\ [V, b]) = \sqrt{\frac{(a*r)^2+(b*s)^2}{a^2+b^2}}$$

Pro dotaz

$$[U, a]\ AND\ [V, b]$$

použijeme vzorec

$$\text{hodnota } ([U, a]\ AND\ [V, b]) = 1 - \sqrt{\frac{a^2*(1-r)^2+b^2*(1-s)^2}{a^2+b^2}}$$

Pro dotaz $[U, a]\ AND\ NOT\ [V, b]$ použijeme vzorec

$$\text{hodnota } ([U, a]\ AND\ [V, b]) = (a * r) * (1 - b * s).$$

Tímto způsobem lze splnit všechny čtyři požadavky uvedené v odstavci 8.1.

Ukazuje se, že rozšíření booleovského modelu vykazují většinou lepší výsledky než standardní model.

8.5 Porovnání booleovského modelu a jeho rozšíření

V tabulce 8.5 je uvedeno srovnání výsledků dosažených pomocí booleovského modelu a jeho rozšíření. Tabulka je převzata ze [SALTON 89].

Byly porovnávány výsledky pro desítky dotazů ve čtyřech různých fondech obsahujících tisíce dokumentů. Hodnocena byla dosažená přesnost při konstantní úplnosti. V uvedeném pramenu není uvedeno, pro jakou hodnotu relevance dotazu a dokumentu byl dokument ještě považován za relevantní.

			přesnost pro konstantní úplnost		
fond	dokumentů	dotazů	booleovský model	fuzzy logika	geometrické rozšíření
CACM	3 204	52	0.1789	0.1551 (-14%)	0.3314 (+ 72%)
CISI	1 460	35	0.1118	0.1000 (-11%)	0.1806 (+ 62%)
INSPEC	12 684	77	0.1159	0.1314 (+13%)	0.2700 (+133%)
MED	1 033	30	0.2065	0.2368 (+15%)	0.5573 (+167%)

Tabulka 8.5: Srovnání booleovského modelu a jeho rozšíření

Z tabulky vyplývá, že použitím fuzzy logiky nemusí dojít ke zlepšení. Použitím geometrického rozšíření však lze zřejmě dosáhnout znatelně lepších výsledků. Přesto je třeba stále mít na zřeteli, že vypočítané hodnoty jsou jenom nedokonalým numerickým vyjádřením značně složitého pojmu "relevance dokumentu k informačnímu požadavku".

Kapitola 9

Další přístupy k vyhledávání textových dokumentů

9.1 Vektorový model

V tomto odstavci ukážeme princip vyhledávání dokumentů založený na chápání dokumentu jako vektoru. Opět předpokládáme, že obsah dokumentu je popsán klíčovými slovy, každému klíčovému slovu je přiřazena váha vyjadřující důležitost slova pro charakteristiku obsahu dokumentu. Předpokládáme i váhy klíčových slov v dotazu. Základní ideou vektorového přístupu je vyjádřit každý dokument i dotaz jako vektory, jejichž složky jsou váhy jednotlivých klíčových slov a využít prostředků vektorového počtu k výpočtu podobnosti dotazu a dokumentu.

V případě geometrického rozšíření chápeme dokument D jako bod P_D v prostoru. Souřadnice bodu P_D jsou dány vahami jednotlivých klíčových slov u dokumentu D. To je prakticky stejný přístup jako u vektorového modelu, bod P_D jednoznačně odpovídá vektoru s počátečním bodem o souřadnicích $< 0, 0, ..., 0 >$ a koncovým bodem P_D. Rozdíl je ve výpočtu podobnosti dotazu a dokumentu. Prostředky které vektorový počet používá pro výpočet podobnosti vektorů neumožňují vyjádřit vztahy dané logickými spojkami AND, OR a NOT, které jsou k dispozici v geometrickém rozšíření booleovského modelu. Na druhé straně je při vektorovém přístupu možno snadno průběžně modifikovat dotaz na základě již vyhledaných dokumentů.

Ukážeme několik možností, jak spočítat podobnost dotazu a dokumentu a naznačíme způsob modifikace dotazu na základě již vyhledaných dokumentů.

Předpokládejme, stejně jako v kapitole 5 o automatickém indexování, že máme informační fond s N dokumenty $D_1, D_2, ...D_N$, k jejichž indexování byla použita klíčová slova $S_1, S_2, ..., S_K$. Váhu slova S_j v dokumentu D_i značíme $w_{i,j}$. K dokumentu D_i je tedy přiřazen vektor $< w_{i,1}, w_{i,2}, ..., w_{i,K} >$.

Zde je třeba upozornit, že z důvodů formálního vyjádření se uvádí váha pro každé ze slov $S_1, S_2, ..., S_K$, včetně nulových vah. Pokud váha některého slova pro dokument je nulová, znamená to, že toto slovo nepatří mezi klíčová slova dokumentu, ani trochu necharakterizuje jeho obsah. Například, je-li $W_{2,1} = 0$, znamená to, že slovo S_1 není klíčovým slovem pro dokument D_2. Fakt, že slovo S_1 není klíčovým slovem pro dokument D_2 je informace, která je využívána při výpočtu podobnosti dokumentu a dotazu. Pokud bychom u dokumentu uváděli jenom jeho klíčová slova s nenulovými vahami, narazili bychom na formální potíže při vyjadřování vektorových operací.

Jak už bylo uvedeno, jako vektor vah pro klíčová slova vyjádříme nejen každý dokument, ale i dotaz. Formálně budeme pro dotaz Q psát $Q =< q_1, q_2, ..., q_K >$. Čísla $q_1, q_2, ..., q_K$ patří

do intervalu $< 0, 1 >$. Uvedený zápis znamená, že váha klíčového slova S_1 v dotazu Q je q_1, váha klíčového slova S_2 je q_2, atd.

Při vyhledávání pomocí vektorového modelu se pro každý dokument spočítá jeho podobnost s dotazem. Výsledkem vyhledávání je seznam dokumentů seřazený podle stupně podobnosti. Pro výpočet se používají obecné míry podobnosti vektorů. Takových měr existuje celá řada, podrobněji se jimi zabývá např. [STROSSA 94]. Uvedeme dva příklady takových měr, kosinovou míru podobnosti a Diceovu míru podobnosti. Následující vzorce platí pro dokument

$$D =< w_1, w_2, ..., w_K >$$

a dotaz

$$Q =< q_1, q_2, ..., q_K > .$$

Kosinová míra podobnosti se počítá jako

$$\frac{\sum_{i=1}^{K} w_i * q_i}{\sqrt{\sum_{i=1}^{K} w_i^2 + \sum_{i=1}^{K} q_i^2}}$$

Diceova míra podobnosti se počítá jako

$$\frac{2 * \sum_{i=1}^{K} w_i * q_i}{\sum_{i=1}^{K} w_i^2 + \sum_{i=1}^{K} q_i^2}$$

Výhodou vektorového přístupu je relativně snadná možnost iterativní modifikace dotazu na základě již vybraných relevantních dokumentů. Modifikace vychází z předpokladu, že jestliže jsou dokumenty relevantní k dotazu, pak si jsou jejich vektory vzájemně podobné. Jestliže v průběhu vyhledávání získáme několik relevantních dokumentů, je možno automaticky modifikovat vektor dotazu tak, aby se co nejvíce podobal vektorům relevantních dokumentů. Použití přeformulovaného dotazu dává šanci na získání dalších relevantních dokumentů. Metodám modifikace se věnuje [SALTON 89].

Mezi nevýhody vektorového přístupu patří chybějící teoretické zdůvodnění výběru vhodné míry podobnosti. Nevýhody plynou i z toho, že jednotlivá klíčová slova nejsou nezávislá. Jestliže se například ve fondu technické literatury vyskytuje u dokumentu jako klíčové slovo termín "bezpečnostní pasy" tak se zvyšuje šance, že se bude vyskytovat i klíčové slovo "pasivní bezpečnost". Z toho vyplývá, že příslušné složky vektoru dokumentu nebudou ortogonální, což bude komplikovat vektorové operace.

9.2 Automatická klasifikace

Automatická klasifikace dokumentů je další přístup, jak využít počítače při vyhledávání dokumentů. Připomeňme systematické selekční jazyky, kterým je věnován odstavec 4.3. Jejich princip spočívá v pokrytí celé oblasti vědění, které se popisované dokumenty týkají hierarchicky uspořádaným systémem tříd. Znaky systematického selekčního jazyka je možno využít při formulaci podmínky na požadované dokumenty. V knihovnách s přímým přístupem bývají knihy patřící do jedné třídy (a tedy označené stejným klasifikačním znakem) umístěny vedle sebe, což usnadňuje výběr relevantních knih.

Cílem automatické klasifikace je automaticky rozdělit dokumenty do tříd vzájemně podobných dokumentů tak, aby bylo možno:

- k nalezenému relevantnímu dokumentu operací typu browse prohlížet podobné dokumenty,

- využít třídy k vyhledávání podle podobnosti dokumentů.

Pro automatickou klasifikaci se využívá shlukové analýzy (cluster analysis) [LUKASOVÁ 85]. Je to disciplína matematické statistiky pracující s obecnými objekty, její metody lze aplikovat i na dokumenty. Místo o rozdělení dokumentů do tříd se proto většinou hovoří o rozdělení do shluků nebo clusterů.

Obecně se metody vytváření shluků dělí na divizivní a aglomerativní. Divizivní metody pracují tak, že na počátku tvoří všechny objekty jeden shluk, který se postupně dělí na menší shluky. Při použití aglomerativní metody tvoří na počátku každý objekt jeden shluk. Jednotlivé shluky se postupně sdružují do větších shluků.

Všechny metody vycházejí z matice párových podobností objektů, v našem případě tedy dokumentů. Pokud je ve fondu N dokumentů D_1, D_2,... D_N, pak jejich matice párových podobností bude mít tvar dle obr. 9.1.

$$\begin{array}{ccccc} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,N} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots & p_{2,N} \\ p_{3,1} & p_{3,2} & p_{3,3} & \cdots & p_{3,N} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ p_{N,1} & p_{N,2} & p_{N,3} & \cdots & p_{N,N} \end{array}$$

Obrázek 9.1: Matice podobnosti dokumentů $D_1, D_2, ...D_N$

Prvky matice jsou čísla $p_{i,j}$ pro $i, j = 1, ..., N$. Číslo $p_{i,j}$ udává podobnost dokumentů D_i a D_j. Lze je spočítat například jako kosinovou míru podobnosti.

Stručně ukážeme příklad aglomerativní metody. Její algoritmus lze popsat takto:

1. Spočti matici koeficientů párových podobností.
2. Z každého dokumentu vytvoř samostatný shluk.
3. Vytvoř nový shluk ze dvou nejpodobnějších shluků i a j. Z matice podobností vynech řádky a sloupce shluků i a j. Do matice podobností doplň řádek a sloupec pro nový shluk.
4. Krok 3 opakuj tak dlouho, pokud existují dva shluky, které lze sloučit.

Pokud jsou oba shluky jednoduché dokumenty, pak je jejich podobnost dána příslušným prvkem matice párových podobností dokumentů. Pokud alespoň jeden ze shluků obsahuje více dokumentů, pak je možno pro výpočet jejich podobnosti použít např. **metodu nejbližšího souseda**. To znamená, že se vytvoří všechny dvojice dokumentů takové, že první dokument je z prvního shluku a druhý dokument z druhého shluku. Ze všech dvojic se vybere dvojice ve které si jsou dokumenty nejpodobnější a podobnost těchto dvou dokumentů se prohlásí za podobnost celých shluků. Analogicky pracuje metoda nejvzdálenějšího souseda, případně metoda průměru shluku.

Při vyhledávání dokumentů je možno využít reprezentaci shluků pomocí centroidů. Pokud dokumentům odpovídají vektory, jejichž složky jsou váhy jednotlivých klíčových slov, pak příkladem centroidu může být např. vektor, jehož i-tá složka je průměrem i-tých složek všech vektorů shluku.

Vyhledávání dokumentů může probíhat např. podle těchto zásad:

- shluky jsou reprezentovány pomocí centroidů,
- hledá se zadaný počet dokumentů,
- vyhledávají se shluky s centroidem co nejpodobnějším dotazu, dokumenty z nich se zařazují do výstupu tak dlouho, až je jich získán zadaný počet,
- z vyhledaného shluku je případně možno vybrat pouze nejpodobnější dokumenty (pracnější, ale lepší výsledky).

Podrobnější a přesnější popis automatické klasifikace dokumentů a použití výsledků při vyhledávání dokumentů přesahuje rozsah těchto skript. Lze jej nalézt např. v [STROSSA 94].

Kapitola 10

Pojmové vyhledávání

10.1 Topic

Pojmové vyhledávání je metoda vyhledávání ve fondech textových dokumentů. Základní myšlenkou je definovat pojem, který nás zajímá, nezávisle na momentálním stavu fondu. Definovaný pojem se může opakovaně používat při vyhledávání relevantních dokumentů mezi nově získanými dokumenty. Na základě definice pojmu a plného textu dokumentu lze stanovit stupeň relevance dokumentu k pojmu. Na výstupu jsou jednotlivé dokumenty řazeny podle stupně relevance.

Již definovaný pojem může být využit i při definici nových pojmů. Například pojmy *sjezd, slalom speciál, obří slalom* a *superobří slalom* mohou být použity při definici pojmu alpské lyžování. Při definici se využívá i hierarchických vztahů mezi pojmy. *Alpské lyžování* je obecnější pojem, než ostatní uvedené, stojí tedy hierarchicky výš.

Pro definici pojmu slouží topik (z anglického topic). Je to formální prostředek pro vyjádření hierarchické struktury definovaného pojmu, důležitosti jeho jednotlivých částí a způsobu, jak se z jednotlivých částí skládá nový pojem. Podrobněji je definici pojmu věnován odstavec 10.2.

Asi by bylo možno diskutovat o tom, zda lze považovat za definici pojmu vektorový dotaz popsaný v předchozí kapitole nebo případně i klasický booleovský dotaz. Prostředky topiku jsou však silnější než booleovský nebo vektorový dotaz. Booleovský dotaz lze pomocí topiku vyjádřit, hierarchické vztahy pojmů lze vyjádřit pouze prostředky topiku. To je hlavní důvod, pro který nebudeme booleovské ani vektorové dotazy považovat za pojmové vyhledávání.

Implementací pojmového vyhledávání pomocí topiků je softwarový systém TOPIC stručně zmíněný v odstavci 7.2. Zde pouze naznačíme bohaté prostředky pro práci s jednotlivými topiky které systém TOPIC poskytuje. Topik je možno nejen snadno definovat, uchovávat nebo využívat v jiném topiku, ale i sdílet různými uživateli. To je mimo jiné další důvod, proč se práce s topikem podobá práci s pojmy více než práce s vektorovými nebo booleovskými dotazy.

10.2 Definice pojmu pomocí topiku

Definici pojmu pomocí topiku naznačíme na příkladu. Již na první pohled zjednodušený příklad definice pojmu je v obr. 10.1.

Základem definice je stromová struktura (the topic tree). Kořen stromu odpovídá definovanému pojmu (the root topic), v obr. 10.1 to je *Praha*. Větve stromu reprezentují dílčí pojmy, z

bude doplněno, jíné příklady topiků jsou k dispozici na slidech dostupných na systému ISIS

Obrázek 10.1: Příklad topiku

nichž se definovaný pojem skládá (branch topic). V obr. 10.1 to jsou *Hradčany* a *Vyšehrad*. V obecném případě mohou být větve ve více úrovních. Listy stromu jsou termíny, jejichž výskyt v textu dokumentu zvyšuje relevantnost dokumentu k definovanému pojmu. V obr. 10.1 to jsou *Vikárka, Daliborka, Loreta, Slavín* a *Šemík*.

Součástí definice dále jsou

- listové operátory, v obr. 10.1 to je STEM,
- modifikátory, v obr. 10.1 to je MANY,
- váhy, v obr. 10.1 např. 0.5. ,
- pojmové operátory, v obr. 10.1 to je ACCRUE,

Listové operátory určují způsob, jakým se odvozují v textu dokumentu vyhledávané výrazy od termínu uvedeného v listu dotazu. Zápis <STEM> *Vikárka* znamená, že se mají vyhledávat všechny výrazy, které mají stejný slovní kmen jako *Vikárka*. Jedná se o implementaci stemování popsaného v odstavci 6.5. Mezi listové operátory dále patří například WILDCARD (realizuje rozšiřování slov popsané v odstavci 6.4), SOUNDEX (implementuje vyhledávání podobně znějících slov, viz odstavec 6.8) nebo THESAURUS (vyhledává i synonyma získaná z tezauru).

Modifikátory slouží pro modifikaci chování některých operátorů. Pokud použijeme pouze <STEM> *Vikárka*, bude výsledkem 0 nebo 1 v závislosti na tom, zda v textu byl či nebyl nalezen některý z takto zadaných výrazů. Použití <MANY><STEM> *Vikárka* způsobí, že výsledkem bude číslo z intervalu $< 0,1 >$ v závislosti na počtu výskytů výrazů zadaných <STEM> *Vikárka*. Dalším modifikátorem je CASE, který znamená že při vyhledávání jsou rozlišována velká a malá písmena. Poslední modifikátor, NOT se chová jako booleovská spojka *and not*.

Váhy umožňují vyjádřit důležitost výrazů uvedených v listech nebo důležitost dílčích pojmů reprezentovaných větvemi stromu.

Pojmové operátory jsou tři: OR, AND a ACCRUE. Používají se v uzlech stromu. Pro každý uzel určují způsob definice pojmu odpovídajícího uzlu pomocí dílčích pojmů reprezentovaných bezprostředně podřízenými uzly. Operátorem je určen způsob výpočtu váhy pojmu. Operátory OR a AND pracují na principu fuzzy logiky. Operátor ACCRUE pracuje na principu "čím více, tím lépe". V dostupné literatuře nejsou uvedeny podrobnosti o způsoby výpočtu váhy pro operátor ACCRUE.

Je třeba zdůraznit, že se jedná o pouhý příklad definice pojmu pomocí topiku. Systém TOPIC obsahuje krom pojmového vyhledávání i další možnosti vyhledávání dokumentů. Jedná se o prostý booleovský dotaz a o formulářový dotaz vycházející ze struktury dokumentu. Je možno pracovat také s dalšími operátory a prostředky než zde bylo uvedeno. Podrobnosti lze nalézt v uživatelské dokumentaci [TOVEK].

Uvedený příklad však nenaznačuje jenom možnosti definice pojmu pomocí topiku. Vyplývá z něho i fakt, že v takové definici je uloženo hodně podrobných znalostí oblasti popisované dokumenty se kterými pracujeme. Konstruováním topiků se často zabývají specialisté. Obvykle se topik ladí podobně jako se ladí programy. Předmětem ladění může být změna vah, přidání nebo rekonstrukce větví nebo listů. Systém TOPIC umožňuje udržovat knihovnu topiků, která představuje cenný zdroj znalostí.

Kapitola 11

Umělá inteligence

11.1 Obrysy pojmu

Umělá inteligence je obor, který v několika málo posledních desetiletích proniká do stále dalších oblastí lidské činnosti. To je mimo jiné jedním z důvodů, proč je obtížné umělou inteligenci nejen přesně definovat ale i pouze přesněji vymezit. Mezi další důvody jistě patří i obtížná definice inteligence přirozené chápané jako inteligence živých organismů. Existují stovky pokusů o definici umělé inteligence [MAŘÍK 93], zcela jednoznačně nelze určit ani její historický počátek.

Cílem těchto skript tedy nemůže být přesná definice umělé inteligence ani její podrobnější popis. Jde pouze o:

- hrubé vymezení pojmu umělé inteligence, které je cílem jak této, tak i následující kapitoly,
- stručnou charakteristiku expertních systémů, které jsou jedním z důležitých výsledků umělé inteligence, viz odstavec 11.2,
- nastínění hlavních rysů dalších vybraných disciplin a aplikačních oblastí umělé inteligence, je v odstavci 11.3,
- několik poznámek k některým teoretickým disciplinám patřících k základům umělé inteligence, je jim věnován odstavec 11.4.

Rozsáhlejší příklad využití umělé inteligence při vyhledávání textových dokumentů je v kapitole 12. Místo výrazu "umělá inteligence" budeme v dalším používat obecně přijatou zkratku AI (z anglického Artificial Intelligence).

Následující tři definice umělé inteligence, Minského definice, definice Richové a Kotkova definice jsou i s komentářem k nim převzaty z [MAŘÍK 93].

Minského definice zní takto: *Umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který - kdyby ho dělal člověk - bychom považovali za projev jeho inteligence.* [MINSKY 67]. Otázka je, jaký postup vykonávaný člověkem již lze považovat za projev inteligence. Lze k ní přistoupit tak, že za projev inteligence budeme považovat postup, při kterém jsou z možných variant vybírány pouze takové, které mají naději na úspěch, nejedná se tedy o mechanické testování jedné varianty za druhou. Čím více variant je oprávněně vynecháno, tím lze použitý postup označit za inteligentnější. Vynechávání beznadějných variant je umožněno využíváním znalostí.

Stroj nebo systém může znalosti získat od člověka nebo je sám odvodit z příkladů úloh a jejich inteligentních řešení. Může se jednat o exaktní znalosti např. ve formě matematických vět nebo fyzikálních zákonů. Mohou to být ale i heuristické poznatky nepodložené přesnou teorií ale získané zkušeností, které často efektivně vedou k řešení.

Definice Richové říká: "*Umělá inteligence se zabývá tím, jak počítačově řešit úlohy, které dnes zvládají lidé lépe.*" [RICH 91]. Výhodou této definice je, že stručně a poměrně přesně vymezuje skutečný obsah AI. Vyhýbá se také filosofickým úvahám zahrnovaným do většiny jiných definic. Za nevýhodu lze však považovat to, že s rozvojem aplikací počítačů se bude obsah pojmu AI měnit.

Kotkova definice vychází z chápání AI jako vlastnosti umělých systémů: "*Umělá inteligence je vlastnost člověkem uměle vytvořených systémů vyznačujících se schopností rozpoznávat předměty, jevy a situace, analyzovat vztahy mezi nimi a tak vytvářet vnitřní modely světa, ve kterých tyto systémy existují, a na tomto základě pak přijímat účelná rozhodnutí, za pomoci schopností předvídat důsledky těchto rozhodnutí a objevovat nové zákonitosti mezi různými modely anebo jejich skupinami.*" [KOTEK 93]. Na této definici je podstatné zdůraznění vnitřních modelů světa, které zahrnují jak modely prostředí tak modely akcí v něm prováděných. Použití počítačů nutně vyžaduje formální modely. Nalezení vhodných formálních modelů pro tyto účely je základem úspěšnosti celého systému.

Znovu je třeba zdůraznit, že se jedná jen o tři z několika stovek definic AI, všímajících si mimo jiné i filosofických a psychologických stránek věci. Některé ze souvisejících problémů jsou naznačeny např. v [GRUSKA 83].

Počátek AI je často spojovám s pracovní konferencí s názvem "The Dartmouth Summer Research Project on Artificial Intelligence", která se konala v létě 1956 v Dartmouth College v USA. Zúčastnilo se jí asi deset odborníků z matematiky, elektrotechniky, lingvistiky, neurologie a psychologie. Jejich úkolem bylo diskutovat o myšlence, že "každé hledisko učení nebo jakýkoliv jiný příznak inteligence může být v principu tak přesně popsán, že může být vyvinut stroj, který ho simuluje".

Důraz byl kladen na myšlenku, že počítač by mohl pracovat se symboly a ne pouze s čísly, jak bylo tehdy obvyklé. Byla vymezena oblast společného zájmu a tím i náplň budoucí vědní disciplíny, pro kterou byl již tehdy použit název *umělá inteligence*. Výsledkem konference byla i předpověď rozvoje umělé inteligence, ve které se mimo jiné uvádělo, že v roce 1970 bude počítač velmistrem v šachu, odhalí nové matematické věty a porozumí přirozenému jazyku tak, že bude moci pracovat jako překladatel.

Nesplnění této předpovědi spolu s naopak poměrně malými výsledky rozsáhlého bádání v AI vedly ke krizi umělé inteligence, která trvala zhruba do poloviny sedmdesátých let. Mimo jiné se ukázalo, že usilovně hledané univerzální systémy schopné řešit problémy bez znalostí jejich specifik jsou slabé pro řešení specializovaných úloh efektivně řešitelných experty majícími specielní znalosti v příslušných oblastech. Ukázalo se, že podstatná je práce se znalostmi, jejich reprezentace a využívání. V této době začaly vznikat expertní systémy, tedy softwarové systémy schopné uchovávat znalosti lidských expertů a využívat je při řešení problémů. Expertním systémům je poněkud podrobněji věnován odstavec 11.2.

Velký význam pro rozvoj umělé inteligence měl a stále má japonský projekt počítačů páté generace vyhlášený v roce 1981 [MAŘÍK 93]. Jeho základní témata se týkala řešení úloh a odvozování, bází znalostí a komunikace s uživatelem. V rámci projektu bylo dosaženo významných výsledků, i když se celkové cíle ukázaly prozatím jako příliš vysoké. V USA se podobným projektem zabývá firma MCC (Microelectronics and Computer Technology Corporation). Evropskou odpovědí na japonský projekt je ESPRIT (European Strategic Program for Research in Information Technology), který trvá od roku 1983 dosud.

Přesto, že koncem osmdesátých let došlo k jistému období skepse, je v současné době patrný trend k široké aplikaci metod a přístupů AI v různých softwarových systémech. Za základní pro-

blém je považována "*integrace dílčích komponent do rozsáhlých celků vykazujících kvalitativně nové chování*" [ROTA 92], [MAŘÍK 93].

Za počátek AI je sice považována Dartmouthská konference v roce 1956, existuje však řada prací, které se různým způsobem dotýkaly problémů později přímo do AI zahrnovaných nebo s ní více či méně souvisících. Je sem zařazována např. práce Leibnize "De Arte Combinatorica" z roku 1666, kterou charakterizoval jako "*všeobecnou metodu, kterou by se dala všechna pravda rozumu redukovat na jistý druh výpočtů*" [GRUSKA 83]. Patří sem i další teoretické práce a problémy zmíněné v odstavci 11.4.

11.2 Expertní systémy

Prakticky významným a široce rozšířeným výsledkem AI jsou expertní systémy. Jejich účel a principy naznačíme na příkladu. Týká se problematiky zpracování dat o spolehlivosti automobilů pro potřeby řízení jakosti [RAUCH 86]. Jedná se o dílčí problém zpracování dat o poruchách v záruční době. Tato data obvykle zahrnují podrobnou identifikaci automobilu a dílu u kterého k poruše došlo včetně příčiny nebo alespoň projevu poruchy. V tabulce 11.1 jsou příklady zaznamenávaných údajů pro několik poruch.

díl	projev poruchy	ujeto km
těsnění kroužek	prasklý	506
těsnění kroužek	netěsní	570
ložisko	hřeje	1250
ložisko	nemazané	89

Tabulka 11.1: Příklady údajů o poruchách

Obvykle jsou k dispozici data pro všechny poruchy v záruční době, již při roční produkci řádově desítek tisíc automobilů se může jednat o stovky tisíc poruch ročně. V datech takového rozsahu je skryto mnoho důležitých faktů o kvalitě konstrukce i výrobního procesu. Hodně lze usoudit např. z odchylek skutečného počtu poruch od očekávaného stavu. V ideálním případě by sice měl být očekávaný počet poruch nulový, z různých příčin tomu tak však není.

Může tedy být například očekáváno, že těsnící kroužky budou praskat u 0.2 % automobilů v záruční době. Jestliže je skutečná četnost prasklých těsnících kroužků větší než očekávaná, pak to bude s vysokou pravděpodobností znamenat chybnou montáž, případně s menší pravděpodobností nízkou pevnost materiálu. Podobně, vyšší netěsnost kroužku než očekávaná může být s vysokou pravděpodobností způsobena špatným povrchovým opracováním a s menší pravděpodobností chybnou montáží. Skutečnou příčinu lze zjistit až podrobnějším technickým rozborem, pravděpodobné příčiny lze však zjistit analýzou dat o poruchách s využitím znalostí a zkušeností z výroby, provozu a oprav automobilů.

Vzhledem k počtu kombinací dílů a potencionálních příčin je však značně náročné provádět pravidelně prvotní odhady potencionálních příčin poruch. Proto byl navržen a také experimentálně provozován expertní systém ARES [IVÁNEK 85A]. Jeho úkolem bylo rutinně provádět tyto analýzy a vhodným způsobem při tom využít reprezentované znalosti expertů.

Expertní znalosti byly reprezentovány pravidly typu

JESTLIŽE ... POTOM ... S VAHOU ... .

vyjadřujícími názor experta na příčiny rozdílů mezi skutečným a předpokládaným počtem poruch.

Příkladem jsou pravidla

JESTLIŽE	*poruchy typu "prasklý" mají větší než očekávanou četnost,*
POTOM	*potenciální příčinou poruch je chybná montáž*
S VAHOU	*0.7*

JESTLIŽE	*poruchy typu "prasklý" mají větší než očekávanou četnost,*
POTOM	*potenciální příčinou poruch je poddimenzování*
S VAHOU	*0.2*

JESTLIŽE	*poruchy typu "zadřený" mají větší než očekávanou četnost,*
POTOM	*potenciální příčinou poruch je chybná montáž*
S VAHOU	*0.2*

Váha nabývá hodnot z intervalu $< 0, 1 >$ a vyjadřuje stupeň jistoty, který expert spojuje závěr pravidla s jeho předpokladem.

V ideálním případě by systém ARES měl mít k dispozici pravidla pro každý díl a každý projev poruchy. Vzhledem ke neúnosnému rozsahu takového souboru pravidel a k vzájemné podobnosti mnoha dílů se většina pravidel vztahuje k technologickým třídám. Příkladem technologické třídy jsou "valivá ložiska" nebo "gumová těsnění". Menší část pravidel se vztahuje k jednotlivým dílům, některá ke všem dílům. Znalosti o struktuře pravidel tvoří jisté *metaznalosti* systému, které dohromady s pravidly tvoří *bázi znalostí*.

Řídící mechanismus organizuje průběh práce systému. Nejprve zkontroluje, zda hodnocený díl měl vůbec významný počet poruch. Jestliže ano, spočítá frekvence jednotlivých projevů poruchy. Dále shromáždí všechna pravidla relevantní k tomuto dílu, t.j. obecná pravidla, pravidla vztahující se k technologické třídě do níž díl patří a případně pravidla pro tento díl specifická. Poté, na základě váhy pravidla a velikosti odchylky skutečné frekvence projevu poruchy od frekvence očekávané, se určí příspěvek pravidla a přičte se k celkovému skore potenciální příčiny uvedené na pravé straně pravidla. Připomínáme, že jedna potenciální příčina může být uvedena na pravé straně několika pravidel. Její celkové skore je tedy součtem příspěvků všech pravidel, na jejichž pravé straně je uvedena. Po vyčerpání všech projevů poruchy seřadí řídící mechanismus jednotlivé potenciální poruchy podle jejich skore a vydá ty, které získaly skore větší než danou mez.

Systém ARES je typický expertní systém v tom smyslu, že odděluje znalosti od dat. Znalosti jsou zde v pravidlech a v metaznalostech, data jsou údaje o počtech jednotlivých projevů poruch. Typické je i oddělení řídícího mechanismus od dat i od znalostí. Lze tedy opakovaně použít stejné znalosti na různá data, při přidání nebo odstranění pravidel zůstává stále stejný řídící mechanismus. Pravidla modelují znalosti expertů, řídící mechanismus nahrazuje způsob odvozování závěrů. Proto se místo řídící mechanismus v souvislosti s expertními systémy používá termín "inferenční (odvozovací) mechanismus".

I když pravidla použitá v systému ARES i jeho inferenční mechanismus jsou velmi pravděpodobně od reality značně vzdálené modely znalostí a odvozování závěrů, přesto ze zkušebního provozu systému ARES vyplynulo že je schopen činit závěry, které jsou v dobré shodě se závěry expertů.

Systém ARES však není typický expertní systém co se týče řetězení pravidel. V tomto systému jsou pouze pravidla, na jejichž levé straně je předpoklad o odchylce skutečné frekvence od předpokládané, na pravé straně pak potencionální příčina poruchy. Nemůže tedy dojít k situaci, kdy by výsledek použití jednoho pravidla byl předpokladem jiného pravidla, což je pro expertní systémy typické.

Je však možno uvažovat o přirozeném rozšíření systému ARES tak, aby docházelo k řetězení pravidel. Lze to udělat například na základě požadavku, aby systém navrhoval nápravná opatření. Toho je možno docílit přidáním pravidel jejichž několik příkladů uvádíme:

JESTLIŽE *potenciální příčinou poruch je chybná montáž,*
POTOM *zvýšit mezioperační kontroly*
S VAHOU *0.7*

JESTLIŽE *potenciální příčinou poruch je poddimenzování,*
POTOM *použít kvalitnější materiál*
S VAHOU *0.3*

JESTLIŽE *potenciální příčinou poruch je poddimenzování,*
POTOM *zvětšit rozměry*
S VAHOU *0.5*

Je zřejmé, že tato pravidla by se aplikovala na výsledky dosud používaných pravidel, docházelo by k jejich řetězení.

Podrobnosti o řetězení pravidel a dalších možnostech expertních systémů, o jejich aplikacích, problémech, perspektivách a dalších souvislostech existuje rozsáhlá literatura, např. [HÁJEK 85], [BERKA 94], [MAŘÍK 96]. Za zmínku stojí rozsáhlé aplikace expertních systémů v ekonomii, například ve finančnictví, kterým je mimo jiné věnován samostatný časopis "INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS IN ACCOUNTING FINANCE AND MANAGEMENT".

11.3 Vybrané partie a oblasti aplikací

Cílem tohoto odstavce je nastínění stručné charakteristiky několika málo vybraných partií a aplikačních oblastí AI. Všimneme si:

- řešení úloh ve stavovém prostoru,
- strojového učení,
- neuronových sítí,
- robotiky,
- porozumění přirozenému jazyku,
- znalostního inženýrství.

Je třeba zdůraznit, že jednotlivé charakteristiky jsou určeny pouze pro hrubé seznámení se s nejdůležitějšími cíli a principy, v žádném případě se nejedná o přesný a úplný popis.

Řešení úloh ve stavovém prostoru vychází z modelování reálného prostředí pomocí možných stavů. Příkladem může být šachová partie, na jejímž počátku je jeden konkrétní stav daný počátečním rozestavením figur. Z počátečního rozestavení lze prvním tahem přejít do některého z množiny přípustných rozestavení, která je dána pravidly šachové hry. Stejně tak lze pro každé přípustné rozestavení určit následující přípustná rozestavení. Každé přípustné rozestavení figur na šachovnici lze chápat jako možný stav modelovaného prostředí. Konec šachové partie nastává při výhře bílého, výhře černého nebo při patu. Pro každé přípustné rozestavení figur lze poznat zda nastal konec partie.

Jestliže řešíme úlohu vyhrát partii bílými figurami, můžeme celou situaci chápat tak, že máme dán počáteční stav, množinu stavů provázaných přípustnými přechody z jednoho stavu do druhého a předpis, jak určit že daný stav odpovídá úspěšnému vyřešení úlohy. Takový stav se nazývá koncovýn stavem. Naší úlohou tedy je nalézt cestu z počátečního do koncového stavu.

Stejným pohledem se lze dívat na řadu dalších úloh. Množina všech přípustných stavů se obvykle nazývá stavový prostor. Reprezentuje se orientovaným grafem, jehož uzly odpovídají

přípustným stavům a orientované hrany (spojnice mezi uzly) odpovídají přípustným přechodům mezi stavy. Danou úlohu lze takto převést na úlohu nalezení cesty orientovaným grafem z počátečního do koncového uzlu. Efektivnost řešení dané úlohy tedy závisí na efektivnosti algoritmů pro prohledávání grafů s případným využitím další informace. Řešení úloh ve stavovém prostoru má řadu aplikací i rozsáhlé teoretické zázemí a souvislosti [MAŘÍK 93].

Strojové učení si klade za cíl odvodit ze zadaných příkladů a protipříkladů podstatné a charakteristické vlastnosti určující daný pojem. Učícímu se systému jsou předloženy informace týkající se jednotlivých objektů charakterizovaných množinou atributů. Pro každý objekt jsou dány hodnoty jeho atributů spolu s údajem zda objekt je či není instancí pojmu. Systém má za úkol nalézt takovou podmnožinu atributů, na jejichž základě by mohl pro další předkládané objekty určovat zda jsou či nejsou instancemi daného pojmu.

Na strojové učení se lze dívat jako na prohledávání stavového prostoru všech možných popisů daného pojmu. Používá se zejména při získávání znalostí pro expertní systémy, pracuje se i na aplikacích v automatizované tvorbě software [MAŘÍK 93].

Stručnou charakteristiku **neuronových sítí** uvedeme citátem použitým za podobným účelem v [HAVEL 88]. Jedná se o motto z historické knihy A. T. Macrobiuse o rituálním tanci: "*Hlasy jednotlivců jsou skryty v chóru, zatímco hlas všech jasně zní. Vyvstává harmonie ...*". Podstatný je zde fakt emergence - vyvstávání, který se týká chování rozsáhlého systému aktivních prvků. V [HAVEL 88] jsou jako další příklady uváděny roj včel, tah mravenců nebo magnetizace kovové tyče.

Neuronové sítě jsou sítě mnoha jednoduchých vzájemně propojených procesorů. Vzhledem k schopnosti takových procesorů modelovat, byť zjednodušeně, neurony centrální nervové soustavy, nazývají se tyto procesory také neurony. Jednotlivé neurony mají několik vstupů a jeden výstup. Vstupy přicházejí z jiných neuronů nebo z vnějšího prostředí, výstup vede do jiného neuronu nebo může být výstupem celé sítě. Vstupy a výstupy neuronu jsou reálná čísla, jejich vzájemná závislost je vlastností neuronu. Jednotlivé sítě mohou být různě geometricky uspořádány, nejdůležitější jsou vrstvené sítě [MAŘÍK 93]. Neurony první vrstvy dostávají signály zvenčí, zpracují je a předají neuronům druhé vrstvy, atd. Výstupy poslední vrstvy lze považovat za výstup celé sítě. Vstupem i výstupem jsou tedy vektory čísel, hodnoty jejich složek nesou informaci.

Podrobnější popis vlastností neuronových sítí pochopitelně přesahuje rozsah těchto skript. Omezíme se na uvedení dvou příkladů oblastí jejich použití. Patří sem např. klasifikace EKG na dvě třídy - normální nebo abnormální. Dalším příkladem je použití v řízení technologických procesů, kdy člověk - expert vydává řídící příkazy na základě znalostí stavu řízeného systému a dlouholeté zkušenosti. Neuronová síť je schopna po jisté době "odpozorovat" chování experta a kopírovat jej. Není přitom nutné, aby expert formuloval svoje zkušenosti např. ve formě pravidel jak to vyžadují expertní systémy.

Robotika vždy měla a stále má velmi úzký vztah k AI. V počátcích AI byla považována za její součást. V dnešní době je robotika samostatnou rozsáhlou vědeckou a technickou disciplinou ve které jsou široce aplikovány metody AI. Existence robotiky by bez využití výsledků AI nebyla možná. Na druhé straně je robotika zdrojem inspirace pro AI. Významnou českou monografií o robotice je [HAVEL 80]. V současnosti je robotika chápána jako "inteligentní vazba od vnímání k akci" [MAŘÍK 93], [BRADY 85]. V rámci robotiky vznikly relativně samostatné disciplíny, například počítačové vidění [HLAVÁČ 92], automatické rozpoznávání nebo zpracování dotykové informace. Za zmínku stojí i fakt, že v současné době prudce se rozvíjející systémy virtuální reality jsou považovány za extrapolaci robotiky.

Porozumění přirozenému jazyku je úloha s mnoha potenciálními významnými aplikacemi od hlasového ovládání robotů, počítačů a jiných strojů přes strojový překlad z jednoho jazyka do druhého až třeba po porozumění úplným textům dokumentů umožňující vyhledání všech dokumentů majících stejný obsah jako zadaný dokument. Ne všechny takové aplikace jsou v současné době realizovány.

Na relativně dobré úrovni je automatický překlad z jednoho jazyka do druhého. Podle [POGNAN 90], [STROSSA 94] se každoročně na světě překládá cca 150 000 000 stran textu. Z toho je např. 35% obchodních informací, 21% průmyslové, 20% vědecké, ... a 0.3% tvoří krásná literatura. Náklady na tuto práci tvoří asi 3 000 000 000 USD. Strojovým překladem se od 50. let zabývají desítky pracovišť, mezi jiným i Univerzita Karlova v Praze, viz např. [HAJIČOVÁ 81]. Významný je např. systém SYSTRAN (SYStem of TRANslation), který dosáhl komerčního úspěch a je dostupný pro více než 10 jazykových párů, mimo jiné i pro pár angličtina-arabština.

Podle [STROSSA 94] se dosud vyvinuté systémy strojového překladu dělí na dvě generace. Systémy první generace překládaly způsobem "slovo za slovo". Přes značnou nepřesnost našly aplikace tam kde byla zapotřebí rychlá orientace o co se asi jedná v cizojazyčném textu. Systémy druhé generace nejprve pomocí analýzy vztahů mezi slovy ve zdrojovém textu určí jisté prvky jazyka, ty se převedou na odpovídající prvky cílového jazyka a poté se na základě znalostí struktury vět v cílovém jazyku syntetizují věty výsledného textu. Za dobrý výsledek se považuje správný překlad asi 70% textu, za maximum se považuje 90% správně přeloženého textu. Systémy strojového překladu vyvíjené v současné době se nedělí na generace, používají se v nich nejrůznější prostředky AI.

Pro úplné strojové porozumění přirozenému jazyku je nutné provést *lexikální analýzu* týkající se jednotlivých slov (např. určení druhu, pádu, čísla,...) dále *syntaktickou analýzu* zabývající se vazbami mezi jednotlivými slovy (v podstatě jde o větný rozbor) a *sémantickou analýzu*, při které se analyzuje smysl celého textu, který je nutno vhodným způsobem formálně reprezentovat. Sémantická analýza je nejobtížnější a zdaleka není uspokojivě řešena. Další informace lze nalézt např. v [HAJIČOVÁ 81] nebo [HAJIČOVÁ 88].

Znalostní inženýrství je disciplina AI zabývající se návrhem, tvorbou, naplňováním a využíváním bází znalostí. Jedná se o úlohy řešené v rámci každé aplikace AI. Příkladem báze znalosti jsou pravidla a metaznalosti systému ARES, existuje však řada různých přístupů k tvorbě báze znalosti. Složitým problémem je i vlastní získávání znalostí od experta. Těmto otázkám jsou věnována např. skripta [BERKA 94].

11.4 Poznámky k teorii

Umělá inteligence má vztah k různým teoretickým disciplinám, některé z nich vznikly nezávisle na AI, na jejich rozvoj však AI má vliv, u jiných disciplin se AI krom rozvoje podílela i na jejich vzniku.

Zásadní otázka je, zda principiálně vůbec lze pomocí počítačů vše spočítat nebo případně pomocí strojové manipulace se symboly nějak zjistit. Z předpokladu kladné odpovědi na tuto otázku vycházel Hilbertův program formalizace matematiky [IVÁNEK 83]. V jeho rámci se měly formálně, pomocí symbolického jazyka vyjádřit axiomy a dedukční pravidla z nichž by se, opět pomocí formálních pravidel, odvozovaly všechny matematické věty. K.Gödel v roce 1931 dokázal, že to nelze, viz např. [MENDELSON 64]. Navíc dokázal, že toto omezení formálního odvozování platí kdykoliv budeme chtít, aby výsledky odvozování zahrnovaly všechny matematické věty platné o přirozených číslech.

Dalším, pro AI důležitým faktem je důkaz existence nerozhodnutelných problémů, který provedl A. Turing. [IVÁNEK 85]. Znamená to, že existují symbolicky formulované otázky vyžadující odpověď ANO nebo NE, ke kterým však nelze nalézt algoritmus, který by odpověď zjistil. Nelze například setrojit program pro počítač, jehož vstupem by byl zdrojový text libovolného pascalovského programu spolu s libovolným řetězcem znaků a výstupem by bylo sdělení, zda se vstupující program zastaví pro daný řetězec znaků nebo ne. Stejně tak nemůže existovat program, který by přečetl libovolnou matematicko formuli a sdělil, zda tato formule je či není větou (v matematickém slova smyslu, lze ji formálně dokázat).

Tato fakta však ukazují meze AI jen zdánlivě. Za prvé, přesto že principiálně nelze formální cestou odvodit všechny matematické věty, člověk jich mnoho objevil a dokázal. Má tedy smysl se zabývat tím jak ho pomocí počítačů napodobit, i s vědomím že vše, stejně jako člověk, počítač neobjeví. Za druhé, existují formální systémy ve kterých lze pomocí počítače rozhodnout, zda předložená formule je či není formálně dokazatelná. Příkladem je výrokový počet, kde pro každou formuli lze pomocí pravdivostních tabulek rozhodnout, zda je či není tautologií. Otázka je, jaké formální systémy mají tuto vlastnost a jak jsou prakticky použitelné.

Zmíněnými problémy se zabývá matematická logika. Její význam pro AI nespočívá jen v uvedených výsledcích. Krom výsledků ukazujících nemožnost formálního výpočtu všech pravd a existence nerozhodnutelných problémů se aparátu matematické logiky používá k formalizaci řešených problémů, k reprezentaci znalostí i ke konstrukci různých inferenčních mechanismů, viz např. [MAŘÍK 93] nebo [JIRKŮ 90].

Barierou při řešení různých problémů nemusí být jen jejich principiální formální neřešitelnost. Pro principiálně řešitelné problémy může být značnou a někdy prakticky nepřekonatelnou překážkou i jejich časová nebo paměťová náročnost. V této souvislosti se hovoří o algoritmické složitosti problémů, která se měří pomocí počtu operací nutných k řešení problému. Zkoumá se závislost tohoto počtu na vhodně vyjádřené velikosti řešeného problému. Několik poznámek o složitosti problémů vyhledávání daného prvku z dané množiny je uvedeno v kapitole 13. Problémy složitosti se podrobně zabývá rozsáhlá literatura, např. [KUČERA 83].

Jinou barierou při řešení problémů je vhodná formalizace všech náležitostí problému. Platí to např. pro úlohy týkající se přirozeného jazyka. Těžká je nejen sémantická analýza, ale i syntaktická analýza zabývající se "pravopisem". Každý překladač programovacího jazyka umí rozhodnout, zda daný program je či není syntakticky správně. Používají se k tomu mimo jiné formální gramatiky [HOPCROFT 78], které umožňují formálně přesně a relativně jednoduše popsat všechny syntakticky správné programy. Podstatně složitější je otázka syntaktické správnosti u přirozeného jazyka [HAJIČOVÁ 81].

Pro reprezentaci problémů (nejen AI) se používá mnoho různých typů datových struktur. Podrobným studiem datových struktur vhodných pro různé problémy a složitostí algoritmů na nich operujících se zabývá velmi rozsáhlá literatura, např. [AHO 74], [KNUTH 73], [KUČERA 83].

Kapitola 12

Umělá inteligence a ukládání a vyhledávání dokumentů

12.1 Cíle kapitoly

V odstavci 11.1 bylo zmíněno, že v současné době je patrný trend k široké aplikaci metod a přístupů AI v různých softwarových systémech, přičemž za základní problém je považována "*integrace dílčích komponent do rozsáhlých celků vykazujících kvalitativně nové chování*". To je patrně i způsob, jakým budou využívány prvky AI v systémech pro ukládání a vyhledávání dokumentů. Ve prospěch uvedeného tvrzení hovoří např. implementace systému Metacat [CHEN 92] pro inteligentní vyhledávání dokumentů. Charakteristickým rysem tohoto systému je právě formalizace různých okruhů znalostí používaných při vyhledávání dokumentů a jejich integrace do jednoho softwarového systému.

Cílem této kapitoly je na pozadí popisu systému Metacat:

- naznačit okruhy znalostí používaných při ukládání a vyhledávání dokumentů spolu s osobami, které s nimi pracují (viz odstavec 12.2),
- poukázat na hlavní zdroje potíží při vyhledávání relevantních dokumentů (viz odstavec 12.3),
- ukázat jednu z možných architektur softwarového systému integrujícího komponenty reprezentující různé dílčí znalosti (viz odstavce 12.4 až 12.7),

Podstatným způsobem je čerpáno z článku [CHEN 92]. Není však cílem vzbudit dojem, že v tomto článku uváděné přístupy jsou jediné možné. Přímo v [CHEN 92] jsou stručně charakterizovány jiné systémy určené pro řešení podobných úloh. Patří mezi ně např. systém COALSORT určený pro vyhledávání v bibliografických databázích týkajících se technologie zpracování uhlí [MONARCH 87] nebo systém I3R (Intelligent Intermediary for Information Retrieval), viz [CROFT 87].

Zkušenosti z experimentů se systémem Metacat ukazují, že jeho pomocí lze úspěšně podporovat uživatele při vyhledávání relevantních dokmumentů. Zároveň se však také ukázala potřeba dalšího experimentování se systémem a jeho dalšího vývoje.

12.2 Osoby a okruhy znalostí

Při ukládání a vyhledávání informací se uplatňují tři okruhy znalostí:

- znalosti charakteristiky obsahu (indexování) dokumentů,
- znalosti předmětné oblasti, jíž se dokumenty týkají,
- znalosti vyhledávacího systému.

Zároveň při ukládání a vyhledávání vystupují tři typy osob (v originále *information agents*):

- uživatel, který potřebuje vyhledat dokumenty relevantní k jeho požadavku,
- indexátor, který se zabývá indexováním dokumentů,
- knihovník, který pomáhá uživateli při práci s vyhledávacím systémem.

Různé osoby mají různou úroveň znalostí z jednotlivých okruhů. I když existují vyjímky, obecně platí že uživatel nemá znalosti o indexování, jeho znalosti předmětné oblasti jsou značně různé. Indexátor má hluboké znalosti z oblasti indexování dokumentů, prací s vyhledávacím systémem se nezabývá. Knihovník má hluboké znalosti indexování i vyhledávacího systému a obvykle přehledné znalosti předmětné oblasti.

12.3 Neurčitost indexování a vyhledávání informací

Hlavním zdrojem potíží při vyhledávání dokumentů je neurčitost, která je vlastní jak již procesu ukládání, tak i procesu vyhledávání. Ukazuje se, že různí indexátoři charakterizují obsah jednoho dokumentu různým způsobem. Stává se dokonce, že jeden indexátor při opakovaném indexování stejného dokumentu použije různé charakteristiky obsahu.

Problémem je i skutečnost, že při indexování je aplikován princip snižování redundance popisu. Mimo jiné to znamená, že pro každý dokument je vybrán takový popis, který není ani širší ani užší než téma dokumentu, je tedy indexován celý dokument, ne jeho části (což by samozřejmě přinášelo zase jiné potíže).

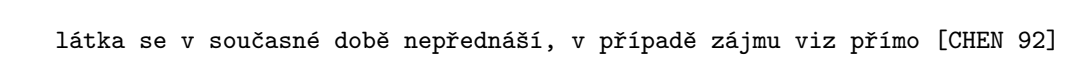
Zdrojem neurčitosti při vyhledávání dokumentů je také tendence uživatelů používat různé vyhledávací výrazy k vyjádření téhož požadavku. Ukazuje se dokonce, že pravděpodobnost použití stejných výrazů k popisu jednoho objektu různými lidmi je menší než 20%. Další potíží spojenou s uživateli je jejich snaha používat spíše širší než užší výrazy pro vyjádření požadavku. Celkovým důsledkem je snížení pravděpodobnosti, že dojde ke shodě charakteristik obsahu použitých indexátorem a uživatelem.

Neurčitosti při vyhledávání dokumentů souvisí i s vyhledávací strategií, kterou uživatel používá. Příkladem jsou strategie "building-block" a "pearl-growing" zmíněné v [CHEN 92]. První z nich spočívá v nejprve samostatném použití několika různých termínů, které jsou později kombinovány pomocí spojek AND, OR a NOT. Druhá strategie používá několik málo termínů k vyhledání prvních dokumentů, z nichž se potom odvodí další vyhledávací termíny. Různé strategie jistě nemusí vést ke stejným výsledkům. Podle [CHEN 92] je v [BATES 79] uvedeno 29 různých strategií které uživatelé používají při vyhledávání dokumentů.

Pokud při vyhledávání dokumentů spolupracuje s uživatelem knihovník, má na výsledek vliv i model uživatele, který si knihovník průběžně vytváří. Tento model zahrnuje např. úroveň vzdělání uživatele, způsob, jak klade otázky, účel vyhledávání a očekávané výsledky.

12.4 Architektura systému Metacat

Architektura systému Metacat je naznačena v obr. 12.1 (jedná se o zjednodušený obr. 2. z [CHEN 92]).



Obrázek 12.1: Architektura systému Metacat

Jedná se o tzv. **blackboard** architekturu, pro kterou jsou charakteristické tři části:

Datové struktury (the blackboard data structure): Jsou to datové struktury obsahující údaje potřebné k řešení problému. Komunikace mezi jednotlivými zdroji znalostí probíhá prostřednictvím těchto datových struktur. Datové struktury systému Metacat jsou charakterizovány v odstavci 12.5.

Zdroje znalostí (the knowledge sources): Jedná se o nezávislé podsystémy, které umožňují pracovat s různými znalostmi týkajícími se problémů k jejichž řešení je celý systém určen. Zdroje znalostí používané systémem Metacat jsou naznačeny v odstavci 12.6.

Řídící modul (the control module) určuje posloupnost operací, které budou prováděny na datových strukturách pomocí jednotlivých zdrojů znalostí. Několik poznámek k řídícímu modulu systému Metacat je v odstavci 12.7.

12.5 Datové struktury

Systém Metacat pracuje s datovými strukturami uvedenými v obr. 12.1. Jedná se o tyto struktury:

- Uživatel (User),
- Úloha (Task),
- Požadavek (Query),
- Dotaz (Index Terms),
- Citace (Citations).

Uživatel (User) je datová struktura obsahující dlohodobě uchovávané údaje o uživateli potřebné pro činnost systému. Jsou to:

- jednoznačná identifikace uživatele,
- stupeň vzdělání (např. profesor, student 1. ročníku, student 2. ročníku, atd.)
- předmětná oblast (biologie, information science, ...),
- stupeň znalostí předmětné oblasti, jíž se dokumenty týkají (nízký, střední, vysoký),
- stupeň znalostí indexování dokumentů (nízký, střední, vysoký),

- stupeň znalostí vyhledávacího systému (nízký, střední, vysoký).

Úloha (Task) je datová struktura charakterizující uživatelovu konkrétní informační potřebu. Zahrnuje tyto položky:

- jednoznačnou identifikaci uživatele,
- typ požadovaných dokumentů (např. kniha nebo článek),
- účel vyhledávání (např. semestrální nebo diplomová práce),
- očekávaný počet vyhledaných dokumentů,
- aktuálnost vyhledaných dokumentů (datum vydání),
- očekávanou úroveň úplnosti (nízká, střední, vysoká, předpokládá se střední),
- očekávanou úroveň přesnosti (nízká, střední, vysoká, předpokládá se střední).

Požadavek (Query) je datová struktura obsahující uživatelem zadané charakteristiky relevantních dokumentů (pokud je zná):

- jména autorů,
- názvy dokumentů
- jednoznačnou identifikaci dokumentů (pokud je známa),
- vyhledávací výrazy zadané uživatelem.

Dotaz (Index Terms) jsou deskriptory odvozené systémem.

Citace (Citations) jsou citace systémem vybraných dokumentů.

12.6 Zdroje znalostí

Zdroje znalostí jsou na sobě nezávislé moduly které komunikují prostřednictvím datových struktur popsaných v předchozím odstavci. Jsou ve vhodnou chvíli aktivovány řídícím algoritmem. Jedná se o následující zdroje znalostí:

- User Model Builder,
- Task Model Builder,
- Suffixing Algorithm,
- Stop Word List,
- Online Thesaurus,
- Known Item Instantiater,
- Heuristic Keyword Searcher,
- Thesaurus Browser.

User Model Builder vytváří dlouhodobý model uživatele. Používá pravidla typu IF ... THEN ... ELSE ... . Systém pracuje podle těchto zásad:

1. Požádá uživatele o jeho identifikační kód, stupeň vzdělání a předmětnou oblast.
2. Poté odvodí stupeň znalostí předmětné oblasti:

   DEFAULT stupeň znalostí předmětné oblasti = STŘEDNÍ

   IF absolvent THEN stupeň znalostí předmětné oblasti = VYSOKÝ

   IF student prvních dvou ročníků THEN stupeň znalostí předmětné oblasti = NÍZKÝ
3. Odvodí stupeň znalostí indexování dokumentů:

   DEFAULT stupeň znalostí indexování dokumentů = NÍZKÝ

   IF specializace v Information Science THEN

   stupeň znalostí indexování dokumentů = VYSOKÝ
4. Při opakovaném používání se stupeň znalostí vyhledávacího systému postupně zvyšuje z počáteční hodnoty NÍZKÝ na STŘEDNÍ při několikrát prováděném vyhledávání a poté na VYSOKÝ při mnohokrát prováděném vyhledávání.

Task Model Builder vytváří model konkrétní uživatelovy informační potřeby. Stejně jako User Model Builder používá pravidla typu IF ... THEN ... ELSE ... . Dohromady se jedná asi o 20 pravidel.

Systém nejprve požádá o:

- typ požadovaných dokumentů (např. kniha nebo článek),
- účel vyhledávání (např. semestrální nebo diplomová práce),
- očekávaný počet vyhledaných dokumentů,
- aktuálnost vyhledaných dokumentů (datum vydání).

Poté průběžně aplikuje tři typy pravidel:

a) Pravidla zajišťující, že typ vyhledávaného materiálu a účel vyhledávání jsou v souladu s uživatelovým profilem. Platí například, že uživatelé s nižší znalostí problematiky vyžadují spíše přehledové materiály a naopak, že uživatelé s vyšší znalostí vyžadují materiály na vyšší odborné úrovni.

b) Pravidla týkající se Library of Congress Subject Headings (LCSH), která reprezentují znalosti LCSH. Jejich pomocí je možno vyjádřit např. požadavek, aby vyhledaný dokument byl sborníkem z kongresu.

c) Pravidla týkající se přesnosti a úplnosti, která vyjadřují například takové skutečnosti, že více vzdělaní uživatelé vyžadují vyšší úplnost nebo že začátečníci nevyžadují vysokou úplnost ale vysokou přesnost.

Suffixing Algorithm umožňuje generovat další potenciální klíčová slova na základě slov zadaných uživatelem. Obsahuje slovník asi 28 000 kořenů slov a asi 30 pravidel pro práci se suffixy (ive, ion, tion, en, ions, atd.). Například ze slova *create* se vytváří slova *creative, creation* a *creations*.

Stop Word List obsahuje asi 160 slov (on, in, ..., she, I, ..., would, will, ...). Používá se pokud je třeba taková slova odstranit z uživatelova požadavku.

Online Thesaurus reprezentuje elektronickou formu znalostí o předmětné oblasti a o indexování. V případě experimentální verse systému Metacat se jednalo o 3500 termínů z oblasti

```
látka se v současné době nepřednáší, v případě zájmu viz přímo [CHEN 92]
```

Obrázek 12.2: Příklad položky online tezauru

matematiky a computer science. V obr. 12.2 je příklad položky tohoto tezauru, kurzivou jsou uvedeny vysvětlivky, které nejsou součástí záznamu.

Suffixing Algorithm, Stop Word List a Online Thesaurus reprezentují znalosti o slovech. Zbývající tři knowledge sources, Known Item Instantiater, Heuristic Keyword Searcher a Thesaurus Browser jsou vyhledávací procedury simulující expertní postupy při vyhledávání.

Known Item Instantiater se používá, pokud uživatel zná některé relevantní dokumenty, případně jenom jejich autora. Charakteristiky známých dokumentů se použijí pro vyhledání nových dokumentů. Uživatel posoudí relevantnost takto vyhledaných dokumentů, charakteristiky vybraných relevantních dokumentů mohou být opět použity k vyhledávání dalších dokumentů.

Heuristic Keyword Searcher slouží k nalezení deskriptorů na základě vyhledávacích termínů zadaných uživatelem. Systém používá 10 funkcí, které různým způsobem porovnávají vyhledávací výraz s deskriptory dokumentů. Funkcím je přiřazena různá váha. Argumentem funkce je vyhledávací výraz, výsledkem funkce je deskriptor nebo prázdná hodnota.

Funkce s nejvyšší vahou vydá deskriptor pouze pokud vyhledávací termín je přímo deskriptorem. Pokud ne, tak se volá funkce s bezprostředně nižší vahou. Ta vydá deskriptor pokud lze deskriptor získat rozšířením vyhledávacího termínu o suffix. Pokud tomu tak není, volá se další funkce, která požaduje souhlas vyhledávacího termínu s nedeskriptorem z tezauru (v tom případě je jejím výstupem příslušný deskriptor). Takto se postupuje, dokud se nenajde deskriptor nebo dokud se nepoužije všech deset funkcí.

Thesaurus Browser je prostředek pro získání dalších deskriptorů z deskriptorů již nalezených. Používá se Online Thesaurus (viz výše), který se chápe jako síť. Jednotlivé deskriptory jsou uzly, odkazy používané v tezauru tvoří spoje sítě. Dosud nalezené deskriptory jsou výchozími uzly pro nacházení dalších deskriptorů. Pro daný výchozí deskriptor se při hledání nových deskriptorů nejprve využijí odkazy na podřazené deskriptory, pak na asociované a nakonec na nadřazené deskriptory. Přitom se dříve navštěvují uzly s malým počtem sousedů, které mají specifičtější význam než uzly s větším počtem sousedů. Nepoužívají se však deskriptory vzdálené o více než dva odkazy a dříve se využívají bližší deskriptory než vzdálenější.

12.7 Řídící modul

Řídící modul pracuje podle těchto zásad:

1. Nejprve se pomocí User Model Builder vytvoří (nebo aktivuje již vytvořený) model uživatele.

2. Pomocí Task Model Builder se zaznamená uživatelova informační potřeba, odpovídající datová struktura je průběžně aktualizována.

3. Pokud uživatel zná některé relevantní dokumenty nebo jejich autory, použije se Known Item Instantiater k nalezení dalších relevantních dokumentů. Pokud ne, pokračuje se dle 5.

4. Nalezené dokumenty se předloží uživateli k posouzení relevance. Pokud je získán požadovaný počet dokumentů, práce systému končí.

5. Podle situace se použije Known Item Instantiater, Heuristic Keyword Searcher nebo Thesaurus Browser k nalezení dalších dokumentů.

6. Heuristic Keyword Searcher pracuje tak, že najde deskriptory a předloží je uživateli. Ten vybere deskriptory, které jsou podle jeho názoru vhodné a na jejich základě se vyhledají další dokumenty. Poté se pokračuje dle bodu 4. Heuristic Keyword Searcher používá Suffixing Algorithm, Stop Word List a Online Thesaurus.

7. Thesaurus Browser také vyhledává nové deskriptory a předkládá je uživateli. Následuje stejný postup jako v bodu 6). Thesaurus Browser však používá pouze Online Thesaurus.

Kapitola 13

Algoritmy a datové struktury pro vyhledávání informací

13.1 Úvod do problematiky

Tato kapitola je věnována ukládání a vyhledávání informace vyjádřené pomocí kódů v tom smyslu, jak bylo uvedeno v úvodu. Kódem se tedy rozumí čísla nebo znakové řetězce. Doposud jsme se zabývali ukládáním informací ve formě volného textu a soustředili jsme se na problémy spojené s charakteristikou obsahu. Principem řešení tohoto problému je vyjádření obsahu pomocí vhodných kódů.

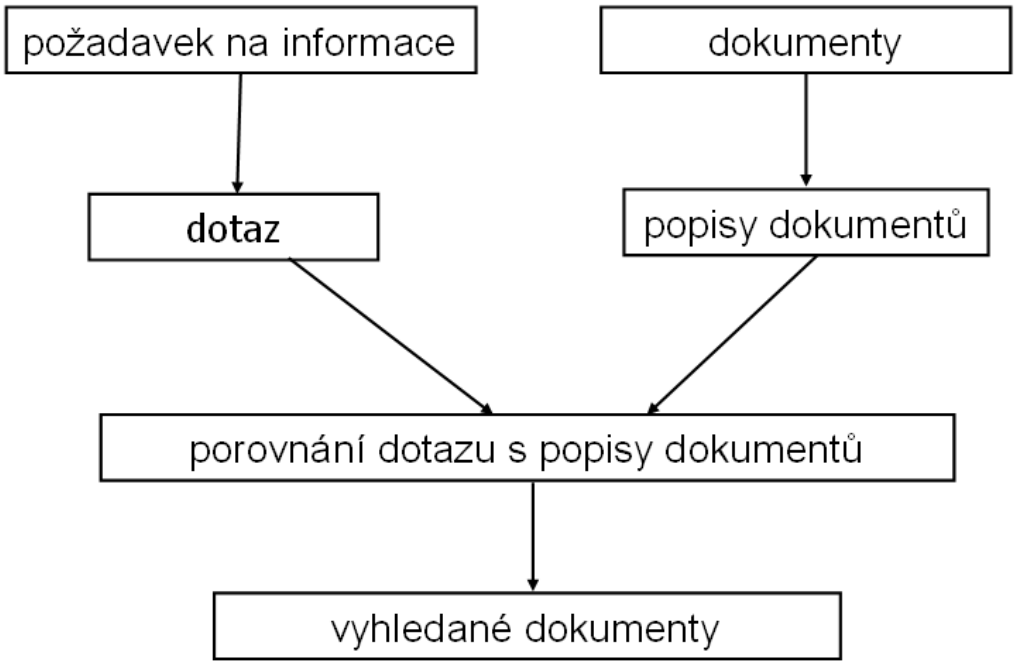
Naprosto zřejmé je to u charakteristiky obsahu pomocí systematických selekčních jazyků a předmětových hesel (viz odstavce 4.3 a 4.4). Je tomu tak ale i při práci s klíčovýni slovy nebo deskriptory. Při vyhledávání pomocí booleovského modelu pracujeme s klíčovými slovy jako s řetězci znaků. V podstatě jde o zjištění, zda zadané slovo patří nebo nepatří mezi klíčová slova dokumentu. O obsah dokumentu se přitom nestaráme. Ten je více nebo méně vhodně vyjádřen klíčovými slovy které jsou dokumentu přiřazeny.

Plnotextové systémy umožňující vyhledávat podle všech nebo většiny slov z textu je z tohoto pohledu možno chápat jako systémy s rozsáhlejší množinou klíčových slov. Více klíčových slov nic nemění na tom, že i v tomto případě pouze zjišťujeme, zda zadané slovo patří nebo nepatří mezi klíčová slova dokumentu. Opět se tedy jedná o práci s kódovanou informací.

Typickým příkladem práce s kódovanou informací je již v úvodu zmíněná úloha vyhledání všech zaměstnanců expedičního oddělení, jejichž měsíční plat je vyšší než 12 000 Kč. Mezi často řešené úlohy patří dále např. zjištění, zda zaměstnanec s daným jménem pracuje v daném oddělení nebo setřídění dodavatelů podle objemu dodávek.

Mezi úlohy týkající se práce s kódovanou informací patří i úlohy vyhledávání dokumentů podle vhodně kódovaného obsahu. Zdůraznit je třeba že se jedná o kódovaný obsah. Pro připomenutí uvádíme v obr. 13.1 schema ukládání a vyhledávání textových dokumentů uvedené již v obr 2.1. Doposud jsme se zabývali

problémy spojenými s charakteristikou obsahu dokumentů případně tvorbou dotazu. Nyní se zabýváme porovnáváním popisů dokumentů a dotazu, tedy manipulací s kódovanou informací.



Obrázek 13.1: Ukládání a vyhledávání dokumentů

Podstatou zmíněných a dalších příbuzných úloh ohledně kódované informace je několik základních operací nad množinami prvků. Patří k nim [AHO 74]:

- operace MEMBER(a,S) která zjišťuje, zda prvek a patří k množině S,
- operace INSERT(a,S) která přidá prvek a k množině S,
- operace DELETE(a,S) která vyloučí prvek a z množiny S,
- operace MIN(S) která najde nejmenší z množiny S.

V souvislosti s aplikacemi počítačů bylo nalezeno mnoho datových struktur a na nich operujících algoritmů pro realizaci uvedených i dalších operací. Tyto algoritmy využívají uspořádání prvků se kterými pracují. Z pohledu algoritmů je lhostejné, o jaké uspořádání se jedná. Může být použito jak např. uspořádání platů podle velikosti tak i uspořádání řetězců písmen podle abecedy (lexikografické uspořádání, které je stejné jako abecední uspořádání hesel ve slovníku).

Různé datové struktury a algoritmy se hodí pro různé typy úloh. Kriteriem použitelnosti je časová a paměťová náročnost algoritmů. Ukazuje se například, že dva různé algoritmy mohou potřebovat k řešení jedné úlohy velmi rozdílnou dobu výpočtu. S těmito problémy je spojena značně rozsáhlá teorie výpočtové složitosti. I pouhý přehled dílčích problémů, kterými se teorie složitosti zabývá je mimo rámec těchto skript. Základní informace lze získat např. v [AHO 74], [KUČERA 83] nebo v [SALTON 89].

Cílem této kapitoly je naznačit, jak výrazně může ovlivnit použitý algoritmus dobu výpočtu. V odstavcích 13.2 až 13.4 jsou postupně stručně popsány tři přístupy k vyhledávání prvků v množině. Jedná se o sekvenční vyhledávání, modifikované sekvenční vyhledávání využívající pravděpodobnosti požadavků na vyhledání jednotlivých prvků a o binární vyhledávání. Zásadní rozdíl mezi sekvenčním a binárním vyhledáváním je ukázán v odstavci 13.5.

Invertovanému souboru, který je datovou strukturou používanou pro vyhledávání dokumentů splňujících danou booleovskou podmínku na výskyt klíčových slov, je věnována samostatná kapitola 14.

13.2 Sekvenční vyhledávání

Tento odstavec se týká vyhledávání prvků v dané množině. Budeme pracovat s množinou jmen autorů. Motivací k této úloze je vyhledávání knih podle autorů. Můžeme si představit, že ke každému autorovi je založen jeden záznam obsahující jeho jméno a odkaz na podrobnější údaje o něm včetně vhodně uloženého seznamu jeho knih. Pro zjednodušení nebudeme pracovat s křestními jmény ani se nebudeme zabývat zmíněným odkazem na podrobnější údaje. Budeme také předpokládat, že nejsou dva autoři se stejným jménem. Jde nám pouze o porovnání doby potřebné k vyhledání zadaného autora při různých přístupech k řešení.

Jedním z možných přístupů k řešení problému je definice dostatečně dlouhého pole řetězců, do kterého ukládáme autory podle přírůstkových čísel, tedy v tom pořadí, jak byly jejich knihy zařazovány do fondu knihovny. Autor je do pole přidán při prvním zařazení některé jeho knihy do fondu. Po zařazení N autorů může příslušné pole vypadat dle 13.1:

položka	autor
1	Kadlec
2	Zounarová
3	Novák
4	Vlach
5	Brach
6	Loukota
.	.
.	.
.	.
N-1	Jánská
N	Doudlebská

Tabulka 13.1: Pole autorů podle přírůstkových čísel

Hledáme-li v takovémto poli např. Loukotu, musíme postupně projít prvních šest položek a porovnat řetězec "Loukota" s řetězcem v položkách uloženým. Vzhledem k sekvenčnímu procházení pole bez využití další informace se tomuto přístupu říká **sekvenční vyhledávání**.

Přirozenou otázkou je, kolik porovnání musíme průměrně udělat při vyhledávání autorů v takto implementované množině autorů. Pokud se hledaný autor v poli nevyskytuje, zjistíme to až po provedení N porovnání. Průměrný počet porovnání je tedy ovlivněn tím, jak časté jsou dotazy na autory neskytující se v seznamu.

Omezíme se proto na hledání průměrného počtu porovnání za předpokladu, že všichni vyhledávaní autoři se v poli vyskytují. Zde dále záleží na tom, jak často je který autor vyhledáván, což je dáno zájmem čtenářů. V extrémním případě může být vyhledáván pouze Kadlec, potom je průměrný počet porovnání 1. V opačném extrémním případě může být vyhledávána pouze Doudlebská, potom je průměrný počet porovnání řetězců N.

Pro odhad průměrného počtu porovnání se proto omezíme na poněkud nereálnou situaci že pro všechny autory je stejná pravděpodobnost vyhledávání. Vyjdeme z toho, že k vyhledání prvního autora zapotřebí 1 porovnání, obecně pro vyhledání k-tého autora je zapotřebí k porovnání. Jestliže tedy budeme hledat každého z N v poli uchovávaných autorů právě jednou, budeme potřebovat celkem

$$1 + 2 + 3 + ... + (N-2) + (N-1) + N = \frac{N * (N+1)}{2}$$

porovnání, průměrně na jednoho autora, tedy $\frac{N+1}{2}$ porovnání.

Toto číslo je někdy používáno jako charakteristika výpočtové (časové) náročnosti algoritmu sekvenčního vyhledávání. Charakteristikou tohoto algoritmu je i to, že přidání nového autora je velmi jednoduché, jedná se o přidání nové položky na konec pole.

13.3 Modifikované sekvenční vyhledávání

V předchozím odstavci jsme naznačili, že pravděpodobnost požadavku na vyhledání se pro jednotlivé autory liší. Tuto informaci lze využít pro optimalizaci průměrného počtu porovnání. Způsob využití je naznačen v tab. 13.2 a tab. 13.3. V tab. 13.2 jsou uvedeny pravděpodobnosti požadavků na vyhledání jednotlivých autorů (omezili jsme se na 5 autorů) Tyto pravděpodobnosti lze získat dlouhodobějším sledováním provozu knihovny. Je přirozené přerovnat seznam autorů tak, aby častěji vyhledávaný autor předcházel méně často vyhledávaného autora. Seznam po přerovnání je v tab. 13.3.

položka	autor	pravděpodobnost
1	Kadlec	0.15
2	Zounarová	0.04
3	Novák	0.40
4	Vlach	0.11
5	Brach	0.30

Tabulka 13.2: Pole autorů podle přírůstkových čísel s pravděpodobnostmi požadavků na vyhledání

položka	autor	pravděpodobnost
1	Novák	0.40
2	Brach	0.30
3	Vlach	0.11
4	Kadlec	0.15
5	Zounarová	0.04

Tabulka 13.3: Pole autorů podle pravděpodobností požadavků na vyhledání

Pokud budeme v případě dle tab. 13.2 vyhledávat autora, pak s pravděpdobností 0.15 to bude Kadlec, na kterého je zapotřebí jedno porovnání, s pravděpodobností 0.04 to bude Zounarová, na kterou jsou zapotřebí dvě porovnání, s pravděpodobností 0.40 to bude Novák, na kterého jsou zapotřebí tři porovnání, atd. Celkový pravděpodobný počet porovnání tedy bude

$$0.15 * 1 + 0.04 * 2 + 0.40 * 3 + 0.11 * 4 + 0.30 * 5 = 3.37$$

Analogickou úvahou pro situaci v tab. 13.3 dospějeme k číslu

$$0.40 * 1 + 0.30 * 2 + 0.15 * 3 + 0.11 * 4 + 0.04 * 5 = 2.09$$

Jestliže budeme u všech autorů předpokládat stejnou pravděpodobnost požadavku na vyhledání, tedy 0.2, dostaneme výsledek

$$0.2 * 1 + 0.2 * 2 + 0.2 * 3 + 0.2 * 4 + 0.2 * 5 = 3.00$$

Ke stejnému výsledku pochopitelně dospějeme použitím vzorce $\frac{N+1}{2}$ uvedeného v předchozím odstavci. Z uvedených čísel vyplývá výhodnost setřídění záznamů pro snížení počtu porovnávání řetězců. Nevýhodou je nutnost zjišťování pravděpodobností přístupů a udržování pořadí záznamů.

13.4 Binární vyhledávání

Binární vyhledávání je založeno na podstatném využití uspořádání vyhledáváných prvků. Vyhledávané pvky jsou v daném poli uloženy uspořádaně, např. od nejmenšího k největšímu. V případě autorů je vhodné použít uspořádání podle abecedy. V tab. 13.4 je příklad pole obsahujícího 15 autorů takto uspořádaných.

položka	autor
1	Brach
2	Civínová
3	Douděra
4	Farský
5	Hošek
6	Chladný
7	Jánská
8	Kavan
9	Loukota
10	Manský
11	Novák
12	Otava
13	Růžičková
14	Vlárská
15	Zounar

Tabulka 13.4: Pole autorů podle abecedy

Způsob využití uspořádání ukážeme na příkladě. Budeme v té souvislosti hovořit o intervalu položek. Interval $< 3, 5 >$ obsahuje položky $3, 4$ a 5, interval $< 1, 15 >$ obsahuje celé pole, interval $< 12, 12 >$ obsahuje jedinou položku 12 atd.

Předpokládejme, že hledáme v seznamu autorů Manského. V prvním kroku porovnáme hledaný řetězec s prostředním řetězcem celého intervalu $< 1, 15 >$. Je jím je 8. položka, tedy Kavan. Podle abecedy je Kavan před Manským (tedy platí "Kavan" $<$ "Manský"). To ale znamená, že Manský může být pouze v intervalu $< 9, 15 >$. Pro tento interval je prostřední 12. položka, tedy Otava. Protože "Otava" $>$ "Manský", může Manský ležet pouze v intervalu $< 9, 11 >$. Zde je prostřední položka 10, porovnáním hledaného řetězce a řetězce u položky 10 je Manský nalezen.

Tento přístup k vyhledávání prvků se nazývá půlením intervalu, používá se i název binární vyhledávání. Lze ho poněkud formálněji vyjádřit např. takto:

I Prvky se uspořádají do pole (např. vzestupně).

II Půlení intervalu:

- Inicializace: interval := celé pole
- Vyhledávanou hodnotu porovnáme s prostředním záznamem intervalu. V závislosti na výsledku:

= … nalezeno, konec

< … dále hledáme v první polovině intervalu

> … dále hledáme ve druhé polovině intervalu

- Interval nelze rozpůlit: nenalezeno, konec.

Podstatné je, že tímto způsobem se výrazně sníží počet potřebných porovnání řetězců. V každém kroku se totiž snižuje počet prvků, které je třeba prohledat nejvýše na polovinu původního počtu. Rovný polovině může být pokud původní počet je sudý, jinak je menší než polovina. Lze dokázat, že maximální potřebný počet porovnání nutných pro prohledání pole délky N je $[\log_2(N+1)]$. Zde $[X]$ značí horní celou část čísla X, např. $[3.45] = 4$.

Intuitivně lze správnost omezení [log2(N + 1)] pro počet porovnání řetězců pro pole délky N nahlédnout z toho, že např. pole o délce $N = 15$ lze naznačeným způsobem dělit nejvýše 4x, přičemž $log_2(15+1) = 4$, neboť $2^4 = 16$.

Tímto způsobem tedy docílíme logaritmického snížení počtu potřebných porovnání. O jak významnou úsporu se jedná je naznačeno v následujícím odstavci. Nevýhodou je potřeba udržovat uspořádání prohledávané množiny prvků, což však v případě málo časté aktualizace nemusí být příliš náročné.

13.5 Srovnání sekvenčního a binárního vyhledávání

Při sekvenčním vyhledávání je za předpokladů uvedených v odstavci 13.2. průměrný počet porovnání objektů dán vzorcem $\frac{N+1}{2}$, kde N je počet objektů v prohledávané množině. Při binárním vyhledávání je v nejhorším případě zapotřebí $[\log_2(N+1)]$ porovnání.

O jak významný rozdíl se jedná je naznačeno v tabulce 13.5, kde jsou uvedeny počty porovnání pro oba přístupy. Pro ilustraci tohoto rozdílu je uveden i odpovídající čas potřebný k vyhledání prvku. Výpočet času vychází z (asi poněkud nereálného) předpokladu, že jedno porovnání trvá 0.001 vteřiny. Čas potřebný k ostatním operacím je zanedbán.

počet	průměrný případ pro sekvenční vyhledávání		nejhorší případ pro binární vyhledávání	
záznamů	porovnání	čas (vteřin)	porovnání	čas (vteřin)
31	16	0.016	5	0.005
119	60	0.060	7	0.007
1 999	1 000	1.000	11	0.011
9 999	5 000	5.000	14	0.014
999 999	500 000	500.000	20	0.020

Tabulka 13.5: Srovnání sekvenčního a binárního vyhledávání

Kapitola 14

Invertovaný soubor

14.1 Princip invertovaného souboru

Invertovaný soubor je datová struktura používaná pro vyhledávání dokumentů splňujících zadanou podmínku týkající se termínů charakterizujících dokument. V jednoduché formě je tato podmínka booleovským výrazem omezujícím výskyty jednotlivých klíčových slov v dokumentu. Složitější podmínka může zahrnovat např. vzájemnou pozici slov nebo požadavek na výskyt slova v určité části dokumentu (např. název článku nebo název časopisu).

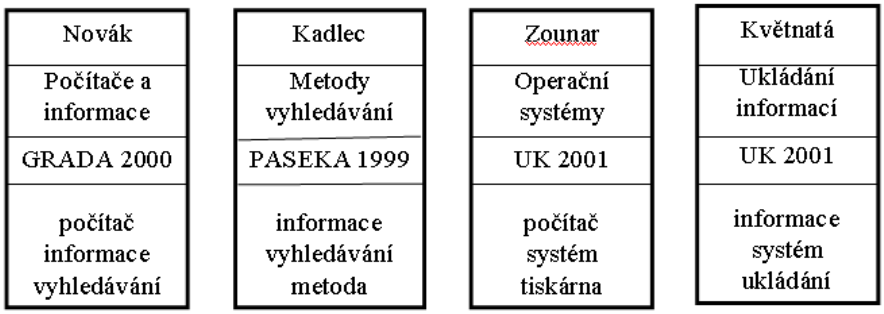
Pro zjednodušení výkladu vysvětlíme princip invertovaného souboru pouze na příkladu klíčových slov. Invertovaný soubor však lze použít a také se v mnoha systémech používá pro vyhledávání podle libovolných termínů (např. autor, název dokumentu, nakladatel, slova z plného textu dokumentu, atd.)

Rozsah prohledávaného fondu dokumentů je často velmi značný (např. statisíce). Může se jednat o klasický knihovní fond reprezentovaný bibliografickými záznamy uchovávanými v počítači, v případě plnotextového systému jsou to celé texty dokumentů. Testovat při každém dotazu záznamy všech dokumentů by bylo časově velmi náročné. Důležitý je i fakt, že jedno klíčové slovo se obvykle vyskytuje v relativně malém počtu dokumentů z celého fondu. Výsledkem testování záznamu dokumentu by tedy ve většině případů bylo konstatování, že dokument slovo neosahuje.

Princip invertovaného souboru spočívá v tom, že informace potřebné k zodpovězení dotazů jsou soustředěny přímo k jednotlivým klíčovým slovům. V nejjednodušším případě se pro každé klíčové slovo jedná o seznam dokumentů, ve kterých se vyskytuje.

V obr. 14.1 jsou uvedeny čtyři zjednodušené záznamy dokumentů, obsahující identifikační číslo dokumentu, autora, název dokumentu a klíčová slova. Invertovaný soubor obsahuje pro každé použité klíčové slovo seznam identifikačních čísel dokumentů, ve kterých se vyskytuje. Seznamy čísel dokumentů odpovídající záznamům z obr. 14.1 jsou v obr. 14.2.

Konkrétním způsobem výpočtu invertovaného souboru se nebudeme zabývat. Lze si však představit, že vzniká invertováním matice dokumentů naznačené v obr. 14.3. Řádky této matice odpovídají dokumentům, sloupce klíčovým slovům. Jestliže dokument obsahuje klíčové slovo, je v příslušném řádku a sloupci matice dokumentů uvedeno "1", v opačném případě "0". Invertováním matice dokumentů dostaneme matici klíčových slov dle obr. 14.4. Seznamy čísel dokumentů uvedené v obr. 14.2 jsou pouze stručným zápisem jednotlivých řádků matice klíčových slov.



Obrázek 14.1: Záznamy dokumentů

Počítač	:	1,3
Informace	:	1,2,4
Vyhledávání	:	1,2
Metoda	:	2
Systém	:	3,4
Tiskárna	:	3
Ukládání	:	4

Obrázek 14.2: Seznamy čísel dokumentů odpovídající záznamům z obr. 14.1

dokument	Počítač	Informace	Vyhledávání	Metoda	Systém	Tiskárna	Ukládání
1	1	1	1	0	0	0	0
2	0	1	1	1	0	0	0
3	1	0	0	0	1	1	0
4	0	1	0	0	1	0	1

Obrázek 14.3: Matice dokumentů

klíčové slovo	dokument			
	1	2	3	4
Počítač	1	0	1	0
Informace	1	1	0	1
Vyhledávání	1	1	0	0
Metoda	0	1	0	0
Systém	0	0	1	1
Tiskárna	0	0	1	0
Ukládání	0	0	0	1

Obrázek 14.4: Matice klíčových slov

14.2 Vyhledávání pomocí invertovaného souboru

Invertovaný soubor je vytvářen pro vyhledávání dokumentů splňujících zadanou podmínku týkající se termínů charakterizujících dokument. V případě booleovského výrazu omezujícího výskyty jednotlivých klíčových slov v dokumentu stačí pro každé klíčové slovo seznam dokumentů, ve kterých se slovo vyskytuje. Dokument může být reprezentován například svým číslem, které zároveň určuje adresu celého záznamu dokumentu. Jednotlivé booleovské operace jsou realizovány pomocí množinových operací na těchto seznamech.

Předpokládejme například, že invertovaný soubor obsahuje klíčová slova dle obr. 14.2. Ta jsou v invertovaném souboru seřazena podle abecedy, viz obr. 14.5.

Informace	:	1,2,4
Metoda	:	2
Počítač	:	1,3
Systém	:	3,4
Tiskárna	:	3
Ukládání	:	4
Vyhledávání	:	1,2

Obrázek 14.5: Invertovaný soubor odpovídající záznamům z obr. 14.1

Pro booleovský dotaz **Informace** AND **Metoda** získáme seznam čísel všech vyhovujících dokumentů jako průnik seznamů pro obě klíčová slova, tedy

$$\{1,2,4\} \bigcap \{2\} = \{2\}$$

Pro booleovský dotaz **Metoda** OR **Počítač** získáme seznam čísel všech vyhovujících dokumentů jako sjednocení seznamů pro obě klíčová slova, tedy

$$\{2\} \bigcup \{1,3\} = \{1,2,3\}$$

Pro booleovský dotaz **Informace** $AND\ NOT$ **Ukládání** získáme seznam čísel všech vyhovujících dokumentů jako množinový rozdíl seznamů pro obě klíčová slova, tedy

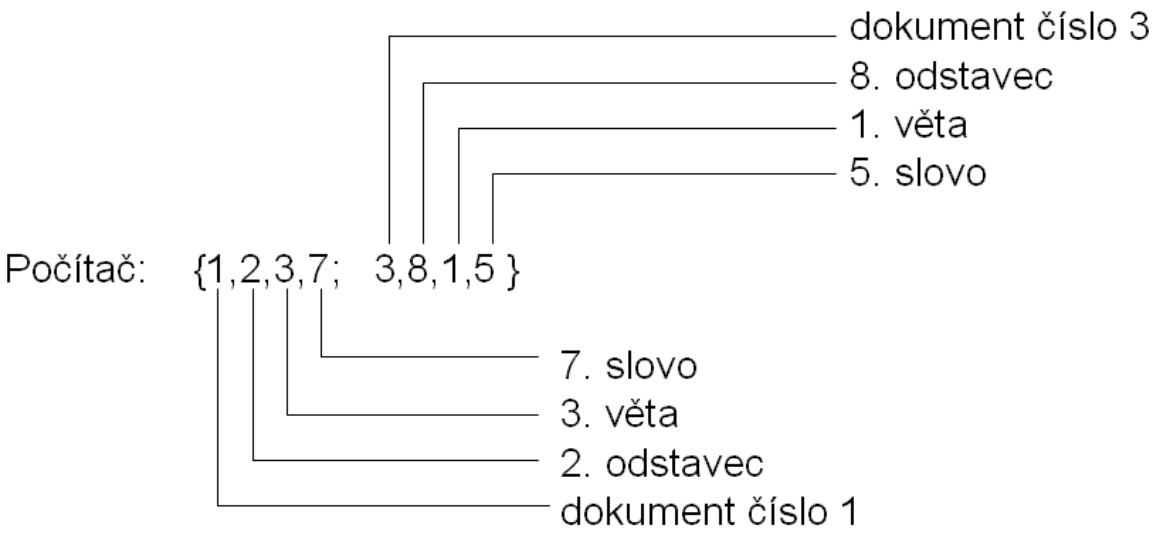
$$\{1,2,4\} - \{4\} = \{1,2\}$$

Složitější booleovský dotaz lze realizovat odpovídající posloupností množinových operací.

V různých vyhledávacích systémech lze používat i další doplňující podmínky na klíčová slova, např. že klíčové slovo A předchází klíčové slovo B o maximální daný počet slov, nebo že dvě klíčová slova jsou ve stejné větě. V tom případě obsahuje invertovaný soubor pro každé klíčové slovo nejen seznam čísel dokumentů ve slově, ale i potřebné údaje o pozici slova v dokumentu. Příklad takto rozšířeného seznamu pro klíčové slovo počítač je v obr. 14.6.

14.3 Pravostranné a levostranné rozšíření

Invertovaný soubor soustřeďuje ke každému klíčovému slovu informace o jeho výskytech ve všech dokumentech fondu. Díky tomu lze rychle vyhledávat všechny dokumenty splňující podmínky týkající se výskytu zadaných slov v dokumentech včetně podmínek na lokalizaci slov uvnitř dokumentů.



Obrázek 14.6: Rozšíření invertovaného souboru o pozici slova

Uspořádání invertovaného souboru podle abecedy umožňuje pracovat s pravostranným rozšířením, viz odstavec 6.4. Příkladem použití pravostraného rozšíření je booleovský dotaz

Inform* AND **Metoda**.

Jestliže mezi použitá klíčová slova začínající řetězcem **Inform** patří slova **Inform**ace, **Inform**atika, **Inform**atizace a žádná jiná, pak výraz **Inform*** AND **Metoda** je zápisem dotazu

(**Inform**ace AND **Metoda**) OR (**Inform**atika AND **Metoda**)
OR (**Inform**atizace AND **Metoda**)

Příslušný úsek invertovaného souboru může vypadat např. takto:

...

Ikona

Imaginární

Informace

Informatika

Informatizace

Ironie

...

Podstatné je, že při vyhodnocování dotazu **Inform* AND Metoda** stačí nalézt v invertovaném souboru první slovo začínající řetězcem **Inform** a vzít v úvahu všechna bezprostředně následující slova také začínající tímto řetězcem. Pochopitelně nemusíme předem vědět, kolik takových slov je.

Invertovaný soubor setříděný podle abecedy však nelze analogicky použít pro levostranné rozšíření. Příkladem levostranného rozšíření je

CD-ROM AND ***ie**.

Záměrem tohoto dotazu je získat všechny dokumenty, mezi jejichž klíčová slova patří **CD-ROM** a název některé vědy končící na **ie**. Může to být například archeologie, filosofie nebo zoologoie. Takovéto názvy věd však budou rozptýleny po celém invertovaném souboru setříděném podle abecedy. Vzhledem k tomu, že předem nevíme, o jaké vědy se jedná, bylo by nutné projít celý invertovaný soubor, který však může být značně rozsáhlý.

Uvedený problém lze odstranit, pokud vytvoříme další, opět podle abecedy setříděný invertovaný soubor, ve kterém budou klíčová slova napsaná pozpátku. Jestliže mezi názvy věd patří archeologie, biologie, chemie, filosofie, filologie a zoologoie, pak příslušný úsek tohoto invertovaného souboru vypadá takto:

...

eifosolif

eigoloehcra

eigoloib

eigololif

eimehc

eiogolooz

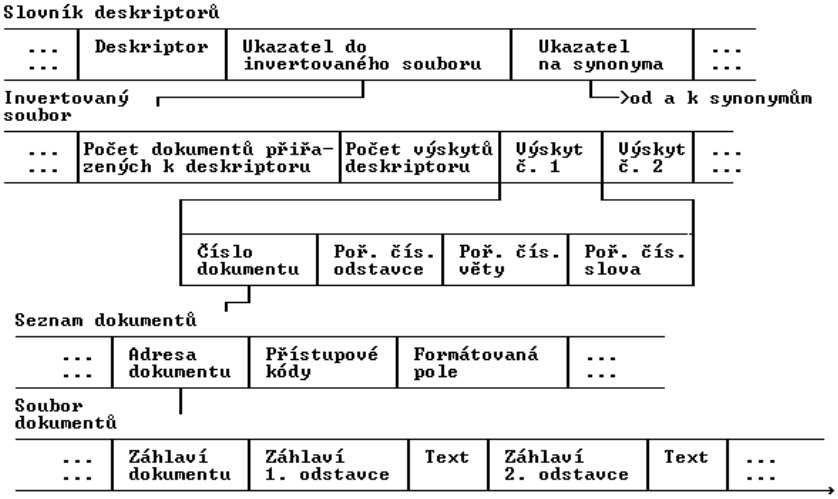
...

Na takto uspořádaný invertovaný soubor již pro zodpovězení dotazu **CD-ROM** AND ***ie** použít analogický postup jako pro pravostranné rozšíření.

14.4 Architektura souborů

Rešeršní softwarové systémy musí krom invertovaného souboru obsahovat i vhodně uložené dokumenty a případně další informace. Způsob uložení těchto informací může být různý. V obr. 14.7 je uvedena dle [SALTON 89] typická architektura souborů rešeršního systému.

Pracuje se zde s deskriptory, ne s klíčovými slovy. Je k dispozici slovník deskriptorů, pro jednotlivé deskriptory jsou dále k dispozici synonyma. Synonyma lze použít k automatickému rozšiřování dotazů, viz odstavec 6.7. Dokumenty jsou uloženy ve dvou souborech. Seznam dokumentů obsahuje záznamy pevné délky, texty proměnlivé délky jsou v samostatném souboru.



Obrázek 14.7: Příklad architektury souborů

Literatura

AHO 74: Aho, A.V. - Hopcroft, J.E.-Ullman, J.D.: The design and Analysis of Computer Algorithms. Addison - Wesley Reading, Mass. 1974.

BABKA 94: Babka, M.: Kde a jak hledat informace o firmách. 1. vydání. Praha, Management Press 1994. 174 s.

BERKA 94: Tvorba znalostních systémů. Praha, Vysoká škola ekonomická 1994, 190 s.

BRADY 85: Brady, J.M.: Artificial Intelligence and Robotics. Artificial Intelligence, vol. 26, 1985, s 79 - 121

CHEN 92: Chen, H.: Knowledge-based document retrieval: framework and design. Journal of information science, 18, 1992, č. 4, s. 293 - 314

CROFT 87: Croft, W.B. - Thompson, R.H.: I3R: a new approach to the design of document retrieval systems. In: B.C.Brookes, ed. Intelligent Information Systems for the Information Society. North-Holland, Amsterdam, 1986, s. 389-404

ČERNÁ 92: Černá, M. - Stöcklová, A.: Služby knihoven a informačních středisek. 1. vydání. Praha, Univerzita Karlova 1992, 98 s.

GRUSKA 83: Gruska, J. a kol.: Počítačová revolúcia. In SOFSEM'83, Brno, ÚVT UJEP 1985, s. 7 - 64

HÁJEK 85: Hájek, P. - Jirků, P.: Lesk a bída expertních systémů. In SOFSEM'85, Brno, ÚVT UJEP 1985, s. 131 - 164

HAJIČOVÁ 81: Hajičová, E. - Plátek, M. - Sgall, P.: Komunikace s počítačem v češtině. In SOFSEM'81, Brno, ÚVT UJEP 1981, s. 85 - 113

HAJIČOVÁ 88: Hajičová, E.: Charakteristika diskursu a implementační možnosti. In SOFSEM'88, Brno, ÚVT UJEP 1981, s. 59 - 81

HAVEL 80: Havel, I.M.: Robotika. Úvod do teorie kognitivních robotů. Praha, SNTL 1980, 279 s.

HAVEL 88: Havel, I.M.: Neuronové počítače a jejich inteligence. In SOFSEM'88, Brno, ÚVT UJEP 1988, s. 83 - 118

HLAVÁČ 92: Hlaváč, V. - Šonka, M.: Počítačové vidění. Praha, Grada 1992, 272 s.

HOPCROFT 78: Hopcroft, J.E.-Ullman, J.D.: Formálne jazyky a automaty. Bratislava, ALFA 1978, 343 s.

IVÁNEK 83: Iványk, J.: Logika pro obor ekonomicko - matematické výpočty. Praha, Vysoká škola ekonomická 1983, 75 s.

IVÁNEK 85: Iványk, J.: Základy matematické informatiky I - Informace a automaty. Praha, Vysoká škola ekonomická 1985, 96 s.

IVÁNEK 85A: Iványek, J. - Rauch, J.: Zkušenosti a plány s využitím expertních systémů při řízení spolehlivosti automobilů. In SOFSEM'85, Brno, ÚVT UJEP 1985, s. 38 - 40

JIRKŮ 90: Jirků, P. - Materna, P: Znalosti, logika, usuzování. In SOFSEM'90, Brno, ÚVT UJEP 1990, s. 187 - 206

KNUTH 73: Knuth, D.E.: The Art of Computer Programming, Vol. III: Sorting and Searching. Addison - Wesley Reading, 7Mass. 1974.

KOVÁŘ 84: Kovář, B.: Věcné pořádání informací a selekční jazyky. 2. vydání. Praha, ÚVTEI 1984, 252 s.

KOTEK 93: Kotek, Z. a kol.: Metody rozpoznávání a jejich aplikace. Praha, Academia 1993.

KUČERA 83: Kombinatorické algoritmy. Praha, SNTL 1983, 280 s.

LUKASOVÁ 85: Lukasová, A. - Šarmanová, J.: Metody shlukové analýzy. Praha, SNTL 1985, 197 s.

MAŘÍK 93: Mařík, V. - Štěpánková, O. - Lažanský, J. a kol.: Umělá inteligence (1). Praha, Academia 1993 , 264 s.

MAŘÍK 96: Mařík, V. a kol.: Umělá inteligence (2). Praha, Academia 1996 (v tisku).

MENDELSON 64: Mendelson, E.: Introduction to Mathematical Logic. Princeton, D. Van Nostrand, 1964

MINSKY 67: Minsky, M.: Computation. Finite and Infinite Machines. Prentice - Hall, Engelwood Cliffs, 1967

MONARCH 87: Monarch, I. - Carbonell, J.G.: CoalSORT: a knowledge-based interface, IEEE EXPERT (Spring 1987), s. 39 - 53

NOVÁK 90: Novák, V.: Fuzzy množiny a jejich aplikace. Praha, SNTL 1990, 295 s.

POGNAN 90: Pognan, P.: Full-Text Analysis for Concept Extraction. Application to Czech Technical Texts (Referát přednesený na konferenci Automatic Processing of Text, konané v Praze 20. - 23. listopadu 1990)

RAUCH 86: Rauch J.: ARIS - an Information System for Monitoring Vehicle Reliability. In Proceedings of the XXI International FISITA Congress, Belgrad, 1986

RAUCH 94: Rauch, J.: Metody zpracování informací I - informační zdroje a služby. Praha, Vysoká škola ekonomická 1994, 55 s.

RICH 91: Rich, E. - Knight, K.: Artificial Intelligence - Second Edition. McGraw Hill, Inc., New York, 1991

ROTA 92: Rota, G.: Artificial Intelligence Today. In Applied Artificial Intelligence, McGraw Hill, Inc., New York 1992, s 594 - 600

SALTON 89: Salton, G.: Automatic Text Processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley Publishing Company, Inc. 530 s.

STROSSA 94: Strossa, P.: Zpracování informačních fondů - sešit č. 2, Algoritmizace a automatizace zpracování textových informací. Praha, Vysoká škola ekonomická 1994, 139 s.

TOVEK: Topic - Systém pro vyhledávání dokumentů. Praha, TOVEK s.r.o, 77 s.

UNESCO 91: Teaching package on standardization in information handling. Prepared for the General Information Programme. - Paris: Unesco, 1991, 476 s.

VICKERY 94: Vickery, B, C.- Vickery, A.: Information Science in theory and practice. London, Bowker-Saur 1994, 387 s.

ŽBIRKA 93: Žbirka, J.: Zpracování úplných textů metodami 3. generace. In: DATASEM'93. Brno, CS-COMPEX 1993, s. 205 - 214.