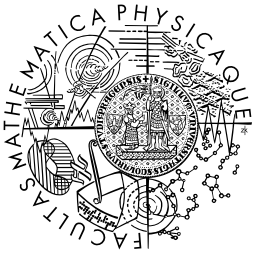


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Patrik Veselý

Vyhledávání známých scén pomocí zpětné vazby a samoorganizujících se map

Katedra softwarového inženýrství

Vedoucí bakalářské práce: doc. RNDr. Jakub Lokoč, Ph.D.

Studijní program: Informatika

Studijní obor: Programování a SW systémy

Praha 2020

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ....

Podpis autora

Chtěl bych poděkovat svému vedoucímu bakalářské práce doc. RNDr. Jakubovi Lokočovi, Ph.D. za ochotu, pomoc a za příležitost posunout své hranice dál. Dále bych chtěl poděkovat RNDr. Miroslavu Kratochvílovi za zapůjčení jeho implementace samoorganizujících se map. Dále bych chtěl poděkovat Bc. Tomášovi Součkovi za poskytnutá data z neuronových sítí.

Název práce: Vyhledávání známých scén pomocí zpětné vazby a samoorganizujících se map

Autor: Patrik Veselý

Katedra: Katedra softwarového inženýrství

Vedoucí bakalářské práce: doc. RNDr. Jakub Lokoč, Ph.D., Katedra softwarového inženýrství

Abstrakt: Vyhledávání v multimediální kolekci bývá často realizováno pomocí textových dotazů a následného seřazení databáze vzhledem k relevanci k poskytnutému dotazu. Nicméně, pokud uživatel hledá pouze jednu konkrétní scénu nebo obrázek, tak musí často prohledávat sekvenčně velkou spoustu výsledků a ani tak nemá garantováno, že hledaný objekt nalezne v rozumném čase. Tato práce se zabývá metodami využití zpětné vazby pro efektivnější dohledávání známých obrázků v rozsáhlé milionové kolekci obrázků. Práce srovnává několik přístupů odhadu relevance a výběru displejů formou simulací zpětné vazby. Experimentálně je prokázáno, že studované modely mohou být významným přínosem pro moderní vyhledávače.

Klíčová slova: zpětná vazba vyhledávání obrázků podle obsahu rankování obrázků hledání známé scény

Title: Known-item search with relevance to SOM feedback

Author: Patrik Veselý

Department: Department of software engineering

Supervisor: doc. RNDr. Jakub Lokoč, Ph.D., Department of software engineering

Abstract: Multimedia searching is usually realized by means of text search, where a large dataset is sorted with respect to a relevance to a given text query. However, if users search for just one scene or image, a sequential browsing of a larger result set is often necessary, without a guarantee that the object is found in a reasonable time. This work focuses on methods relying on relevance feedback for more effective searching in a large collection of one million images. Several relevance update and display selection approaches are compared using simulations of relevance feedback. Our experiments reveal that the investigated models are a benefit to modern multimedia search engines.

Keywords: key words relevance feedback content-based image retrieval image ranking known-item search

Obsah

Úvod

V posledních desetiletích informační technologie zažívají opravdový rozkvět a pronikají do všech odvětví jako je třeba průmyslová výroba, výuka nebo zábava. S tímto pokrokem přichází možnost i potřeba ukládat užitečné informace nejen jako text, ale i jako druh multimédií. Například jako obrázek nebo video. S každým dnem se tyto multimediální databáze rozrůstají, což často vytváří obtížné výzvy. Jedna z těchto výzev je nalezení známého objektu (například scény z videa) v rozsáhlé multimediální databázi bez jakékoliv znalosti metadat tohoto objektu (název videa, čas scény, tagy). Tento typ vyhledávání se nazývá vyhledávání známého objektu (*known-item search*).

Vyhledávání scén na základě obsahu

Vyhledávání známých scén na základě jejich obsahu se jeví jako těžký úkol a trendy ukazují, že univerzální metoda k nalezení hledané scény zatím neexistuje. Aktuální studie navíc ukazují, že nestačí pouze model pro uspořádání databáze vzhledem k jednomu dotazu, ale že jsou zapotřebí i metody pro interaktivní vyhledávání (Rossetto a kol. (2020); Lokoč a kol. (2019)).

Jeden ze způsobů vyhledávání, který je lidem nejbližší, je popis slovy. Mezi výhody tohoto přístupu je možné zařadit jeho jednoduchost a intuitivnost, protože člověk je zvyklý se tímto způsobem vyjadřovat. Jedním z hlavních nedostatků pro formulaci dotazu je však jazyková vybavenost a omezené množství informací o hledané scéně, což může vést k neúplnému nebo zavádějícímu popisu. Další problém se skrývá v subjektivnímu popisu, tedy vstup od více uživatelů se může i výrazně lišit.

Další možná metoda je pomocí barevné skici, kdy uživatel načrtne hledanou scénu. Takto lze snímek z hledané scény najít relativně rychle při správné reprodukci barev, což je výhoda. Ve skutečnosti ale může být pro uživatele velmi obtížné po delší době načrtnout zapamatovanou scénu a odhadnout barvy dostatečně přesně. Další často využívaná metoda pro vyhledávání obrázků je podle podobnosti k jinému obrázku. V této metodě se vyberou a vrátí ty vizuálně či sémanticky nejpodobnější obrázky k zadanému dotazu formou vzorového obrázku. Tato metoda může být použita i pro vyhledávání scén, kdy se scéna rozdělí na jednotlivé reprezentativní snímky. Výhodou této metody je to, že uživatel může dát přesnější a bohatější informaci o scéně, kterou hledá. Problém však zůstává v tom, jak tento vzorový obrázek získá, protože to je podobně těžké jako nalézt hledaný snímek, popřípadě hledanou scénu.

Poslední metoda, kterou bych rád zmínil, je vyhledávání na základě zpětné vazby (Cox a kol. (2000)). V této metodě uživatel vybírá vzorové snímky, které jsou nejpodobnější k hledanému objektu, ale nevyžaduje nutně perfektní vzorový obrázek. Touto zpětnou vazbou se systém učí o relevanci snímků v databázi. Metodou zpětnovazebního učení se budu v této práci zabývat. Existují samozřejmě i další metody - například metoda vyhledávání založená na pozici objektu nebo vyhledávání pomocí mluveného slova.

Efektivita hledání ve videu

Velkou výzvou při tvorbě vyhledávacích algoritmů pro velké kolekce videa je vyhodnocování jejich přesnosti (Lokoč a kol. (2018)). V případě modelů pro rankování objektů databáze vzhledem k dotazu se často využívají benchmark datasety, kde je pro každý dotaz definován korektní výsledek. S takovýmto datasetem je možné měřit efektivitu hledání například pomocí přesnosti a úplnosti, nebo jejich agregace pomocí F1-score nebo MAP. Problém však nastává při měření efektivity interaktivního vyhledávacího systému, kde není možné vytvořit benchmark páry dotaz - výsledek. Uživatel reaguje na základě aktuálně zobrazených informací.

Pro potřeby vyhodnocování interaktivních systémů jsou proto často prováděny evaluace s reálnými uživateli a měří se úspěšnost (případně i rychlost) hledání. Aby byly zajištěny stejné podmínky pro srovnávané systémy, tak jsou organizovány evaluační kampaně jako Video Browser Showdown (Lokoč a kol. (2019); Rossetto a kol. (2020)) nebo Lifelog Search Challenge (Gurrin a kol. (2019)). Tyto soutěže probíhají jednou za rok. Jsou organizovány tak, že se týmy z celého světa sejdou v jedné místnosti a ve stejném čase řeší časově omezené úlohy prezentované na dataprojektoru. Týmy mají za úkol najít krátkou scénu z předem dané rozsáhlé kolekce videa, kterou si před soutěží mohou libovolně předzpracovat.

Přínosy práce

V rámci této práce se zaměřuji na metody hledání ve videích, během kterých se uživatel interaktivně zapojuje do procesu hledání. Konkrétně tato práce zkoumá systém, ve kterém uživatel vybírá ze zobrazeného displeje obrázky, které mu nejvíce připomínají hledaný obrázek. Systém následně tuto zpětnou vazbu vyhodnotí a aktualizuje odhady relevance jednotlivých obrázků v databázi. Na základě těchto nových odhadů pak vybírá nový displej a předkládá jej uživateli do další iterace.

Práce zkoumá různé strategie aktualizací relevance obrázků a výběru displejů, kdy se primárně vychází z výzkumu Cox a kol. (2000). Také se zaměřuji na automatizaci evaluací, které jsou nesmírně důležité pro konfiguraci systému pro soutěže typu Video Browser Showdown. Mezi hlavní přínosy patří:

- Srovnání existujících přístupů s metodou využívající SOM
- Rozsáhlé evaluace založené na simulacích reálných uživatelů
- Integrace modelu do systému SOMHunter

Výsledky této práce také pomáhají při návrhu komplexnějších nástrojů vyvíjených v rámci naší výzkumné skupiny SIRET. Dva z těchto nástrojů byly přijaty na mezinárodní soutěže. První byl nástroj SOMHunter (Kratochvíl a kol. (2020)), který vyhrál mezinárodní soutěž Video Browser Showdown 2020 v Daejeon, Korea. Druhý je vylepšená verze nástroje SOMHunter pro Lifelog Search Challenge (Mejzlík a kol. (2020)). Nástroj SOMHunter je zveřejněn pod svobodnou licencí na veřejném repozitáři[1].

[1] https://github.com/siret/somhunter

Dále jsem spoluautor dvou dalších článků "Is the Reign of Interactive Search Eternal? Findings from the Video Browser Showdown 2020" (aktuálně v recenzním řízení časopisu ACM TOMM, Lokoč a kol. (2020b)) a "A W2VV++ Case Study with Automated and Interactive Text-to-Video Retrieval" (aktuálně přijat na konferenci ACM Multimedia 2020[2], Lokoč a kol. (2020a)). Z těchto článků jsem převzal upravené verze grafů 4.4 a 4.3, na kterých jsem pracoval.

V kapitole 1 se budu zabývat obecným modelem pro zpracování zpětné vazby z výzkumu Cox a kol. (2000), kde shrnu teoretický základ a doplním o možnosti inicializace jiným algoritmem. V kapitole 2 se budu zabývat možnostmi výběru snímků pro displej. Z tohoto displeje pak nadále uživatel vybírá zpětnou vazbu a tím učí systém to, co je relevantní. Zahrnuji metody z výzkumu Cox a kol. (2000), také doplňuji další metody založené na samoorganizujících se mapách (Kohonen (1990)). Třetí kapitola se věnuje automatickým evaluacím algoritmů popsaných v předchozích kapitolách. Nejprve je prezentován model pro odhad výběru snímků z displeje podle reálných uživatelů. Na zkoumání tohoto modelu se podílím ve skupině SIRET. Poté s tímto modelem otestuji množinu konfigurací s cílem najít konfiguraci zvyšující úspěšnost nalezení hledaného snímku. V kapitole 4 se pak věnuji integraci modelu zpětné vazby do systému SOMHunter a jeho zapojení do soutěže Video Browser Showdown 2020.

[2]https://2020.acmmm.org

1. Model pro využití zpětné vazby

V rámci této kapitoly připomenu základní formální rámec pro vyhledávání pomocí zpětné vazby, který byl navržen ve výzkumu Cox a kol. (2000). Budu se držet i stejné notace pro lepší orientaci čtenáře v původním zdroji.

Nechť Img je množina všech obrázků. Snímky z databáze $S \subset Img$ budeme označovat $O_1, ..., O_n$ a hledaný objekt jako T. Během dotazování je uživatel v každé iteraci t vyzván k výběru podmnožiny z displeje D_t, která je nejpodobnější k hledanému objektu T. Tedy hledání jako takové můžeme zapsat jako posloupnost displejů a uživatelských akcí. Historii hledání do iterace t budeme označovat H_t. Uživatelskou akci, jakožto výběr podmnožiny z D_t, budeme označovat A_t. Poté můžeme H_t rozepsat jako množinu $\{D_1, A_1, D_2, A_2, ..., D_t, A_t\}$.

Po t iteracích budeme odhadovat pravděpodobnost objektu O_i, že tento objekt je T za podmínky historie hledání H_t. Tuto pravděpodobnost zapíšeme jako $P(T = O_i|H_t)$. Následně s historií H_t musíme zvolit displej D_{t+1}. Výběru displeje se budu věnovat v kapitole 2.

1.1 Bayesova formulace

Tato sekce předpokládá čtenářovu znalost podmíněné pravděpodobnosti a věty o úplné pravděpodobnosti. Viz Zvára (2019) kapitola 2.

Věta 1. *Nechť T je hledaný snímek z databáze S a H_t je historie hledání. Pak platí, že*

$$\begin{aligned} P(T = O_i|H_t) &= \frac{P(H_t|T = O_i)P(T = O_i)}{P(H_t)} \\ &= \frac{P(H_t|T = O_i)P(T = O_i)}{\sum_{j=1}^{n} P(H_t|T = O_j)P(T = O_j)} \end{aligned}$$

Důkaz. První nerovnost plyne z definice podmíněné pravděpodobnosti a druhá rovnost platí podle věty o úplné pravděpodobnosti.

□

Věta 1 ukazuje, že můžeme zjistit pravděpodobnost, že snímek O_i je hledaný, z vyhodnocení pravděpodobnosti dané historie H_t. Označení $P(T = O_i)$ reprezentuje apriori pravděpodobnost. Tato pravděpodobnost může být inicializována jako uniformní nebo v závislosti na jiné modalitě (předchozí hledání, model klíčových slov).

Cox a kol. (2000) ukázal, jak tento přístup využít k odvození algoritmu pro inkrementální výpočet. Nejprve předpokládejme, že D_t je deterministická funkce z H_{t-1}.

Věta 2. *(Inkrementální vzorec) Nechť T je hledaný snímek z databáze S a H_t je*

historie hledání. Pak platí, že

$$\begin{aligned} P(T = O_i|H_t) &= P(T = O_i|D_t, A_t, H_{t-1}) \\ &= \frac{P(D_t, A_t|T = O_i, H_{t-1})P(D_t, T = O_i|H_{t-1})}{\sum_{j=1}^{n} P(D_t, A_t|T = O_j, H_{t-1})P(T = O_j|H_{t-1})} \\ &= \frac{P(A_t|T = O_i, D_t, H_{t-1})P(T = O_i|H_{t-1})}{\sum_{j=1}^{n} P(A_t|T = O_j, D_t, H_{t-1})P(T = O_j|H_{t-1})} \end{aligned}$$

Důkaz. První rovnost vyplývá z definice historie hledání H_t. Druhá rovnost vyplývá z předchozí věty 1. Poslední rovnost plyne z našeho předpokladu, že D_t je deterministická funkce z H_{t-1}.

□

Nyní máme inkrementální vzorec pro výpočet pravděpodobnosti, že daný objekt je hledaný. Hodnotu $P(T = O_i|H_{t-1})$ již máme předem spočítanou z minulé iterace. V první iteraci je tato hodnota inicializovaná algoritmy, které jsou diskutovány v podkapitole 1.4. Znamená to tedy, že jediná hodnota, která nám zbývá zjistit, je $P(A_t|T = O_i, D_t, H_{t-1})$. Tato pravděpodobnost je v literatuře označována jako *uživatelský model*, protože určuje pravděpodobnost, že uživatel provede akci A_t.

1.2 Uživatelský model

Uživatel má za úkol vybrat snímek z displeje, který je nejpodobnější hledanému, což budeme předpokládat, že je časově invariantní. Dále se pro zjednodušení předpokládá, že různí uživatelé budou vybírat stejně. Tedy *uživatelský model* zjednodušíme na

$$P(A_t|T = O_i, D_t).$$

Dále k odhadu chování uživatele budeme potřebovat odhadnout vizuální podobnosti jednotlivých snímků. K tomu nám poslouží charakteristické rysy z hlubokých neuronových sítí (aktuálně využíváme reprezentace obrázků z modelu W2VV++ Li a kol. (2019)). Zavedeme si funkci $f : Img \to R^n$, která nám bude mapovat snímek O_i do prostoru charakteristických rysů a vrátí daný vektor charakteristických rysů $v_f \in R^n$. Následně budeme uvažovat funkci vzdálenosti

$$d(A, B) = 1 - cossim(f(A), f(B)),$$

kde A a B jsou jednotlivé snímky z databáze, funkce f je mapovací funkce z předchozí věty a funkce *cossim* je kosínova podobnost definovaná

$$cossim(v_A, v_B) = \frac{v_A \cdot v_B}{||v_A|| \cdot ||v_B||}$$

Jednotlivé snímky z displeje označíme

$$D_t = \{X_{t,1}, ..., X_{t,|D_t|}\}.$$

Poté uživatelský model invariantní v čase můžeme rozepsat jako

$$P(A_t|T = O_i, X_{t,1}, ..., X_{t,|D_t|}).$$

K odhadu uživatelského modelu pro výběr právě jednoho snímku $X_{t,a}$ použijeme funkci *softmin*.

$$P(A_t = \{X_{t,a}\}|T = O_i, X_{t,1}, ..., X_{t,|D_t|}) \approx \frac{exp(\frac{-d(X_{t,a},T)}{\sigma})}{\sum_{X_{t,k} \in D_t} exp(\frac{-d(X_{t,k},T)}{\sigma})}$$

Zobecnění pro více snímků může být dosaženo předpokladem, že pravděpodobnosti dvou akcí jsou nezávislé. Tím pádem akce obou zároveň je jejich produkt. Nyní A_t zastává množinu snímků.

$$P(A_t|T = O_i, X_{t,1}, ..., X_{t,|D_t|}) \approx \prod_{X_{t,a} \in A_t} \frac{exp(\frac{-d(X_{t,a},T)}{\sigma})}{\sum_{X_{t,k} \in D_t \setminus A_t \cup \{X_{t,a}\}} exp(\frac{-d(X_{t,k},T)}{\sigma})}$$

Tento uživatelský model dosadíme do věty 2, a poté dostaneme konečný inkrementální vzorec, který budeme používat.

$$P(T = O_i|H_t) = \frac{P(T = O_i|H_{t-1}) \prod_{X_{t,a} \in A_t} \frac{exp(\frac{-d(X_{t,a},O_i)}{\sigma})}{\sum_{X_{t,k} \in D_t \setminus A_t \cup \{X_{t,a}\}} exp(\frac{-d(X_{t,k},O_i)}{\sigma})}}{\sum_{j=1}^{n} P(T = O_j|H_{t-1}) \prod_{X_{t,a} \in A_t} \frac{exp(\frac{-d(X_{t,a},O_j)}{\sigma})}{\sum_{X_{t,k} \in D_t \setminus A_t \cup \{X_{t,a}\}} exp(\frac{-d(X_{t,k},O_j)}{\sigma})}}$$

1.3 Algoritmus

Označme P_t jako pole pravděpodobností pro každý objekt z databáze

$$P_t[i] = P(T = O_i|H_t), O_i \in S$$

Algorithm 1: Krok aktualizace pravděpodobností

Vstup: $P_{t-1}, likes, display, \sigma$
Výstup: P_t
res is an array of floats of size $|S|$ initialized with 1
for O_i ***in S*** **do**
    dispSum := 0
    for ***X in display*** **do**
        if *X not in likes* **then**
            dispSum := dispSum + exp(- dist(X, O_i) / σ)
    for ***L in likes*** **do**
        likeVal := exp(- dist(L, O_i) / σ)
        res[i] := res[i] * likeVal / (dispSum + likeVal)
    res[i] := res[i] * P_{t-1}[i]
scoresSum = 0
for O_i ***in S*** **do**
    scoresSum := scoresSum + res[i]
for O_i ***in S*** **do**
    res[i] := res[i] / scoresSum
return res

Tento krok aktualizace pravděpodobností můžeme opakovat do té doby, dokud uživatel nenajde hledaný snímek, nebo dokud to nevzdá. Tím dostaneme celý algoritmus hledání pomocí zpětné vazby.

Algorithm 2: Algoritmus hledání pomocí zpětné vazby

Vstup: P_0, σ
probabs = P_0
while *Target not found* **do**
    display = generateDisplay(probabs)
    likes = showDisplay(display) `// Show display and get feedback`
    probabilities = newProbs(probabs, likes, display, σ) `// Algorithm 1`

Algoritmy pro generování displeje budu probírat v kapitole 2.

1.4 Inicializace pravděpodobností

Nyní již máme algoritmus pro inkrementální výpočet pravděpodobností relevance. Pro úplnost nám již chybí jen hodnoty P_0 jako vstup do algoritmu 2.

1.4.1 Uniformní inicializace

Přímočarým řešením je přiřadit všem snímkům z databáze S stejnou pravděpodobnost a tím předpokládat, že apriori nemáme žádnou informaci o relevanci snímků.

$$\forall i \in \{1, \ldots, |S|\} : P(T = O_i) = \frac{1}{|S|}$$

Pokud ale máme již nějakou znalost o relevanci snímků v databázi, tak tento přístup se stává neefektivním, protože tuto informaci ignoruje.

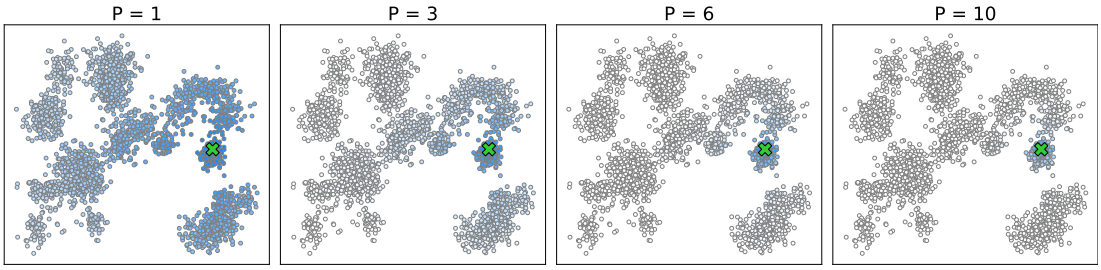
1.4.2 Inicializace výstupem jiného algoritmu

Informace o relevanci snímků může být i transformovaný výstup jiných algoritmů, například podobnost k barvám, pozice objektů nebo hledání podle klíčových slov. Obecně řečeno jakýkoliv algoritmus, kterým dokážeme efektivně ohodnotit relevanci každého snímku k dotazu, je možné použít pro potřeby inicializace.

Inicializace pomocí *query embeddingu* textového dotazu

Query embedding je technika mapování dotazu do společného prostoru charakteristických rysů. Tedy pro každý dotaz může být vygenerovan vektor v prostoru charakteristických rysů snímků z videa. Tento přístup používá i například metoda W2VV++, která mapuje textový dotaz do prostoru charakteristických rysů (Li a kol. (2019)).

Poté, co jsme obdrželi vektor charakteristických rysů, potřebujeme přiřadit bližším snímkům v prostoru vyšší pravděpodobnost. K tomu využijeme heuristiku, která pomocí vzdáleností v daném prostoru charakteristických rysů odhadne každému snímku pravděpodobnost relevance.



Obrázek 1.1: Ukázka inicializace pomocí *query embedding* v prostoru charakteristických rysů. Malé kroužky představují jeden snímek, čím modřejší tím vyšší pravděpodobnost relevance. Zelený křížek je namapovaný dotaz do prostoru charakteristických rysů. Pro účely vizualizace byla použita L1 vzdálenost.

Nechť q je textový dotaz popisující hledaný obrázek T. Uvažujme funkci f_t, která mapuje textové dotazy do stejného prostoru jako mapuje funkce f obrázky. Model W2VV++ trénuje f_t tak, aby nejbližší snímky odpovídaly co nejlépe danému textovému popisu. Díky těmto funkcím je možné ohodnotit každý obrázek O_i v databázi vzdáleností od dotazu

$$d'_i = 1 - cossim(f_t(q), f(O_i)).$$

Dále by se nám mohlo hodit nastavit "sílu" inicializace, a proto zavedeme parametr P, který bude sloužit k tomuto účelu. Následně zavedeme funkci

$$W(x) = exp(-x * P),$$

která vrátí váhu relevance daného snímku pro danou vzdálenost od dotazu.

Nakonec vytvoříme odhad pravděpodobnosti ještě před prvním displejem následovně:

$$P(T = O_i) \approx \frac{W(d'_i)}{\sum_{j=0}^{|S|} W(d'_j)}$$

Nyní zavedeme ještě intuitivní pohled na parametr P. Čím vyšší P, tím užší okolí vektoru dotazu bude mít vyšší pravděpodobnost relevance. Vizualizace této inicializace viz 1.4.2 pro různá nastavení P.

2. Výběr displeje

Poté, co jsme získali pravděpodobnosti relevance pro každý snímek z databáze, máme před sebou další úkol. Nyní nás bude zajímat, jak správně vybrat displej, který bude prezentován uživateli. Uživatel má za úkol vybrat z tohoto displeje podmnožinu nejrelevantnějších snímků, a poté pokračuje do další iterace. Dále budeme předpokládat v celé kapitole relaci $|S| \geq |D|$.

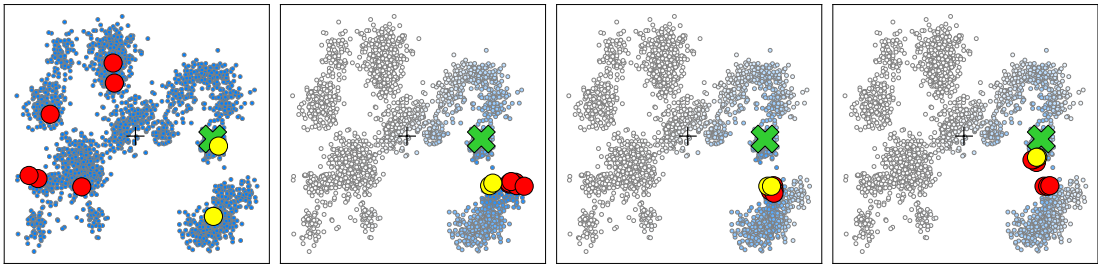
2.1 Nejpravděpodobnější displej

Prvním možným, a nejjednodušším, přístupem je vybrat $|D|$ nejpravděpodobnějších snímků z databáze.

Tímto přístupem obvykle získáme snímky, které si jsou velmi podobné. To může být pro uživatele nepříjemné. Nejenže uživateli může déle trvat výběr relevantního snímku, ale zároveň může dávat méně přesnou zpětnou vazbu. Zároveň v prostoru charakteristických rysů je displejem pokrytá jen malá část a zbytek prostoru, který by mohl být také relevantní pro uživatele, je zanedbán. Dále malá diverzita snímků na displeji může způsobit tzv. "přeučení", což je způsobeno větším počtem vybraných snímků v průběhu hledání, které nejsou příliš podobné hledanému snímku, i když to byly ty nejpodobnější z displeje.

Naopak při dobré zpětné vazbě může tento displej brzy vybrat hledaný snímek, protože ten již má vysokou pravděpodobnost relevance.

Vizualizace hledání a generování displeje touto metodou viz 2.1.

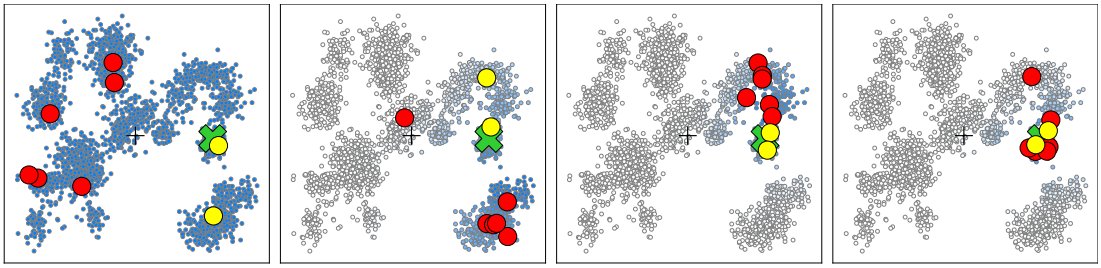


Obrázek 2.1: Ukázka hledání v prostoru charakteristických rysů pro metodu 2.1. Malé kroužky představují jeden snímek, čím modřejší tím mají vyšší relevanci. Zelený křížek označuje hledaný snímek. Velké kroužky označují snímky, z aktuálního displeje. Žluté jsou vybrané uživatelem jako relevantní a zbytek je červený. Pro účely vizualizace byla použita L1 vzdálenost.

2.2 Náhodný vážený displej z distribuce

Jedna z nevýhod nejpravděpodobnějšího displeje je nízká variabilita snímků na displeji. To může ztížit výběr relevantního snímku a mít za následek horší zpětnou vazbu od uživatele.

Tento přístup se snaží poskytnout uživateli lepší nadhled a tím získat kvalitnější zpětnou vazbu, která povede k rychlejšímu nalezení hledaného snímku.



Obrázek 2.2: Ukázka hledání v prostoru charakteristických rysů pro metodu 2.2. Malé kroužky představují jeden snímek, čím modřejší tím mají vyšší relevanci. Zelený křížek označuje hledaný snímek. Velké kroužky označují snímky, z aktuálního displeje. Žluté jsou vybrané uživatelem jako relevantní a zbytek je červený. Pro účely vizualizace byla použita L1 vzdálenost.

Displej D_t se vybere jako náhodný vážený výběr z distribuce $P(T = O_i|H_{t-1})$ bez opakování.

Viz ukázka hledání v prostoru charakteristických rysů a výběru displeje obrázek 2.2. Z této vizualizace je patrné, že takto vybrané displeje postupně konvergují k hledanému snímku.

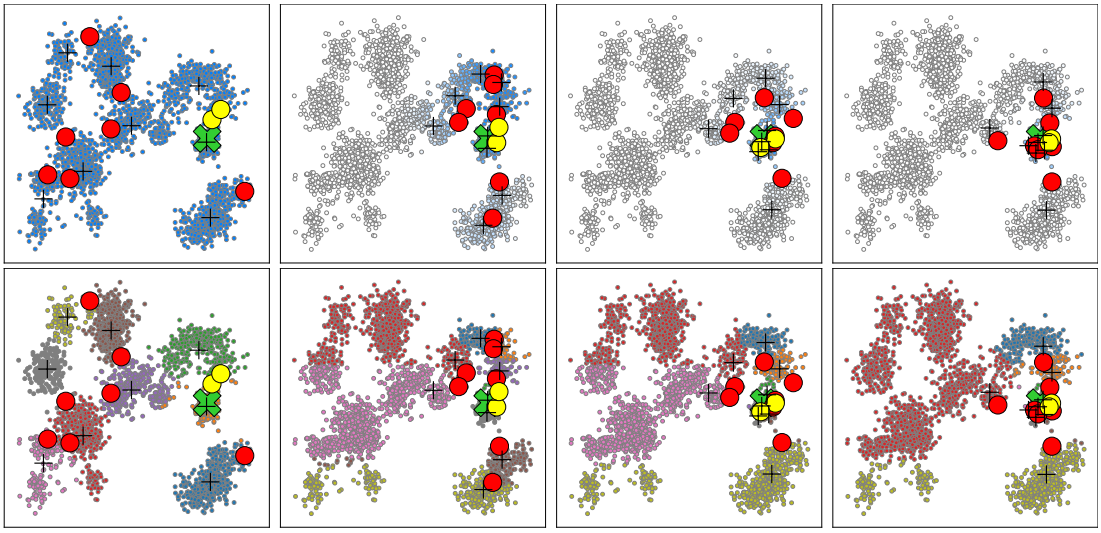
2.3 SOM displej

Nevýhoda předchozích metod je například v tom, že výběr snímků může uživatelům připadat nesystematický. Dále z neuspořádaného displeje je náročnější vybírat relevantní snímky a náhodný výběr také nemusí být vždy výhodný. To pak může ovlivnit i kvalitu zpětné vazby. Dále by se nám mohlo hodit systematičtější prohledávání daného prostoru charakteristických rysů, abychom lépe zacílili hledaný snímek. Tento problém se snaží řešit SOM displej, který využívá samoorganizujících se map (*self-organizing map*, Kohonen (1990)).

Samoorganizující se mapy jsou druhem neuronových sítí, které se řadí do třídy s označením kompetitivní nebo "bez dozoru" (*unsupervised*). Neurony této sítě se aktualizují postupně v iteracích, kdy jeden nebo množina vstupních vektorů aktualizuje váhy daného neuronu v závislosti na vzdálenosti ve vektorovém prostoru. Tyto vstupní vektory jsou v našem případě vektory charakteristických rysů pro každý vybraný snímek z videa. Existuje více strategií výběru vstupních dat pro aktualizaci sítě, kdy nejčastěji se využívá uniformní náhodný výběr z celé množiny učících dat. Tímto přístupem se síť snaží naučit topologii daného datasetu ve vektorovém prostoru R^n a pokryje jednotlivé shluky (*clusters*).

Pro naše účely můžeme využít vážený náhodný výběr z distribuce $P(T = O_i|H_t)$. Tento způsob výběru bude upřednostňovat relevantnější snímky. Díky tomu bude prostor, který je pro uživatele zajímavější, lépe pokryt neurony sítě a rozčleněn do shluků odrážející distribuci relevancí v prostoru. Tato metoda učení sítě je převzata ze systému SOMHunter (Kratochvíl a kol. (2020)).

Po učícím procesu sítě získáme jednotlivé shluky a pro každý shluk vybereme jednoho reprezentanta. Množina těchto reprezentantů představuje displej, který je zobrazen uživateli. Pro výběr jednoho reprezentanta ze shluku můžeme využít předchozí metody generování displejů. Buď vybereme ten snímek ze shluku, který má největší pravděpodobnost relevance, nebo vybereme náhodně váženě z



Obrázek 2.3: Ukázka hledání v prostoru charakteristických rysů pro metodu 2.3. Malé kroužky představují jeden snímek. V první řadě platí, že čím modřejší je výplň tím vyšší relevanci má snímek. V druhé řadě barva odpovídá příslušnosti do shluku. Zelený křížek označuje hledaný snímek. Velké kroužky označují snímky, z aktuálního displeje. Žluté jsou vybrané uživatelem jako relevantní a zbytek je červený. Černé znaky plus označují neuron sítě. Pro účely vizualizace byla použita L1 vzdálenost.

distribuce pravděpodobností relevance snímků daného shluku. Tyto metody výběru reprezentanta mají rozdílné vlastnosti a jejich efektivitu budeme zkoumat a porovnávat experimentálně.

V extrémním případě může být daný shluk prázdný. V tom případě pro každý prázdný shluk najdeme nejbližší jiný shluk, ze kterého je ještě možné vybrat reprezentanta, který ještě není na displeji. Takový shluk existuje, protože předpokládáme, že $|S| \geq |D|$. Poté nového reprezentanta vybereme z takto nalezeného shluku stejnou metodou, jako z neprázdného shluku s podmínkou navíc, že se nebude žádný snímek opakovat na displeji.

Ve vizualizaci 2.3 můžeme pozorovat, jak se síť postupně učí podle relevance a tím se neurony blíží k hledanému cíli. V druhé řadě můžeme také pozorovat, jak se mění shluky v průběhu učení. V tomto případě reprezentant jednoho shluku je vybrán jako ten nejpravděpodobnější.

3. Srovnání zpětnovazebních algoritmů

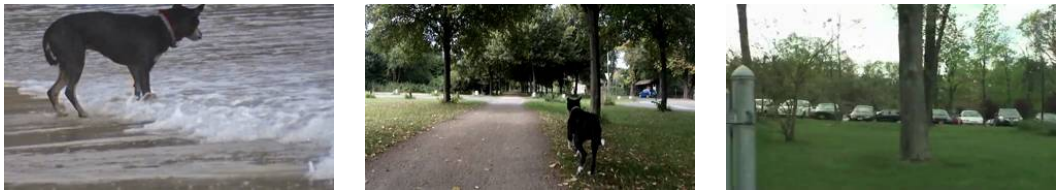
Představené algoritmy pro zpětnou vazbu nám dávají způsob řešení úloh, ve kterých hledáme známý objekt v multimediálních databázích. Jejich účinnost neumíme analyticky odhadnout, a proto ji ověříme nejprve pomocí automatizovaných simulací, kde budeme předpokládat více druhů uživatelů.

3.1 Simulovaní uživatelé

Automatické testování zpětnovazebních algoritmů není triviální úkol, protože rozhodnutí uživatele o tom, co je relevantní, je závislé na mnoha proměnných a často může být i subjektivně zkresleno. Navíc rozhodnutí uživatele může být učiněno na základě sémantických vlastností snímku, které nebyly zachyceny v charakteristických rysech, nad kterými je počítána vzdálenost. Díky tomu tato zpětná vazba může být i zavádějící. Při vyhledávání obrázků (*image retrieval*) se vyskytuje obdobný problém nazývající se sémantická mezera (*semantic gap*, Hare a kol. (2006)). V tomto problému jde o překročení mezery mezi nízkoúrovňovou reprezentací dat a vysoko-úrovňovým popisem zachycujícím sémantické vztahy. Naopak rozhodnutí uživatele může být učiněno na základě nízkoúrovňových vlastností snímku, které mohou být zanedbané nebo dokonce chybět v charakteristických rysech. Například je to podobnost barev, osvětlení či pozice hran. Pro ukázku daného problému viz obrázek 3.1.

První typ simulovaného uživatele je v literatuře označován jako tzv. *Ideální uživatel*. Tento uživatel vždy označuje ty snímky, které jsou podle charakteristických rysů nejpodobnější vzhledem ke hledanému obrázku T. Díky tomu je jeho rozhodnutí vždy deterministické. K lepšímu odhadu výkonu reálného uživatele potřebujeme nasimulovat i situace, kdy uživatel bude označovat jako relevantní i ty snímky, které jsou vzdálenější v prostoru charakteristických rysů.

Druhý typ simulovaného uživatele budeme označovat jako *Uživatel s řízeným výkonem* (*Performance controlled user*, zkr. *PCU*). Ten bude vybírat relevantní snímky s náhodným šumem, který bude simulovat nepřesnost reálného uživatele. Ve výzkumu Cox a kol. (2000) je obdobně simulován uživatel, který při výběru relevantního snímku zavádí nepřesnost. Nejprve se spočítá váha relevance snímku



Obrázek 3.1: Ukázka nejednoznačnosti označení relevantního snímku. Uprostřed je hledaný snímek a uživatel se má rozhodnout, zda je více relevantní levý nebo pravý snímek.

$R(X_{i,D})$ z displeje D následovně

$$R(X_{i,D}) = \frac{(\frac{1+cossim(X_{i,D},T)}{2})^E}{\sum_{j \in D}(\frac{1+cossim(X_{j,D},T)}{2})^E}, \tag{3.1}$$

kde:

- $R(X_{i,D})$: je váha relevance snímku $X_{i,D}$,
- T: je hledaný snímek,
- D: je zobrazený displej uživateli,
- $cossim(f(X_{i,D}), f(T))$: je kosínova podobnost charakteristických rysů mezi $X_{i,D}$ a T,
- E: je parametr uživatele (čím vyšší tím dává podobnější výsledky Ideálnímu uživateli).

Poté se podle těchto vah náhodně váženě bez opakování vybere požadovaný počet snímků. Tento uživatel nám pomůže lépe odhadnout výkon reálného uživatele a optimální konfiguraci.

3.1.1 Srovnání simulovaného a reálného uživatele

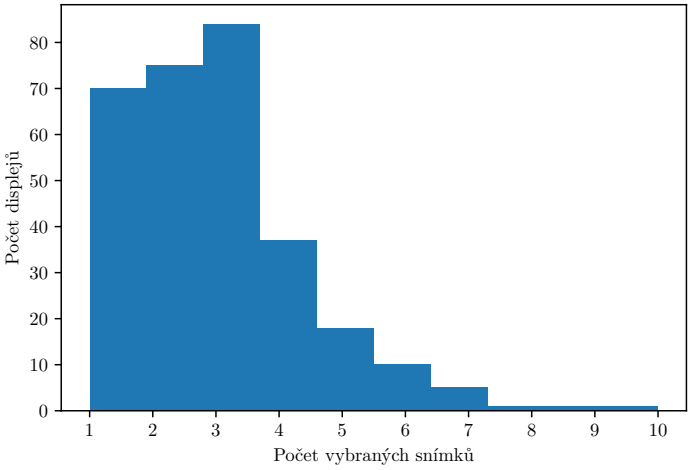
Nyní potřebujeme ověřit, že výše popsaný algoritmus automatického výběru snímků aproximuje efektivitu reálného uživatele. K ověření byl použit software, který vyšel z open source nástroje SOMHunter[1].

Uživatel mohl začít inicializací klíčovými slovy a následně zadával zpětnou vazbu. Dále si mohl zvolit typ displeje, ze kterého provede výběr relevantních snímků. Na výběr měl dva druhy displejů - nejpravděpodobnější a SOM displej. Průzkumu se účastnili dva uživatelé. První poskytl 105 anotací displeje a druhý 197. Z displeje mohl být vybrán libovolný počet snímků. Průměrný počet činil 2,75 snímků a jejich četnost je zobrazena v obrázku 3.2.

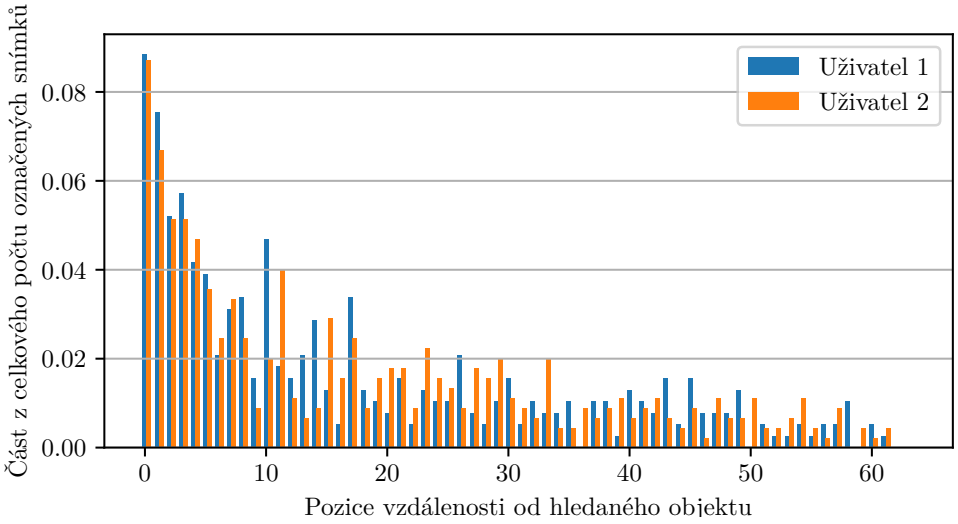
Zachycení vztahu mezi prostorem charakteristických rysů, displejem a výběrem relevantních snímků uživatele provedeme pomocí histogramů přes všechny displeje s označením relevantních snímků. Oba respondenty můžeme takto srovnat viz 3.3. Z tohoto grafu je patrné, že výběry dvou různých respondentů jsou navzájem podobné. Na ose x je pořadí snímku z displeje seřazené podle vzdálenosti k hledanému snímku. Na ose y je část snímků ze všech označených, které byly na dané pozici. Tři nejbližší snímky tvoří 21,6% ze všech označených snímků. Dále více jak polovina všech označených snímků byla do pozice 11. Z toho je zřejmé, že vzdálenost v prostoru charakteristických rysů a vzdálenost vnímána lidským uživatelem je podobná.

Nyní potřebujeme optimalizovat parametr E ze vzorce 3.1 tak, aby se takto simulovaný PCU uživatel choval co nejpodobněji jako ten reálný. K tomu použijeme histogramy probírané v odstavci výše, kde srovnáme výběr simulovaného PCU uživatele a jednotlivého reálného uživatele. Srovnání bude probíhat na stejných displejích, ze kterých vybíral i reálný uživatel. Bude se vybírat i stejný počet snímků. Mezi každým takto vygenerovaným histogramem pro simulovaného

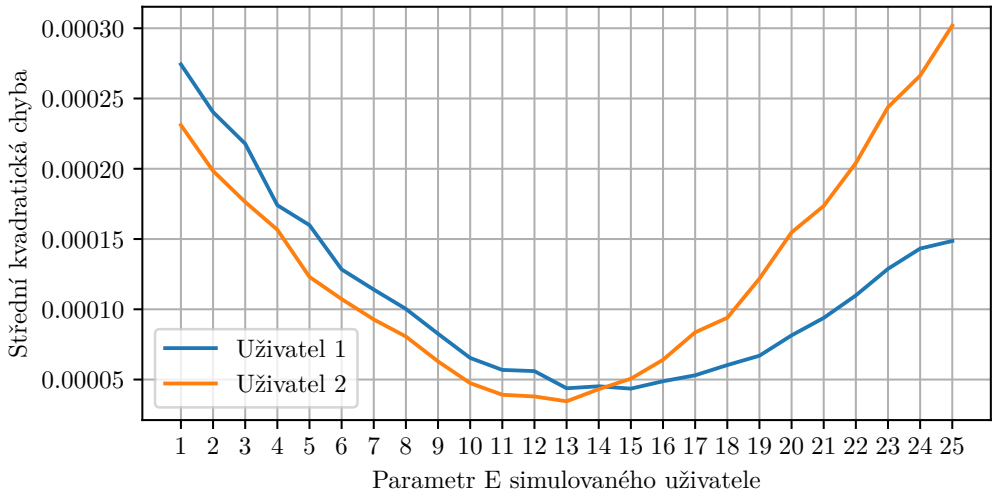
[1] https://github.com/siret/somhunter



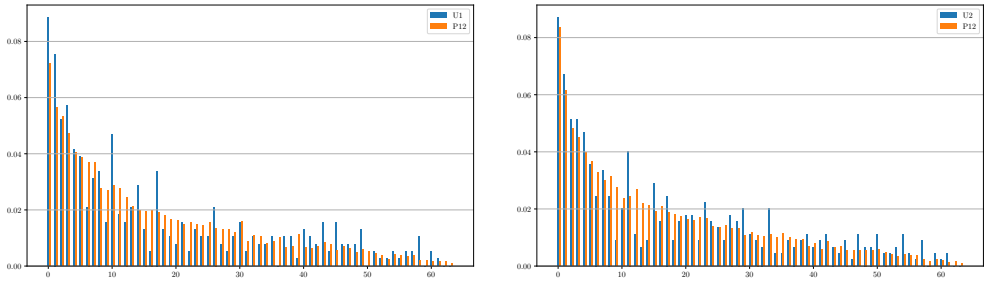
Obrázek 3.2: Histogram počtu displejů v závislosti počtu vybraných snímků z displeje.



Obrázek 3.3: Histogram počtu označených snímků jako relevantní v závislosti na pořadí podle vzdálenosti.



Obrázek 3.4: Střední kvadratická chyba histogramů reálného uživatele a simulovaného PCU uživatele s parametrem E.



Obrázek 3.5: Histogramy počtu označených snímků v závislosti na pořadí podle vzdálenosti. Vlevo je porovnaný výběr prvního respondenta se Simulovaným PCU uživatelem a vpravo je porovnaný výběr druhého respondenta se Simulovaným PCU uživatelem.

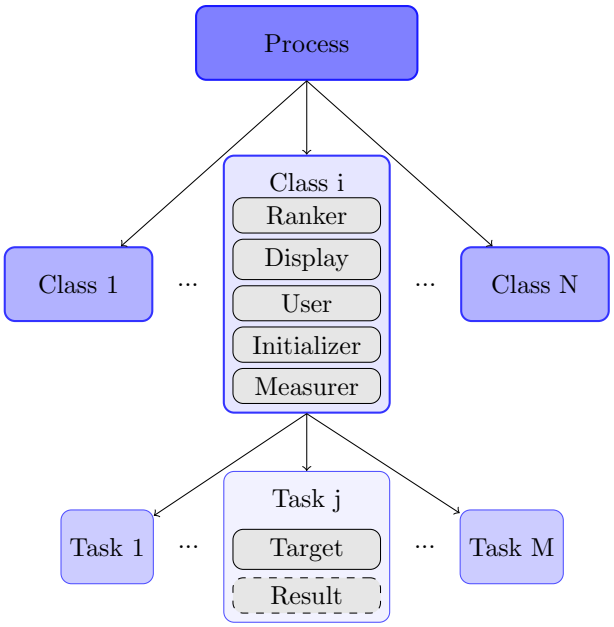
uživatele s parametrem E a histogramem reálného uživatele spočítáme střední čtvercovou odchylku. Výsledkem je graf 3.4. Z něho je patrné, že simulovaní PCU uživatelé s jednou z nejmenších odchylek k reálnému uživateli jsou při nastavení parametru $E \in [11, 15]$.

V dalších experimentech budeme nazývat Simulovaný PCU uživatel toho uživatele s parametrem $E = 12$. Histogramy výběrů pro oba respondenty porovnané se Simulovaným PCU uživatelem jsou znázorněny v grafu 3.5.

3.2 Software pro automatické vyhodnocení

Jak jsem již nastínil výše, vyhodnocení účinnosti zpětnovazebních učících algoritmů musíme vyhodnotit experimentálně. Pro tyto účely jsem vytvořil software RFTester[2]. Pravděpodobnosti, se kterými se počítalo v kapitolách 1 a 2 budu dále označovat jako skóre nebo skóre relevancí, abychom zmírnili požadavky. Jediné, co budeme předpokládat je, že pro každé skóre s bude platit $0 \leq s \leq 1$.

[2]https://gitlab.mff.cuni.cz/veselp/rftester



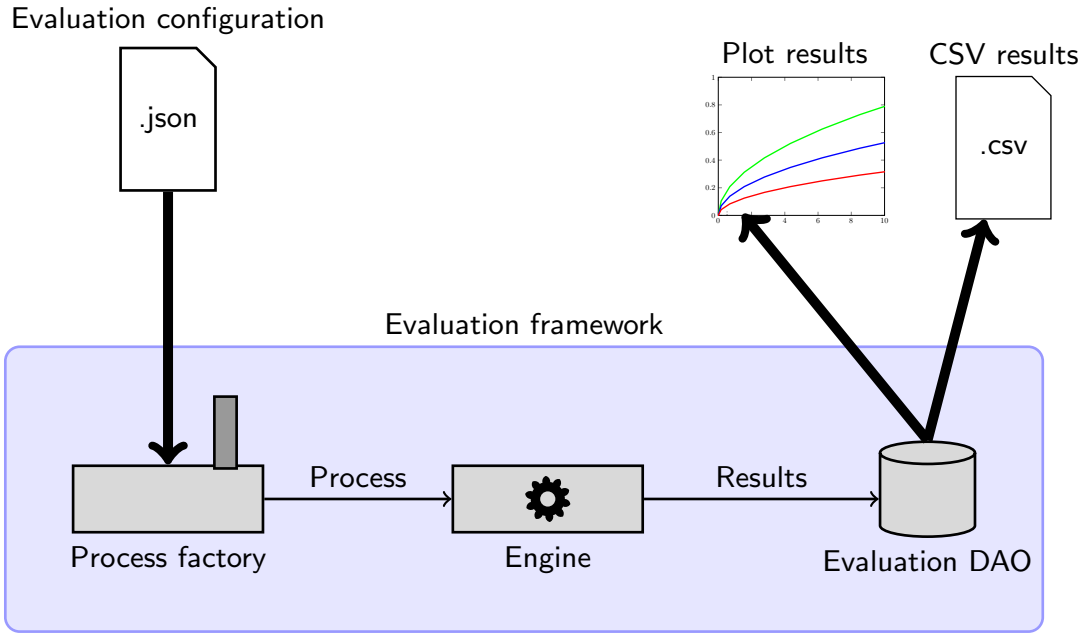
Obrázek 3.6: Hierarchie struktur pro vyhodnocení komponent zpětnovazebního učení.

Každý experiment se skládá z několika komponent, které mají vliv na celkový výsledek:

- *Ranker* je komponenta, která aplikuje zpětnou vazbu uživatele a aktualizuje skóre relevancí. Obecně implementuje algoritmus 2 nebo jeho odvozené verze.
- *Display generator* má za úkol vybrat množinu snímků, které se zobrazí simulovanému uživateli. Tyto snímky jsou vybrány na základě skóre a jejich výběr je popsán v kapitole 2.
- *Initializer* připravuje skóre relevancí ještě před prvním displejem. Implementuje tedy algoritmy popsané v 1.4.2.
- *Simulated user* je komponenta, která z displeje vybere vždy určitý počet relevantních snímků pro zpětnou vazbu. Zde se implementují uživatelé diskutovaní v sekci 3.1.
- *Measurer* implementuje požadovanou vzdálenostní funkci nad daným vektorovým prostorem charakteristických rysů.

Experimenty probíhají po dávkách, které nazývám *procesy* (*process*). Každý proces se skládá z takzvaných *tříd* (*class*), které mají již konkrétní komponenty a jejich parametry. Následně tyto třídy se skládají z *úkolů* (*task*), což jsou jednotlivá hledání, které nejenže znají komponenty hledání ze své třídy, ale zároveň mají i svůj cíl, co mají hledat. Viz 3.6.

Konfigurace simulací jsou předány systému ve formátu JSON. Tento soubor je zpracován v *Process factory* (*EvaluationProcessFactory*), kde se připraví vnitřní struktury pro evaluaci. Název každé komponenty musí odpovídat právě jedné C#



Obrázek 3.7: Abstraktní architektura SW.

třídě, která je při běhu k dispozici a implementuje požadované rozhraní komponenty. Dále takto vytvořený proces je předán k zpracování do *Engine*, který paralelně zpracovává jednotlivé *úkoly* (*Tasks*). Poté, co jsou všechny *úkoly* zpracované, jsou předány do *Evaluation DAO*, který se stará o persistenci dat a poskytuje je k následnému zpracování ve formě grafů nebo CSV souborů. Viz 3.7.

Hledané snímky budou vybrány z interního benchmarku naší výzkumné skupiny. Tyto snímky budou v rámci jednoho evaluačního procesu stejné. Při inicializaci klíčovými slovy je první displej výběr nejrelevantnějších snímků. Při uniformní inicializaci je vybrán pevně daný displej, který je pro všechny evaluační úkoly stejný.

3.3 Výsledky automatických testů

Úspěšnost zpětnovazebních učících technik v hledání známého objektu může záviset na více proměnných, jak jsme již probrali výše, a proto nejprve budeme muset odhadnout možné parametry a spustit evaluační framework. Toto první hledání parametrů může vygenerovat velké množství konfigurací, které musí být otestovány, a proto omezíme počet hledaných snímků za účelem zrychlení. Následně, až získáme hrubý odhad, můžeme omezit konfigurace jen na ty, u kterých byl jasný trend větší úspěšnosti než u ostatních.

3.3.1 Úvodní nalezení parametrů

V úvodním hledání budeme zkoumat, jaký má vliv na úspěšnost typ uživatele, typ displeje, různé hodnoty σ u *rankeru* a různé druhy inicializace. Tato inicializace může být uniformní nebo pomocí klíčových slov a jejich *W2VV++ query embeddingu* s různou hodnotou parametru P. Celkem se bude hledat 40 snímků z benchmarku, které budou pro všechny konfigurace stejné.

Úspěšnost budeme vyhodnocovat pomocí grafů počtu vyřešených úloh v závislosti na pořadí displeje. Na ose y bude část vyřešených úloh (1.0 = všechny úlohy vyřešeny) do displeje na ose x. Dále budeme sledovat, kdy uživatel nalezl libovolný snímek z hledané scény nebo z hledaného videa. Hledaná scéna je úsek videa, který je oddělen střihem a který obsahuje hledaný snímek. Hledané video je video, které obsahuje hledaný snímek. Naším úkolem bude najít konfiguraci, která stabilně bude mít nejlepší úspěšnost na počet displejů.

Úspěšnost ideálního uživatele

Výsledky testů jsou vyobrazeny v grafech 3.8, 3.9 a 3.10, kde v grafu 3.8 je úspěšnost hledání právě hledaného snímku. Graf 3.9 zobrazuje úspěšnost hledání libovolného snímku z hledané scény. Dále v grafu 3.10 je úspěšnost hledání libovolného snímku z hledaného videa.

Jako první dominantní jev můžeme pozorovat chování parametru sigma u *Rankeru*. Bez ohledu na displej a druh inicializace je nejlepší testovaná konfigurace parametru $\sigma = 0.01$. Dokonce při libovolné konfiguraci s tímto parametrem bylo vždy do čtvrtého displeje vyřešeno 100% úloh. Trend ukazuje, že čím je bližší sigma k nule, tím má ideální uživatel vyšší úspěšnost.

Typ displeje nehrál významnou roli v těchto simulacích. U nejúspěšnější konfigurace parametru σ bylo vše nalezeno do čtvrtého displeje bez ohledu na použitou metodu generování displeje. Lehké zlepšení oproti ostatním je možné pozorovat u metody generování displeje výběrem nejpravděpodobnějších snímků. Ta, na rozdíl od ostatních metod, dokázala najít vše dokonce do třetího displeje. Tedy konfigurace s parametrem $\sigma = 0{,}01$ a generováním displeje nejrelevantnějšími snímky je optimální nastavení pro vyřešení všech úkolů na co nejméně displejů.

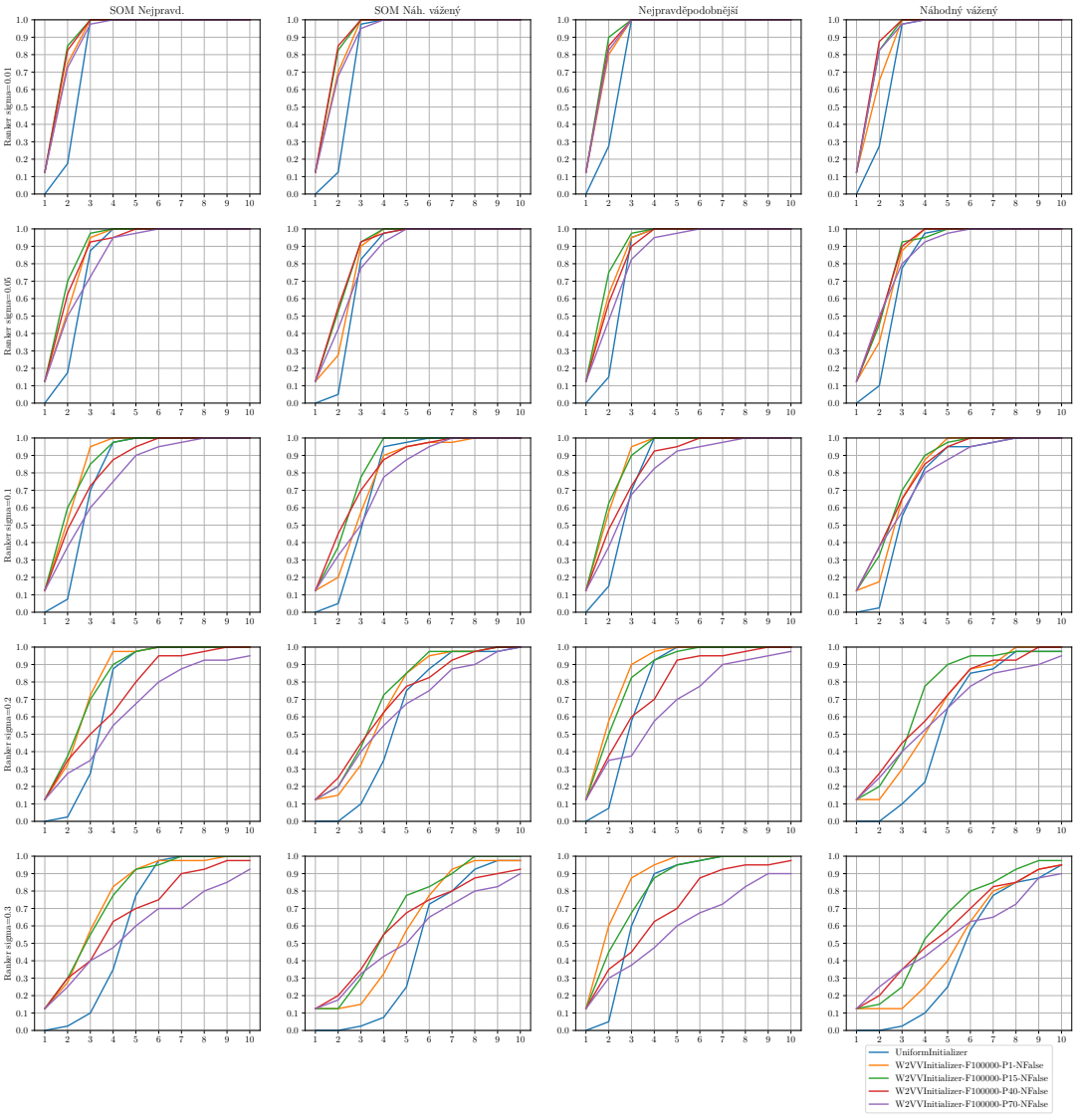
Inicializace klíčovými slovy pro konfiguraci *Rankeru* s parametrem $\sigma = 0{,}01$ velmi pomohla na prvních dvou displejích. Na druhém displeji je možné pozorovat nárůst úspěšnosti hledání až o 70 procentních bodů s inicializací klíčovými slovy oproti uniformní inicializaci. Od třetího displeje jsou rozdíly minimální nebo žádné.

Úspěšnost nalezení hledané scény se nijak výrazně neliší od úspěšnosti nalezení hledaného snímku. Úspěšnost nalezení hledaného videa je vyšší oproti úspěšností hledaného snímku. Především u konfigurací s inicializací klíčovými slovy a s parametrem *Rankeru* $\sigma = 0.01$ se úspěšnost nalezení videa na druhém displeji pohybuje až k 97,5%.

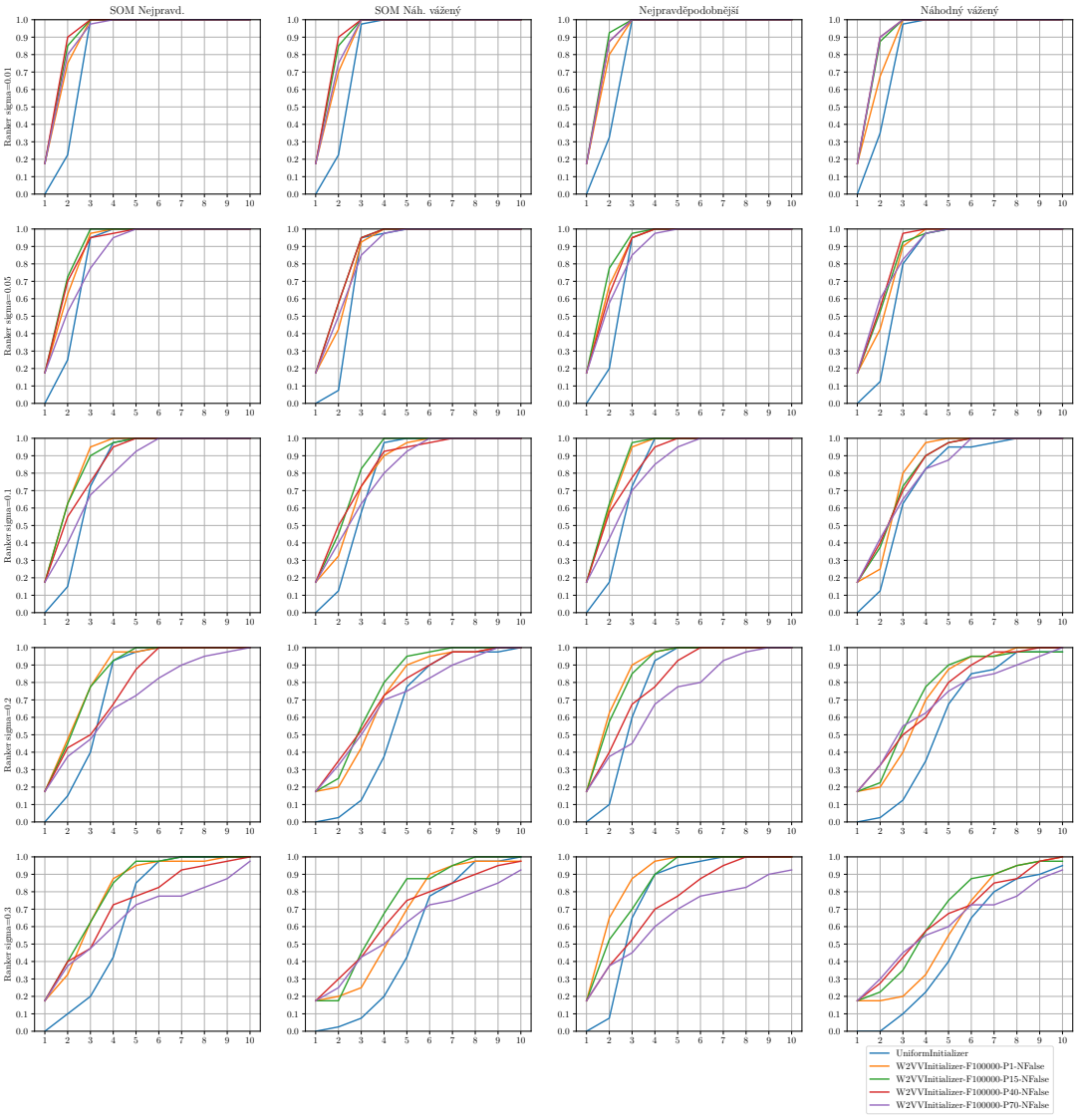
Úspěšnost Simulovaného PCU uživatele

Obdobně, jako v předchozí části, máme rozčleněné výsledky do tří kategorií. První grafy 3.11 ukazují úspěšnost hledání snímku, grafy 3.12 ukazují úspěšnost hledání libovolného snímku z hledané scény a grafy 3.13 ukazují úspěšnost hledání libovolného snímku z videa.

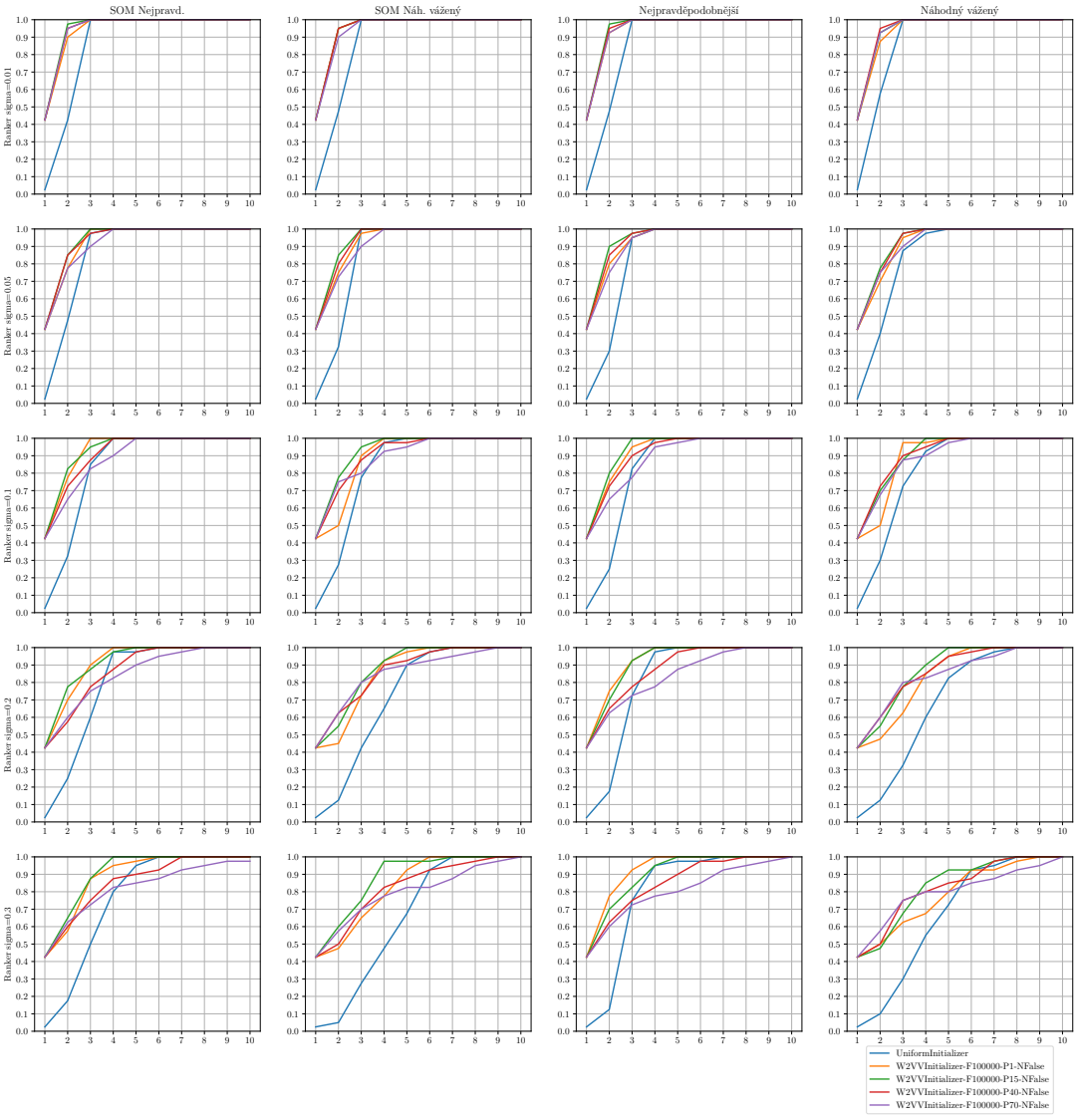
Hlavní faktor, který ovlivnil úspěšnost, byl parametr sigma. Na rozdíl od předchozího uživatele, optimální hodnota vyšla $\sigma = 0.1$. Optimální nastavení tohoto parametru ideálního uživatele vyšlo u nově testovaného uživatele s nejnižší úspěšností. Ukazuje se, že optimální nastavení tohoto parametru je závislé i na daném typu uživatele. Toto nastavení může být závislé i na dalších proměnných jako například typ hledaného snímku. Proto hledáme optimální nastavení pro průměrný



Obrázek 3.8: Část nalezených snímků do displeje na ose x. Zpětnou vazbu dával ideální uživatel.



Obrázek 3.9: Část nalezených scén, podle libovolného snímku ze scény, do displeje na ose x. Zpětnou vazbu dával ideální uživatel.



Obrázek 3.10: Část nalezených videí, podle libovolného snímku z videa, do displeje na ose x. Zpětnou vazbu dával ideální uživatel.

případ.

Obdobně, jako u předchozího uživatele, i tady nehrál velký rozdíl způsob výběru displeje. Můžeme porovnat i přesná čísla úspěšností z tabulek 3.14 a zjistíme průměrný vliv displejů na úspěšnost pro optimální parametr sigma přes všechny druhy inicializace. Z toho je vidět, že v různých stádiích hledání a pro různé cíle (přímo hledaný snímek/libovolný snímek ze scény) byla vždy optimální metoda výběru různá. Nejméně účinný byl displej Náhodný vážený, který ve všech čtyřech tabulkách má nejnižší průměr.

Typ inicializace se u tohoto uživatele ukázal být významným faktorem ovlivňující úspěšnost nalezení známého snímku. Zejména ze začátku hledání dosahuje tento způsob výrazný nárůst úspěšnosti nalezení hledaného snímku. V některých případech to je o více než 20 procentních bodů. I v pokročilých stádiích hledání má tato inicializace lepší úspěšnost než uniformní inicializace. Optimální hodnota parametru P z těchto výsledku vychází v rozsahu 15 až 40.

Celková úspěšnost hledání u displeje 10 pro optimální konfigurace dosahuje kolem 80-90% vyřešených úloh, což je výrazný rozdíl oproti ideálnímu uživateli, který v optimální konfiguraci byl schopen vyřešit vše do displeje 3.

3.4 Optimalizace parametru a ověření výsledku

Nyní již víme, že Ideální uživatel dokáže pro nejefektivnější testované konfigurace s $\sigma = 0{,}01$ nalézt vše do třetího displeje a Simulovaný PCU dosahuje nejlepšího výkonu s parametrem $\sigma = 0{,}1$ a inicializací klíčovými slovy. Dále víme, že ideální hodnoty parametru P pro inicializaci klíčovými slovy se pohybují v rozmezí 15 až 40. V této sekci nalezneme optimum pro inicializaci klíčovými slovy a ověříme rozdíly v generování displejů pro Simulovaného PCU uživatele.

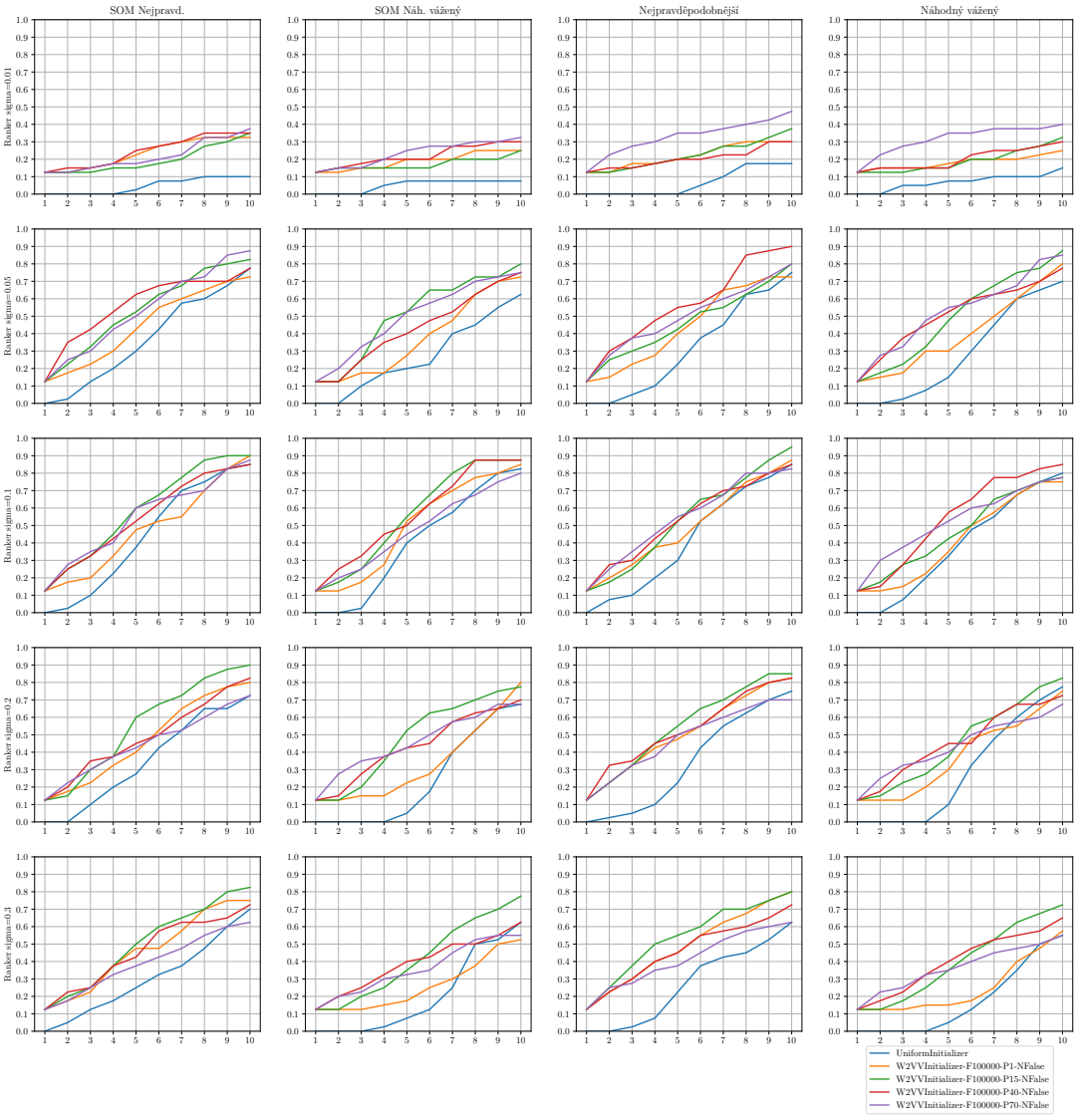
Pro tyto účely bude každá konfigurace hledat 100 snímků z benchmarku, každý dvakrát k získání stabilnějších výsledků. Stejně jako v předchozí sekci, tak i zde jsou počáteční displeje a cílové snímky stejné pro všechny konfigurace.

Výsledky evaluací jsou zobrazeny v grafech 3.15, 3.16 a 3.17. Pořadí celku grafů je stejné jako v předchozí části a tedy, 3.15 zobrazuje úspěšnost nalezení hledaného snímku, 3.16 zobrazuje úspěšnost nalezení libovolného snímku ze scény a 3.17 zobrazuje úspěšnost nalezení libovolného snímku z hledaného videa.

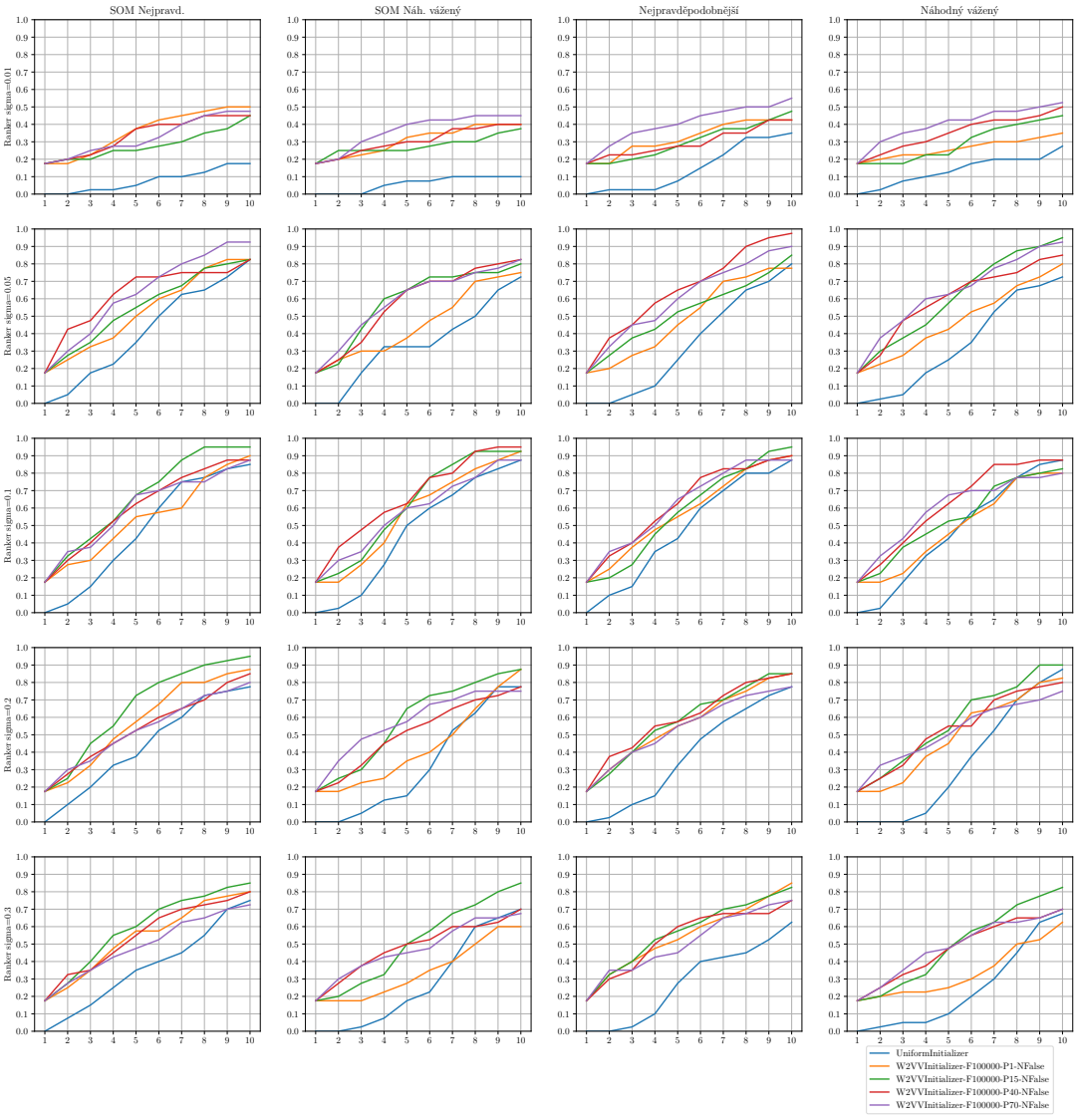
Jak lze pozorovat, rozdíly v parametru $P \in [10{,}40]$ na úspěšnost nalezení snímku, scény i videa jsou minimální a tedy můžou být zvoleny i na základě uživatelovi preference.

Dále efektivita displejů pro $P \in [20, 35]$ je téměř totožná. Rozdíl je pozorovatelný u $P \in [10, 15]$, kdy z počátku Náhodný vážený a SOM Náhodný vážený jsou méně efektivní. V pozdějších fázích hledání tyto displeje jsou nakonec stejně efektivní jako zbylé dva. Dokonce v případě $P = 10$ jsou displeje Náhodný vážený a SOM Náhodný vážený dosahují u displeje číslo 10 mírně lepší úspěšnosti než Nejpravděpodobnější a SOM nejpravděpodobnější.

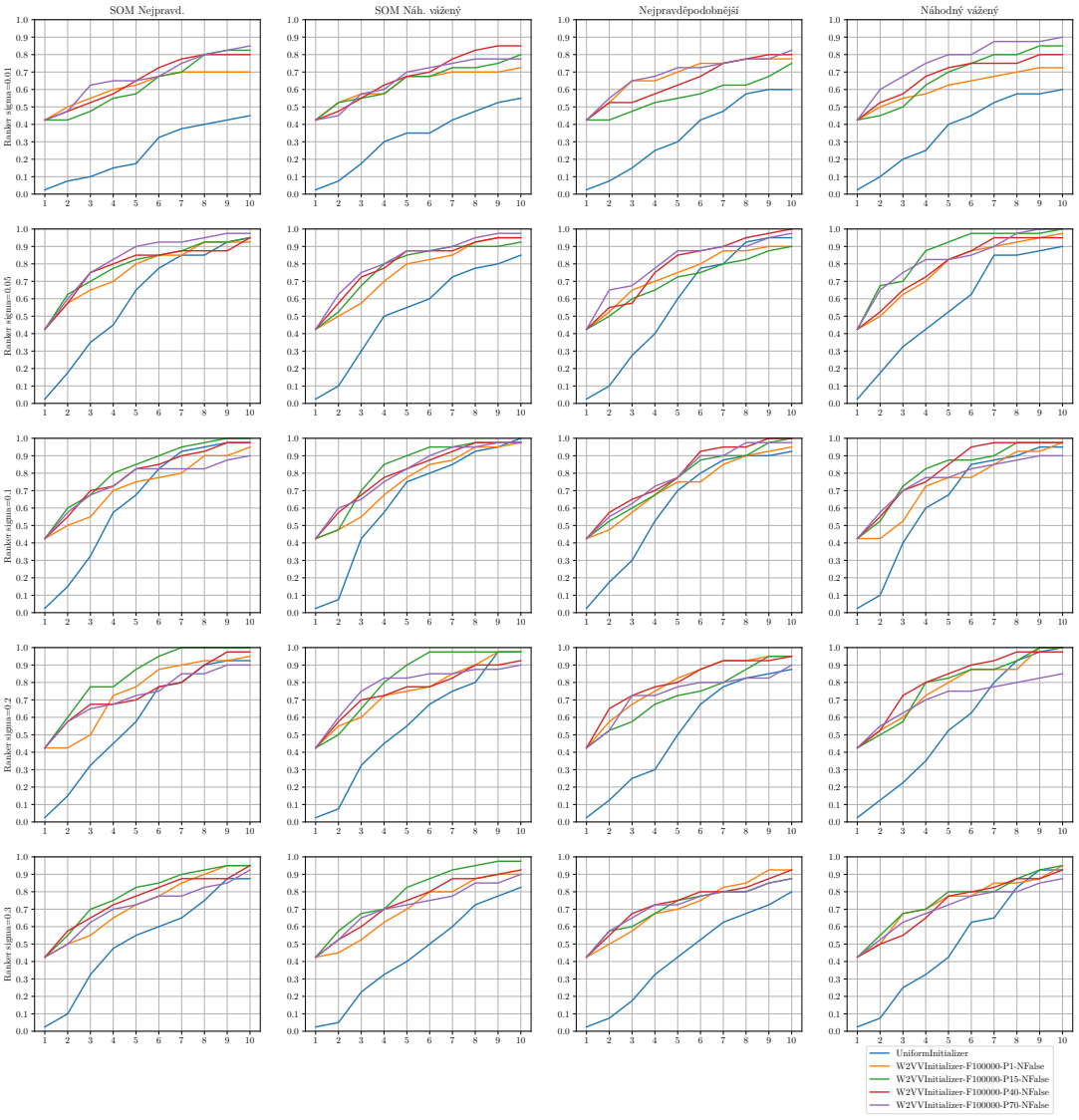
Zajímavý ukazatel je také úspěšnost nalezení libovolného snímku z videa. Při libovolné z optimálních konfigurací se tato úspěšnost pohybuje kolem 95% na desátém displeji.



Obrázek 3.11: Část nalezených snímků do displeje na ose x. Zpětnou vazbu dával Simulovaný PCU uživatel.

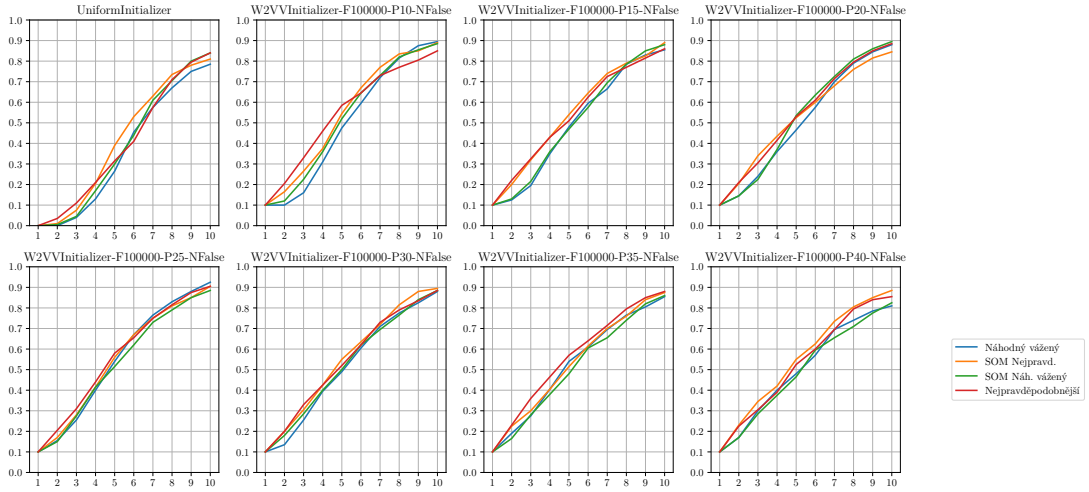


Obrázek 3.12: Část nalezených scén, podle libovolného snímku ze scény, do displeje na ose x. Zpětnou vazbu dával Simulovaný PCU uživatel.

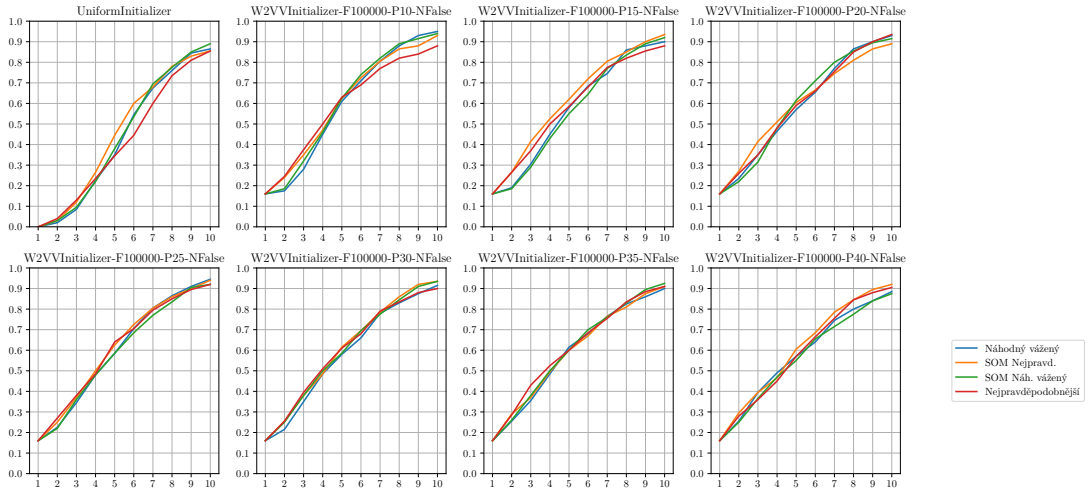


Obrázek 3.13: Část nalezených videí, podle libovolného snímku z videa, do displeje na ose x. Zpětnou vazbu dával Simulovaný PCU uživatel.

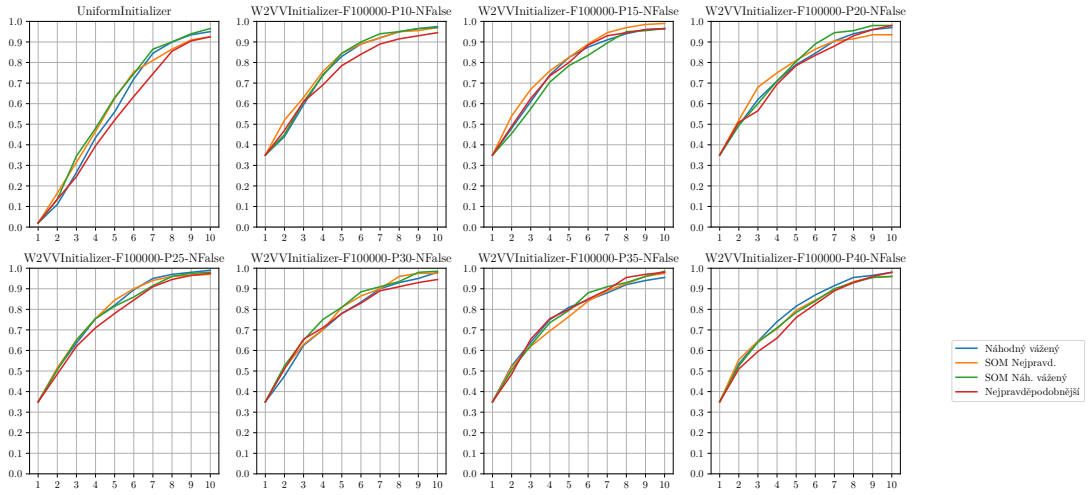
Obrázek 3.14: Úspěšnost hledání do displeje 5 a 10 pro *Ranker* s parametrem $\sigma = 0{,}1$. Horní dvě tabulky představují úspěšnost hledání daného snímku. Spodní dvě tabulky představují úspěšnost hledání libovolného snímku z hledané scény. Levé tabulky představují úspěšnost do displeje 5 a pravé představují úspěšnost do displeje 10.



Obrázek 3.15: Část nalezených snímků do displeje na ose x pro různé parametry inicializace podle klíčových slov. Zpětnou vazbu dával Simulovaný PCU uživatel.



Obrázek 3.16: Část nalezených scén, podle libovolného snímku ze scény, do displeje na ose x pro různé parametry inicializace podle klíčových slov. Zpětnou vazbu dával Simulovaný PCU uživatel.



Obrázek 3.17: Část nalezených videí, podle libovolného snímku ze videa, do displeje na ose x pro různé parametry inicializace podle klíčových slov. Zpětnou vazbu dával Simulovaný PCU uživatel.

4. SOMHunter na Video Browser Showdown

V rámci této kapitoly se budu věnovat integraci modelů zpětnovazebního učení do systému SOMHunter a přínosu těchto modelů k výkonu vítězného nástroje SOMHunter (Kratochvíl a kol. (2020)) na Video Browser Showdown 2020 v Daejeon, Korea.

4.1 Integrace zpětnovazebního učení

V tomto nástroji byl implementován algoritmus pro inicializaci klíčovými slovy modelu W2VV++ (Li a kol. (2019)). Dále obsahoval algoritmus zpětnovazebního učení, který byl popsán v kapitole 1. Jako poslední možnost dotazu umožňoval nástroj vybrat vzorový obrázek a zobrazit k němu nejpodobnější objekty z databáze. K tomu využíval charakteristické rysy z modelu W2VV++.

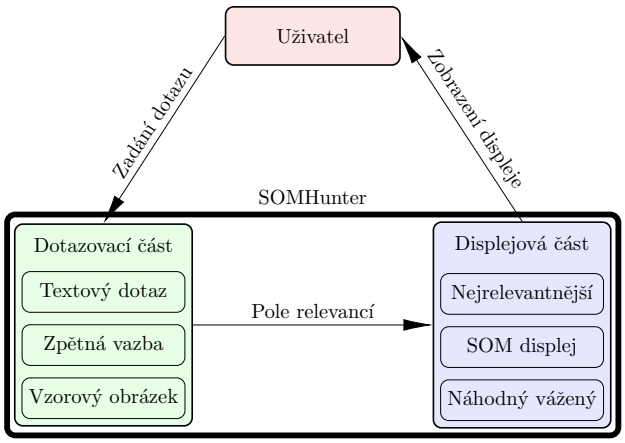
Uživatel si mohl vybrat mezi několika druhy displejů. První byl SOM displej, který vybíral reprezentanta náhodně váženě z vytvořených shluků. Druhý byl seznam seřazený podle skóre relevancí. Poslední byl náhodný vážený displej.

Diagram 4.1 znázorňuje postup hledání pomocí systému SOMHunter. Zde je vidět, že zpětnovazební učení je jeden z druhů dotazování, který má vliv na celkové pole relevancí. Z tohoto pole relevancí se následně generují displeje.

4.2 Nastavení Video Browser Showdown

Osobně jsem se účastnil soutěže Video Browser Showdown 2020 jako jeden z autorů systému SOMHunter (Kratochvíl a kol. (2020)). Video Browser Showdown 2020 probíhal v rámci konference MMM2020[1], která se konala v Daejeonu v Korejské republice. Několik týmů ve stejný čas hledá hledaný segment videa,

[1] http://www.mmm2020.kr/



Obrázek 4.1: Diagram interaktivního hledání v systému SOMHunter.

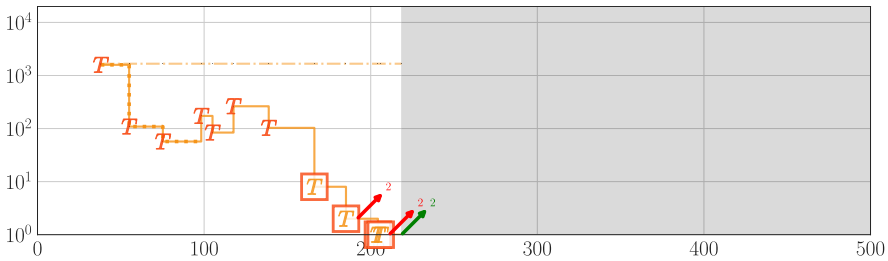
Obrázek 4.2: Foto našeho týmu na Video Browser Showdown.

který je promítán dataprojektorem všem účastníkům soutěže. Přikládám snímek 4.2, který ukazuje jeden z soutěžních dní. Na fotografii jsem já a můj kolega František Mejzlík.

Celkem se plnilo 22 úkolů. Z toho 10 hledaných scén bylo hledáno dvěma expertními uživateli a byly zadány jako textový popis s časovým limitem 8 minut. Dalších 6 scén bylo hledáno rovněž dvěma expertními uživateli a byly zadány vizuálně, tedy hledané scény byly promítnuty všem soutěžícím s časovým limitem 5 minut. Posledních 6 scén bylo hledáno dvěma uživateli, kteří poprvé v životě použili příslušný nástroj. Každý z nich hledal 3 scény. Tyto úlohy byly zadány vizuálně s časovým limitem 5 minut.

4.3 Výkon systému

V této sekci shrnu celkový výkon systému SOMHunter. Nejprve shrnu úspěšnost nalezení hledaného úseku videa, poté vysvětlím způsob analýzy výkonu, kde ukážu detailní analýzu vybraného hledání a agregovaně shrnu efektivitu jednotlivých komponent systému.

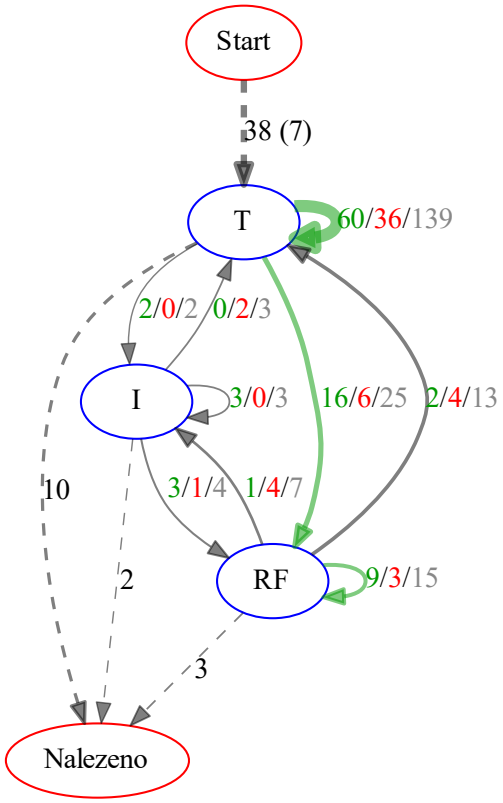


Obrázek 4.3: Pořadí snímku a videa od začátku úlohy. Na ose x je čas od začátku úkolu a na ose y je pozice hledaného objektu v databázi po seřazení podle skóre relevancí. Plná linka označuje pořadí videa a tečkovaná označuje pořadí scény. Značka T označuje, že na daném uspořádání měl vliv textový dotaz. Čtverec označuje, že byla použita zpětná vazba. Červená značka představuje změnu dotazu. Červená šipka označuje odeslání chybného snímku, zelená označuje nalezení hledaného snímku. Upravená verze grafu z Lokoč a kol. (2020a).

Expertní tým SOMHunteru vyřešil 80% textových úkolů, kde průměr přes všechny týmy byl 54%, a 83% vizuálních úkolů, kde průměr byl 55%. Nováčkové, kteří používali tento nástroj, vyřešili 33% úloh a průměr přes všechny týmy byl 18%. Po celkovém bodování se SOMHunter umístil na prvním místě a tím porazil zbylých devět týmů. Pro detailnější analýzu výsledků i dalších týmů viz "Is the Reign of Interactive Search Eternal? Findings from the Video Browser Showdown 2020".

Analýza výkonu interaktivních systémů pro vyhledávání není triviální úloha. Výkon se musí vyhodnocovat zpětně po evaluaci s reálnými uživateli a systém musí zaznamenávat k tomu potřebná metadata. Proto nástroj SOMHunter zaznamenával při každé změně dotazu část nejrelevantnějších snímků a též to, jaké druhy dotazů měly vliv na dané uspořádání. Po soutěži bylo tedy možné zrekonstruovat časovou osu s pozicí hledané scény a videa při seřazení podle skóre relevancí. Viz ukázka 4.3, kde uživatel začal textovým dotazem, který několikrát upravil, a poté použil zpětnou vazbu. Tato zpětná vazba dostala snímek mezi top 10 snímků z databáze seřazených podle skóre relevancí, což se textovým dotazem nepodařilo. Po dvou pokusech byl nakonec nalezen a odeslán ten správný snímek na server VBS.

Dále můžeme agregovaně sledovat vliv různých reformulací dotazů pomocí přechodového diagramu 4.4, kde je zachycena každá reformulace a její vliv na pořadí hledaného videa. Každé hledání začíná ve vrcholu "Start". Pokud byla hledaná scéna nalezena, tak z posledního dotazu vede hrana do vrcholu "Nalezeno". Vrchol "T" označuje dotaz klíčovými slovy, "I" dotaz vzorovým obrázkem a "RF" dotaz za použití zpětné vazby. Většina úloh byla vyřešena pomocí dotazu klíčovými slovy, ale zpětná vazba také obecně pomáhala a vedla k vyřešení některých úloh.



Obrázek 4.4: Změna pořadí videa po reformulaci dotazu. Vrcholy označují typ dotazu a hrany označují reformulaci. Trojice čísel zachycuje kolikrát se pořadí hledaného videa zlepšilo/zhoršilo/neurčeno. Upravená verze grafu z Lokoč a kol. (2020a).

Závěr

V této práci jsem se zabýval interaktivními metodami vyhledávání ve videích. Hlavní metoda byla použita z výzkumu Cox a kol. (2000). Tato metoda využívá zpětné vazby a metody bayesovského učení pro odhad relevance každého snímku z databáze videí. Dále jsem porovnával různé strategie pro generování displejů, ze kterých uživatel může dávat zpětnou vazbu systému. Porovnal jsem metody z původního výzkumu s novými metodami, které využívaly samoorganizující se mapy (Kohonen (1990)).

Evaluace efektivity interaktivních nástrojů pro vyhledávání je nelehký úkol. Obvykle jsou k tomu třeba evaluace s reálnými uživateli, což může být nepraktické. Za tímto účelem se každoročně uskutečňují evaluační kampaně jako Video Browser Showdown nebo Lifelog Search Challenge. Bohužel k správnému nastavení, které by optimalizovalo efektivitu systému, je potřeba prohledání velkého prostoru parametrů a strategií. To právě může být problémové s reálnými uživateli. Proto jsem se pokusil odhadnout chování reálného uživatele a jeho model výběru z displeje, který bude vybírat podobně jako skutečný uživatel. K vytvoření tohoto modelu byla potřeba data od reálných uživatelů. Vytvořený Simulovaný reálný uživatel byl nastaven tak, aby vybíral s náhodným šumem, ale zároveň jeho výběr korespondoval s výběrem reálného uživatele.

Po vytvoření Simulovaného reálného uživatele jsem se snažil optimalizovat efektivitu hledání pro různé konfigurace. Více nezávislých komponent a jejich parametrů dávalo velký prostor, který musel být prohledán za účelem nalezení optima. K tomu byl využit software, který spouštěl hledání se simulovanými uživateli: Ideálním a Simulovaným reálným. Optimální hodnoty některých parametrů byly nalezeny snadno a jiné bylo třeba přeměřit, již se zmenšeným prostorem parametrů. Ideální uživatel pro optimální strategie z testovaných konfigurací dokázal vyřešit všechny úlohy do třetího displeje. Oproti tomu se ukázalo, že náhodný šum Simulovaného reálného uživatele výrazně snížil úspěšnost nalezení snímku. Při optimálním nastavení se jeho úspěšnost na desátém displeji pohybovala kolem 90%.

Po odhadu vhodných parametrů a strategií byly některé algoritmy implementovány do nástroje SOMHunter, který se ukázal jako efektivní v mezinárodní soutěži Video Browser Showdown v korejském Daejonu v lednu 2020. Z logů tohoto nástroje byl kromě efektivity textového hledání identifikován i pozitivní vliv zpětnovazebního učení na několik řešených úloh.

Proto bychom se rádi do budoucna zaměřili na další zkoumání v této oblasti.

Náš tým se plánuje i v budoucnu věnovat rozvoji a dalšímu zkoumání v této oblasti. Jsme přesvědčeni, že je stále možnost posouvat dále hranice interaktivního hledání na základě obsahu pomocí zpětnovazebního učení. Například se chceme věnovat možnosti zpětné vazby ve formě negativního označení snímku z displeje či adaptivnímu přizpůsobení parametru σ pro lepší zachycení relevancí a dosažení nezávislosti na charakteristických rysech.

Literatura

Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V. a Yianilos, P. N. (2000). The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, **9**(1), 20–37. ISSN 1941-0042. doi: 10.1109/83.817596.

Gurrin, C., Schoeffmann, K., Joho, H., Leibetseder, A., Zhou, L., Duane, A., Dang-Nguyen, D.-T., Riegler, M., Piras, L., Tran, M.-T., Lokoč, J. a Hürst, W. (2019). [invited papers] comparing approaches to interactive lifelog search at the lifelog search challenge (lsc2018). *ITE Transactions on Media Technology and Applications*, **7**(2), 46–59. doi: 10.3169/mta.7.46.

Hare, J. S., Lewis, P. H., Enser, P. G. B. a Sandom, C. J. (2006). Mind the gap: another look at the problem of the semantic gap in image retrieval. In Chang, E. Y., Hanjalic, A. a Sebe, N., editors, *Multimedia Content Analysis, Management, and Retrieval 2006*, volume 6073, pages 75 – 86. International Society for Optics and Photonics, SPIE. doi: 10.1117/12.647755. URL `https://doi.org/10.1117/12.647755`.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, **78**(9), 1464–1480. ISSN 1558-2256. doi: 10.1109/5.58325.

Kratochvíl, M., Veselý, P., Mejzlík, F. a Lokoč, J. (2020). Somhunter: Video browsing with relevance-to-som feedback loop. In Ro, Y. M., Cheng, W.-H., Kim, J., Chu, W.-T., Cui, P., Choi, J.-W., Hu, M.-C. a De Neve, W., editors, *MultiMedia Modeling*, pages 790–795, Cham, 2020. Springer International Publishing. ISBN 978-3-030-37734-2.

Li, X., Xu, C., Yang, G., Chen, Z. a Dong, J. (2019). W2VV++: fully deep learning for ad-hoc video search. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 1786–1794. doi: 10.1145/3343031.3350906. URL `https://doi.org/10.1145/3343031.3350906`.

Lokoč, J., Souček, T., Veselý, P., Mejzlík, F., Ji, J., Xu, C. a Li, X. (2020a). A w2vv++ case study with automated and interactivetext-to-video retrieval. In *Accepted to: MM '20: Proceedings of the 28th ACM International Conference on Multimedia*, pages 1–9, New York, NY, USA, 2020a. Association for Computing Machinery.

Lokoč, J., Veselý, P., Mejzlík, F., Kovalčík, G., Souček, T., Rossetto, L., Schoeffmann, K., Bailer, W., Gurrin, C., Sauter, L., Song, J., Vrochidis, S., Wu, J. a Þór Jónsson, B. (2020b). Is the reign of interactive search eternal? findings from the video browser showdown 2020. In *Submitted to: ACM Transactions on Multimedia Computing, Communications, and Applications*, pages 1–9, New York, NY, USA, 2020b. Association for Computing Machinery.

Lokoč, J., Kovalčík, G., Münzer, B., Schöffmann, K., Bailer, W., Gasser, R., Vrochidis, S., Nguyen, P. A., Rujikietgumjorn, S. a Barthel, K. U. (2019). Interactive search or sequential browsing? a detailed analysis of the video browser showdown 2018. *ACM Trans. Multimedia Comput. Commun. Appl.*, **15**(1). ISSN 1551-6857. doi: 10.1145/3295663. URL `https://doi.org/10.1145/3295663`.

Lokoč, J., Bailer, W., Schoeffmann, K., Muenzer, B. a Awad, G. (2018). On influential trends in interactive video retrieval: Video browser showdown 2015–2017. *IEEE Transactions on Multimedia*, **20**(12), 3361–3376.

Mejzlík, F., Veselý, P., Kratochvíl, M., Souček, T. a Lokoč, J. (2020). Somhunter for lifelog search. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*, LSC '20, page 73–75, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371360. doi: 10.1145/3379172.3391727. URL `https://doi.org/10.1145/3379172.3391727`.

Rossetto, L., Gasser, R., Lokoc, J., Bailer, W., Schoeffmann, K., Muenzer, B., Soucek, T., Nguyen, P. A., Bolettieri, P., Leibetseder, A. a Vrochidis, S. (2020). Interactive video retrieval in the age of deep learning - detailed evaluation of vbs 2019. *IEEE Transactions on Multimedia*, pages 1–1.

Zvára, Karel. Štěpán, J. (2019). *Pravděpodobnost a matematická statistika*. Matfyzpress. ISBN 9788073782184.

Seznam obrázků

Seznam tabulek