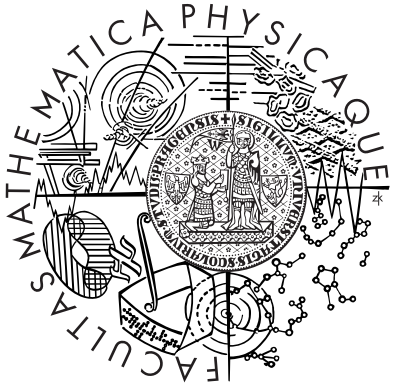
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

Diplomová práce



Bc. Jana Kravalová

Využití syntaxe v metodách pro vyhledávání informací

Ústav formální a aplikované lingvistiky
Vedoucí diplomové práce: Mgr. Pavel Pecina, Ph.D.

Studijní program: Matematická lingvistika

Ráda bych poděkovala vedoucímu své práce Mgr. Pavlu Pecinovi, Ph.D., nejen za odborné vedení během mé práce, ale také za několikaleté odborné vedení během bakalářského a magisterského studia, kdy mi vždy ochotně a pohotově poradil ve všech oblastech mého studijního oboru. Také bych chtěla poděkovat Milanu Strakovi za rady k optimalizaci programu, především k mapování souborů do paměti. V neposlední řadě bych velmi ráda poděkovala svým rodičům, kteří mě při studiu podporovali.

Prohlašuji, že jsem svou diplomovou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 16.4.2009 Jana Kravalová

Abstrakt

Název práce: *Využití syntaxe v metodách pro vyhledávání informací*

Autor: *Bc. Jana Kravalová*

Katedra (ústav): *Ústav formální a aplikované lingvistiky*

Vedoucí diplomové práce: *Mgr. Pavel Pecina, Ph.D.*

e-mail vedoucího: *pecina@ufal.mff.cuni.cz*

Abstrakt: V posledních letech výzkumu v oblasti vyhledávání informací je věnována značná pozornost metodám založeným na jazykovém modelování. I přesto, že tento přístup dovoluje použití libovolného jazykového modelu, většina publikovaných experimentů byla prováděna s klasickým n-gramovým modelem (mnohdy pouze s unigramovým modelem). Cílem diplomové práce je navrhnout, implementovat a vyhodnotit (na českých datech) metodu, která by pravděpodobnostní model obohatila o použití syntaktické informace získané automaticky (strojově) z dokumentů i dotazů. V předkládané práci se pokusíme vhodným způsobem zavést syntaktickou informaci do jazykových modelů a experimentálně srovnáme navržený přístup s výsledky unigramového a bigramového povrchového modelu. Kromě využití syntaktické informace se zaměříme také na vliv vyhlazování, stemmingu, lemmatizace, použití stopwords a metody rozšiřování dotazů – pseudo relevance feedback. Provedeme také detailní analýzu použitých systémů vyhledávání informace a podrobně popíšeme jejich vlastnosti. Experimenty budou prováděny na české testovací kolekci z Cross Language Evaluation Forum 2007 Ad-Hoc Track ([1]) a předkládané výsledky lze tedy srovnat s výsledky publikovanými v [19] a [4].

Klíčová slova: vyhledávání informací, jazykové modelování, závislostní syntax, vyhlazování

Abstract

Title: *Information Retrieval Using Syntax Information*

Author: *Bc. Jana Kravalová*

Department: *Institute of Formal and Applied Linguistics*

Supervisor: *Mgr. Pavel Pecina, Ph.D.*

Supervisor's e-mail address: *pecina@ufal.mff.cuni.cz*

Abstract: In the last years, application of language modeling in information retrieval has been studied quite extensively. Although language models of any type can be used with this approach, only traditional n-gram models based on surface word order have been employed and described in published experiments (often only unigram language models). The goal of this thesis is to design, implement, and evaluate (on Czech data) a method which would extend a language model with syntactic information, automatically obtained from documents and queries. We attempt to incorporate syntactic information into language models and experimentally compare this approach with unigram and bigram model based on surface word order. We also empirically compare methods for smoothing, stemming and lemmatization, effectiveness of using stopwords and pseudo relevance feedback. We perform a detailed analysis of these retrieval methods and describe their performance in detail.

Keywords: information retrieval, language modelling, depenency syntax, smoothing

Obsah

1 Úvod

Pravděpodobnostním metodám založeným na jazykovém modelování je v posledních letech výzkumu v oblasti vyhledávání informací věnována značná pozornost, viz např. jednu z nejcitovanějších prací [20]. I přesto, že tento přístup dovoluje použití libovolného jazykového modelu, většina experimentů, jejichž výsledky byly doposud publikovány, byla prováděna s klasickým n-gramovým modelem, založeným na povrchovém slovosledu.

Myšlenka použití jazykového modelu využívajícího syntax není v této oblasti zcela nová, ale doposud nebyl prokázán významnější přínos. Z publikovaných prací, které se věnují využití syntaxe ve vyhledávání informací, jmenujme například [12], [18] či [7].

Jedním z důvodů nízké úspěšnosti syntaktických jazykových modelů může být fakt, že dosud publikované práce vyhodnocovaly tento přístup na kolekci dokumentů v angličtině, která má poměrně pevný slovosled a přínos využití syntaxe zde nemusí být tak významný. Dalším důvodem může být poměrně nízká úspěšnost syntaktických parserů.

Cílem diplomové práce je navrhnout, implementovat a vyhodnotit (na českých datech) metodu, která by pravděpodobnostní model pro vyhledávání informací obohatila o využití syntaktické informace získané automaticky (strojově) z dokumentů i dotazů. Zaměříme se také na porovnání unigramových a bigramových modelů, porovnání metod pro stemming a lemmatizace, použití zpětné vazby (pseudo relevance feedback) oproti systému bez zpětné vazby a porovnání jednoduchých a kombinovaných modelů. Pojmem vyhledávání informací rozumíme úlohu uspořádání dokumentů podle klesající relevance pro zadaný dotaz.

Předkládaná práce je rozdělena do šesti kapitol. V kapitole 2 definujeme zadání úlohy a teoretické základy řešení úlohy. V kapitole 3 popíšeme data použitá pro vývoj a testování, v kapitole 4 pak ukážeme konkrétní řešení dané úlohy a hlavní přínos práce. Výsledky a diskuze nad nimi jsou prezentovány v kapitole 5. Práci uzavřeme v kapitole 6.

2 Teorie

2.1 Zadání úlohy

V této práci se zabýváme úlohou *vyhledávání informací (information retrieval)*. Zadání úlohy je pro zadanou kolekci dokumentů a zadaný dotaz setřídit dokumenty v kolekci podle klesající relevance k dotazu.

V textu budeme používat následující pojmy:

- *dokument* – prostý textový dokument bez další strukturace, tj. bez klíčových slov či zařazení do kategorie, pouze opatřen jedinečným identifikátorem
- *kolekce* – množina všech dokumentů ve vyhledávacím systému
- *téma* – prostý text obsahující několik vět specifikujících informační potřebu uživatele v přirozeném jazyce
- *dotaz* – posloupnost slov pro vyhledávání ve vyhledávacím systému, kterou jsme vytvořili z tématu
- *slovo, token* – jedno slovo (token) v přirozeném českém jazyce, což v češtině přibližně odpovídá rozdělení na slova podle mezer
- *term* – indexační jednotka z pohledu vyhledávacího systému. Může to být jedno slovo, dvojice slov bezprostředně následujících za sebou nebo dvojice slov, která tvoří syntaktický vztah. Zřejmě tedy počet termů nemusí odpovídat počtu slov.

2.2 Jazykový model ve vyhledávání informací

Pro řešení úlohy vyhledávání informací bylo navrženo a testováno mnoho různých systémů vyhledávání. Z nejvýznamnějších směrů můžeme jmenovat teoretické modely, jako například logické a pravděpodobnostní modely ([22], [6], [23]) a na druhé straně různé varianty vektorových modelů ([24], [25]).

V posledních letech je věnována značná pozornost pravděpodobnostním metodám založeným na jazykovém modelování ([20], [3], [17]). Základní myšlenka tohoto přístupu je vytvoření (odhadnutí) jazykového modelu pro každý

dokument v kolekci. Míra relevance dokumentu pak odpovídá pravděpodobnosti dotazu podle tohoto jazykového modelu. Výhoda tohoto přístupu spočívá v tom, že statistické metody již byly široce zpracovány v oblasti rozpoznávání mluvené řeči ([9]) a ve zpracování přirozeného jazyka vůbec ([15]).

Cílem statistického jazykového modelování je vybudovat pro daný jazyk *jazykový model*, který přiřazuje sekvencím slov pravděpodobnost, s jakou byla daná sekvence vygenerována v jazyce odpovídajícímu jazykovému modelu. Ve vyhledávání informací odpovídá jeden jazykový model jednomu dokumentu. V pravděpodobnostním pojetí pak pokládáme pravděpodobnost dotazu za pravděpodobnost, že byl dotaz „vygenerován“ jazykovým modelem odpovídajícím danému dokumentu. Pro daný dotaz Q a dokument D označujeme tuto pravděpodobnost jako $P(Q|D)$.

Abychom mohli setřídit dokumenty podle relevance, zajímá nás ale pravděpodobnost $P(D|Q)$, kterou ale můžeme s použitím pravděpodobnosti $P(Q|D)$ získat pomocí Bayesova vzorce (pro daný dotaz Q):

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)},$$

kde $P(Q)$ je stejné pro všechny dokumenty a apriorní pravděpodobnost $P(D)$ bereme jako uniformní pro všechny dokumenty, v souladu s běžnou zvyklostí ([20], [3]). Máme tedy:

$$P(D|Q) \approx P(Q|D).$$

Přístup jazykového modelování dovoluje použití libovolného jazykového modelu, přičemž se nejčastěji jedná o klasické n-gramové modely založené na povrchovém slovosledu (viz např. [9]). Cílem této práce je využít v jazykovém modelování syntaktickou informaci v českém textu. Přesná definice jazykového modelu založeného na syntaxi bude vyložena v kapitole 2.4.3. Předem poznamenáváme, že syntaxí rozumíme závislostní syntax.

2.3 Značení a definice

Dále budeme v textu používat následující značení:

- $Q_{raw} = \langle q_1,q_2,\ldots,q_N\rangle$ – dotaz, query. Uspořádaná posloupnost prvků q_i jsou slova dotazu, tak, jak jdou za sebou v povrchovém slovosledu.
- $Q = (t_1,t_2,\ldots,t_T)$ – dotaz, query. Neuspořádané[1] prvky t_i jsou termy dotazu, tedy při unigramovém modelu přesně odpovídají jednotlivým slovům dotazu, v bigramovém povrchovém modelu se jedná o povrchové dvojice slov dotazu, při syntaktickém modelu jsou to syntaktické relace.
- N – skutečná délka neboli počet slov dotazu
- T – počet termů v dotazu, tedy pro unigramový model $T = N$, pro bigramový model $T = N - 1$, pro syntaktické stromy záleží na počtu stromů v dotazu[2]
- D – dokument
- C – kolekce
- $C_D(t)$ – počet výskytů (raw frequency) termu t v dokumentu D (dokument D je bez indexu, pokud nehrozí záměna)
- $C_C(t)$ – počet výskytů (raw frequency) termu t v celé kolekci
- $P_D(t) = P(t|D)$ – pravděpodobnost termu t v dokumentu D
- $P_C(t) = P(t|C)$ – pravděpodobnost termu t v kolekci C
- $|D|$ – počet termů v daném dokumentu
- $|C|$ – počet termů v celé kolekci, čili součet přes všechny dokumenty
- $ndocs$ – počet dokumentů v kolekci
- $r(q_i)$ – syntaktický rodič slova q_i

[1] Jakmile vytvoříme z tématu dotaz pro vyhledávací systém (použitím syntaktických dvojic), chápeme dotaz jako skupinu termů, na jejichž pořadí při vyhodnocování vyhledávacím systémem nezáleží.

[2] Syntaktický strom pro jednu větu je souvislý a má jeden pomocný kořen, k němuž se připojují slova, která nemají ve větě rodiče (v českém stromě je to většinou sloveso). Takový vztah slova k pomocnému kořenu neindexujeme, takže počet termů ve větě je přesně N bez počtu slov, která se připojují k pomocnému kořenu.

- $\langle q_i,q_{i+1}\rangle$ – povrchový bigram q_i, q_{i+1} (v tomto pořadí). Ve vzorcích píšeme místo $P_D(\langle q_i,q_{i+1}\rangle)$ zkráceně $P_D(q_i,q_{i+1})$.
- $\langle q_i,\star\rangle$ – povrchový bigram, kde první slovo je q_i a druhé slovo libovolné následující slovo
- $(r(q_i),q_i)$ – syntaktický bigram, $r(q_i)$ je syntaktickým rodičem q_i. Podobně jako u povrchových bigramů zkracujeme $P_D((r(q_i),q_i))$ na $P_D(r(q_i),q_i)$.
- $(r(q_i),\star)$ – syntaktický bigram, ve kterém se vyskytuje syntaktický rodič slova q_i jako rodič a slovo na něm závislé je libovolné

Při popisu pravděpodobnosti používáme následující značení:

- P – skutečná pravděpodobnost
- $\hat{P}$ – pravděpodobnost odhadnutá pomocí odhadu maximální věrohodnosti (maximum likelihood estimation, MLE, viz kapitolu 2.4)
- $\tilde{P}$ – pravděpodobnost po vyhlazování

2.4 Popis jazykových modelů

Při experimentech budeme používat modely tří typů: unigramový model jako základ (baseline), bigramový povrchový model jako prostředek pro srovnání a především bigramový syntaktický model. V případě unigramového a bigramového povrchového modelu se jedná o všeobecně známé a používané modely. V případě bigramového syntaktického modelu jsme jednoduše rozšířili všeobecně chápaný pojem povrchového bigramu na bigram „slovo a jeho syntaktický rodič“. Jako odhad všech pravděpodobností byl použit odhad maximální věrohodnosti – maximum likelihood estimation (MLE).

2.4.1 Unigramový model

Unigramový model je nejpoužívanější z pravděpodobnostních modelů. Využívá silný předpoklad vzájemné nezávislosti slov. Je podobný vektorovému modelu s mírou tf-idf ([14]).

V tomto modelu si termy t_i a slova dotazu q_i zcela odpovídají. Platí tedy $t_i = q_i$, tj. $P_D(t_i) = P_D(q_i)$ a $T = N$.

$$P_D(Q) = P_D(t_1,t_2,\ldots,t_T) = P_D(q_1,q_2,\ldots,q_N) \approx \prod_{i=1}^{N} P_D(q_i)$$

$$\hat{P}_D(Q) = \prod_{i=1}^{N} \frac{C_D(q_i)}{|D|}$$

2.4.2 Bigramový povrchový model

V bigramovém povrchovém modelu jsou za termy pokládány vždy dvojice slov po sobě bezprostředně následujících v povrchovém slovosledu. Platí $t_i = \langle q_i,q_{i+1}\rangle$ a $P_D(t_i) = P_D(q_i,q_{i+1})$ a $T = N - 1$.

$$P_D(Q) = P_D(t_1,t_2,\ldots,t_T) = P_D(q_1,q_2,\ldots,q_N) \approx \prod_{i=1}^{N-1} P_D(q_{i+1}|q_i)$$

Podmíněná pravděpodobnost:

$$\prod_{i=1}^{N-1} \hat{P}_D(q_{i+1}|q_i) = \prod_{i=1}^{N-1} \frac{C_D(q_i,q_{i+1})}{C_D(q_i,\star)}$$

Sdružená pravděpodobnost je jiným způsobem, jak odhadnout pravděpodobnost $P_D(Q)$. V tomto přístupu chápeme dotaz jako překrývající se bigramy (dle [4]):

$$\prod_{i=1}^{N-1} \hat{P}_D(q_i,q_{i+1}) = \prod_{i=1}^{N-1} \frac{C_D(q_i,q_{i+1})}{|D|}.$$

Při použití bigramového (ať už povrchového či syntaktického) modelu samostatně nepředpokládáme zlepšení oproti výsledkům unigramového modelu. Bigramové modely chceme použít v kombinovaci s unigramovým modelem.

2.4.3 Bigramový syntaktický model

V syntaktickém modelu opouštíme povrchový slovosled a termem je zde dvojice slov v přímém syntaktickém vztahu, tedy platí $t_i = (r(q_i),q_i)$ a $P_D(t_i) =$

$P_D(r(q_i),q_i)$, T je rovno počtu syntaktických vztahů v syntaktickém stromu dotazu a $|D|$ je rovno počtu syntaktických vztahů v dokumentu D. Syntaxí rozumíme závislostní syntax.

$$P_D(Q) = P_D(t_1,t_2,\ldots,t_T) \approx \prod_{q_i:\exists r(q_i)} P_D(r(q_i),q_i)$$

Podmíněná pravděpodobnost:

$$\prod_{q_i:\exists r(q_i)} \hat{P}_D(q_i|r(q_i)) = \prod_{q_i:\exists r(q_i)} \frac{C_D(r(q_i),q_i)}{C_D(r(q_i),\star)}$$

Sdružená pravděpodobnost:

$$\prod_{q_i:\exists r(q_i)} \hat{P}_D(r(q_i),q_i) = \prod_{q_i:\exists r(q_i)} \frac{C_D(r(q_i),q_i)}{|D|}$$

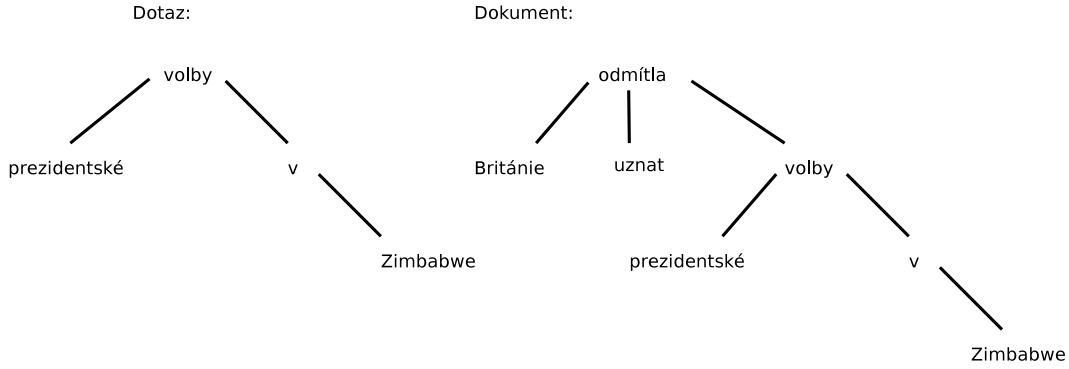
Příklad výpočtu pravděpodobnosti syntaktického stromu v dokumentu ukazuje obrázek 1.[3] Zadáním tématu je věta „prezidentské volby v Zimbabwe“. Dokument, jehož relevanci odhadujeme, obsahuje text „Británie odmítla uznat prezidentské volby v Zimbabwe“. Termy v dotazu jsou tedy syntaktické dvojice „volby“ → „prezidentské“, „volby“ → „v“ a „v“ → „Zimbabwe“. Podobně sestavíme termy ze syntaktických relací v dokumentu. Sdružená pravděpodobnost dotazu bude rovna $P = \frac{C(volby,prezidentské)}{6} \cdot \frac{C(volby,v)}{6} \cdot \frac{C(v,Zimbabwe)}{6} = \frac{1}{6} \cdot \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{216}$.

2.5 Vyhlazování

Při odhadu pravděpodobnosti pomocí MLE v jazykovém modelování je vždy klíčový způsob vyhlazování, tedy metoda upravení odhadu pravděpodobností tak, abychom získali přesnější odhady. Při odhadování pravděpodobností pomocí MLE z dat totiž dochází k „přecenění, nadhodnocení“ pravděpodobností

[3] Syntaktický (závislostní) parser vrací pro každou syntaktickou dvojici i typ vztahu, např. podmět, objekt, atd. Typ vztahu v našem systému neindexujeme, zajímá nás vždy jenom existence syntaktického vztahu.

Obr. 1: Závislostní strom pro dotaz „prezidentské volby v Zimbabwe“ a dokument „Británie odmítla uznat prezidentské volby v Zimbabwe“



jevů viděných v datech. Naopak jevy, které se v datech nevyskytly, získávají nulovou pravděpodobnost. Nulové odhady pravděpodobností způsobují jednak problémy při výpočtu celkové pravděpodobnosti pomocí násobení, jednak zcela vylučují daný jev jen proto, že se nevyskytl v trénovacích datech. Při odhadu pravděpodobností na omezeném množství textu, jako je například jeden dokument, je vyhlazování obzvlášť důležité.

Vyhlazování je metoda, která upraví odhady pravděpodobností tak, že zmenší rozdíly mezi příliš velkými a příliš malými odhady. Výsledkem je, že nulové odhady jsou mírně zvýšeny na nějakou malou nenulovou hodnotu a tyto hodnota je proporcionálně odebrána velkým odhadům. Toto se děje při zachování sumy pravděpodobnosti rovno 1.

Typický způsob evaluace kombinuje odhad pravděpodobnosti získaný na dokumentu s odhadem pravděpodobnosti získaným na velkých datech, v našem případě na celé kolekci dokumentů.

2.5.1 Jelinek-Mercer

Tato metoda spočívá v lineární interpolaci modelu ($\hat{P}_D$, MLE odhad) s modelem získaným na celé kolekci ($\hat{P}_C$, také MLE odhad), přičemž jejich poměr je určován parametrem $\lambda \in \langle 0,1 \rangle$, viz [10].

$$\tilde{P}_D(t) = \lambda \hat{P}_D(t) + (1-\lambda)\hat{P}_C(t) \quad \text{pro } \lambda \in \langle 0,1 \rangle$$

2.5.2 Dirichlet

Pro unigramový model je Dirichletovo vyhlazování dáno jako (viz [13])

$$\tilde{P}_D(t) = \frac{C_D(t) + \mu \hat{P}_C(t)}{|D| + \mu} \quad \text{pro } \mu \in \mathbb{R}^+.$$

Pro bigramové modely je tento model třeba rozšířit. Pro bigramový model s podmíněnou pravděpodobností to již provedli autoři v [11] jako

$$\tilde{P}_D(t) = \tilde{P}_D(q_{i-1}|q_i) = \frac{C_D(q_{i-1},q_i) + \mu \hat{P}_C(q_i|q_{i-1})}{C_D(q_{i-1},\star) + \mu}.$$

Pro bigramové modely se sdruženou pravděpodobností jednoduše chápeme bigram jako term, takže výsledný vzorec je

$$\tilde{P}_D(t) = \tilde{P}_D(q_{i-1},q_i) = \frac{C_D(q_{i-1},q_i) + \mu \hat{P}_C(q_{i-1},q_i)}{|D| + \mu}.$$

2.6 Evaluace ve vyhledávání informací

Pro vyhodnocování úspěšnosti výsledku při vyhledávání informací se standardně používají míry *recall*, *precision*, *average precision* a případně pro více dotazů *mean average precision*. Míry recall a precision jsou definovány pro klasifikační úlohy a jestliže je používáme pro určení úspěšnosti vyhledávacího systému, počítáme je vždy pro t prvních dokumentů, kde t je stanovený práh (treshold).

2.6.1 Recall

Recall (R) – poměr počtu relevantních nalezených dokumentů ku všem relevantním dokumentům. Recall tedy udává pravděpodobnost, že bude relevantní

dokument nalezen. Tato míra samotná ale nestačí k vyhodnocení systému, protože pokud označíme všechny dokumenty jako relevantní (nastavíme práh t na počet všech dokumentů v systému), dosáhneme triviálně recall 1.

2.6.2 Precision

Precision (P) – poměr počtu relevantních nalezených dokumentů ku počtu nalezených dokumentů. Precision tedy udává pravděpodobnost, že dokument nalezený systémem je skutečně relevantní. Precision lze také vyhodnotit nad zadanými t prvními dokumenty, pak se jedná o *precision at t docs*, např. precision at 5, precision at 10, ... Taková míra má význam například pro internetové vyhledávače, které se snaží dosáhnout vysokého precision na první stránce s výsledky vyhledávání.

2.6.3 Average precision (AP)

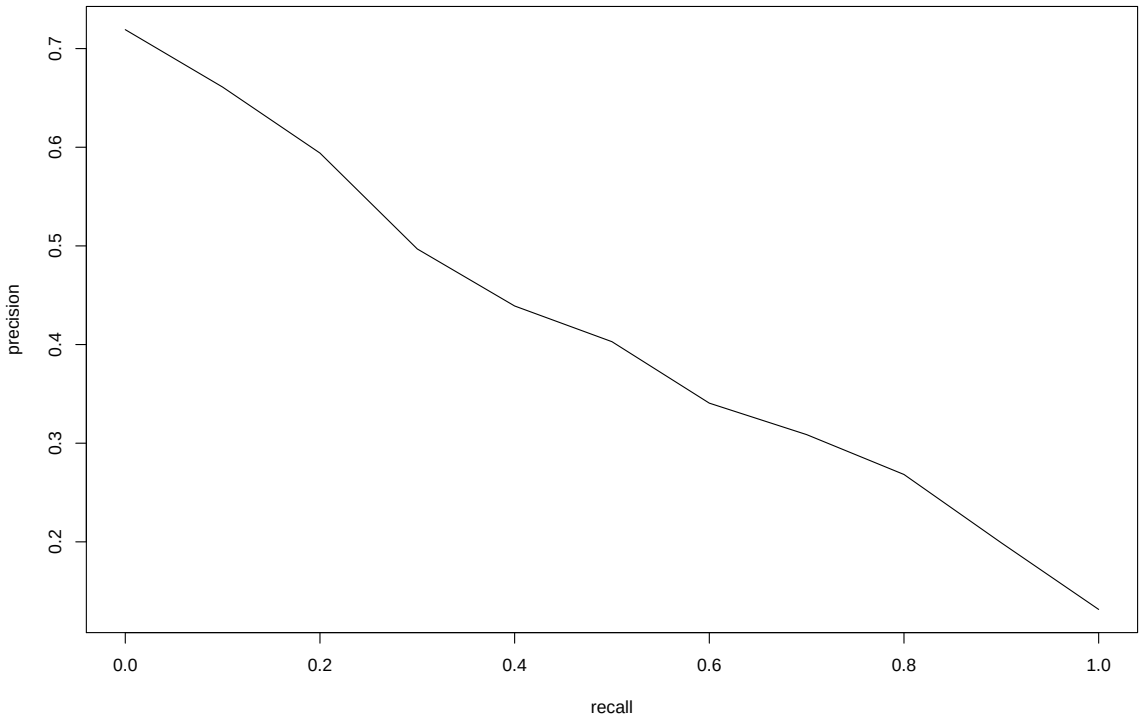
Ani recall, ani precision nelze použít samotně pro určování úspěšnosti systému, protože každá z nich vyhodnocuje pouze jednu stránku chování systému. Jejich vzájemný vztah je takový, že s klesajícím recall stoupá precision a naopak. To názorně zobrazuje precision-recall křivka na obrázku 2. Tak může jeden systém s výborným recall mít nízké precision a naopak a takové dva systémy je obtížné porovnat. Proto se zavádí jediná míra, která kombinuje oba tyto rozměry do jednoho – average precision, která navíc nevyžaduje použití parametru t.

Average precision (AP) – average precision je střední hodnotou precision jako funkce recall pro recall s uniformním rozdělením. Odhad average precision počítáme jako průměr z precision spočítaných po zkrácení setříděného seznamu vrácených dokumentů po každém relevantním dokumentu, tedy pro všechny hodnoty recall.

$$AP = E[P(R)] \quad \text{pro } R \sim U(0,1)$$

$$\widehat{AP} = \frac{\sum_{r=1}^{\#retrieved}(P(r) \cdot rel(r))}{\#relevant}$$

Obr. 2: Precision-recall křivka



kde r je pořadí dokumentu, *rel(r)* je binární funkce, která nabývá 1, pokud je dokument na r-tém místě relevantní a 0 jinak, $\#retrieved$ je počet vrácených dokumentů, $\#relevant$ je počet relevantních dokumentů a $P(r)$ je precision pro prvních r dokumentů.

Tento přístup navíc řeší problém odhadování správného tresholdu t, který jsme potřebovali při určování recall a precision.

2.6.4 Mean Average Precision (MAP)

Jestliže je naším cílem vyhodnotit systém pro více dotazů, aby naše hodnocení bylo spolehlivější, používáme míru *mean average precision (MAP)*, která je definována jako střední hodnota z AP jednotlivých dotazů.

$$MAP = E[AP]$$

$$\widehat{MAP} = \frac{\sum_{i=1}^{\#queries} AP_i}{\#queries}$$

Funkce MAP (resp. AP) má kromě svých výhod – zkombinování recall a precision do jednoho čísla a hodnocení pořadí vrácených dokumentů – i nevýhody. Její interpretace může být obtížná. Kromě toho je funkce MAP velmi závislá na datech, na kterých se měří. Stejný systém může na různých kolekcích a různých množinách dotazů vykazovat naprosto rozdílné hodnoty MAP. Proto je vždy nezbytně nutné, aby se dva systémy porovnávaly vždy na naprosto stejných kolekcích dokumentů a stejných množinách dotazů.

Nejlepší výsledky na kolekci použité pro naše experimenty (viz kapitola 3 pojednávající o datech) byly publikovány v práci [4].

2.6.5 Testy signifikance

Výsledky jednotlivých modelů je vhodné statisticky srovnat pomocí vhodného testu signifikance, například pomocí Wilcoxonova testu. Testy signifikance jsme provedli, ale kvůli velkému rozptylu AP na jednotlivých dotazech (viz výsledky v kapitole 5.3) je obtížné najít signifikantní rozdíl.

3 Data

Experimenty jsme prováděli na kolekci českých dokumentů vytvořené pro Cross Language Evaluation Forum 2007 Ad-Hoc track ([1], [19]). Jedná se o 81735 novinových článků Mladé fronty Dnes (2002) a Lidových novin (2002). Jako dotazy jsme použili 50 českých dotazů rovněž vytvořených pro účely CLEF 2007.

3.1 Vlastnosti kolekce dokumentů

Vlastnosti kolekce dokumentů shrnuje tabulka 1.

Tab. 1: Vlastnosti kolekce dokumentů

počet dokumentů	81735
velikost před syntaktickou analýzou	178 MB
velikost po syntaktické analýze	1,3 G
počet slov v celé kolekci	28587766
průměrný počet slov v dokumentu	349,76
počet unikátních slov v kolekci	556701
průměrný počet unikátních slov v dokumentu	179,38

3.2 Témata/dotazy

Kolekce obsahuje 50 témat zadaných v přirozeném jazyce a součástí úlohy je tedy vytvořit ze zadaných témat dotaz pro vyhledávací systém. Jak ukazuje příklad typického tématu na obrázku 3, zadaná témata se skládají ze tří částí: `<title>`, `<desc>` a `<narr>`, přičemž `<title>` zadává několikaslovné heslo tématu (klíčová slova či krátkou frázi), `<desc>` podrobněji popisuje téma a `<narr>` velmi podrobně popisuje vlastnosti relevantních dokumentů. Pokud neuvedeme jinak, pracujeme vždy s celým textem tématu, tedy se všemi třemi částmi[4]. Vlastnosti témat shrnuje tabulka 2.

[4]Dotaz tedy vzniká jako konkatenace částí `<title>`, `<desc>` a `<narr>`.

Obr. 3: Ukázka tématu

```
<top lang="cs">
  <num>10.2452/432-AH</num>
  <title>Prezidentské volby v Zimbabwe</title>
  <desc>
     Kdo vyhrál prezidentské volby v Zimbabwe v březnu 2002?
  </desc>
  <narr>
    Relevantní dokumenty uvádějí jméno vítěze prezidentských voleb
    v Zimbabwe.
  </narr>
</top>
```

Tab. 2: Vlastnosti témat (počítány pro všechny tři části `<title>`, `<desc>` a `<narr>`)

počet dotazů	50
počet relevantních dokumentů ke všem dotazům	762
průměrný počet relevantních dokumentů pro dotaz	15,24
medián počtu relevantních dokumentů pro dotaz	10,5
maximum relevantních dokumentů pro dotaz	47
minimum relevantních dokumentů pro dotaz	2

3.3 Rozdělení dat na trénovací a testovací

Pro účely experimentů bylo třeba rozdělit data na část trénovací a testovací. Celkových 50 dotazů tedy bylo rozděleno na 10 trénovacích dotazů a 40 testovacích dotazů.

Menší podmnožinu 10 trénovacích dotazů jsme použili k počátečnímu nastavení systému, odhadnutí parametrů λ a μ pro vyhlazování (viz kapitola 2.5), odhadnutí koeficientů pro lineární kombinace modelů (viz kapitola 4.4.1. Pro odhadnutí parametrů metodou expectation-maximization (EM, viz kapitola

4.4.2) jsme potřebovali větší množství dat v přirozeném jazyce jako tzv. heldout-data. Použili jsme 600 novinových článků z Lidových novin z webové kolekce z roku 2003. Podrobněji o odhadování parametrů viz kapitolu 4.

Výsledky uvedené v kapitole 5 jsou výsledky experimentů provedených na druhé, nezávislé množině 40 dotazů. Při porovnávání výsledků například s výsledky publikovanými v [4] je tedy nutné vzít v úvahu, že naše experimenty probíhaly pouze na podmnožině 40 dotazů oproti celkovému množství 50 dotazů. U nejlepšího dosaženého výsledku uvádíme pro porovnání tedy i výsledek dosažený na celé množině všech 50 dotazů, což není zcela korektní a číslo uvádíme pouze pro srovnání.

Rozdělení dotazů na trénovací a testovací podmnožinu jsme provedli náhodně. Vlastnosti obou množin jsou uvedeny v tabulce 3.

Tab. 3: Vlastnosti trénovací a testovací sady dotazů (počítány pro všechny tři části `<title>`, `<desc>` a `<narr>`)

	trénovací	testovací	všechny
průměrný počet slov v dotazu	48,35	44,67	46,14
maximální počet slov v dotazu	98	87	98
minimální počet slov v dotazu	24	26	24
průměrný počet relevantních dokumentů	16,3	14,5	15,24
maximální počet relevantních dokumentů	47	44	47
minimální počet relevantních dokumentů	4	2	2

4 Metodologie

Tato kapitola podrobně popisuje indexaci dokumentů, vytvoření dotazů pro vyhledávací systém ze zadaných témat, vyhodnocení dotazů a technické detaily implementace.

4.1 Lemmatizace a stemming

Určitá forma stemmingu je zahrnuta ve většině současných vyhledávacích systémů. *Stemming* je zobrazení ze slova (slovní formy) na slovní základ, bázi či kořen. Stemming může mít lingvistickou motivaci, kdy se snažíme například převést slova na jejich kořeny či základní formy (např. první pád jednotného čísla pro podstatné jméno) nebo se může jednat o tzv. *Porter-style* stemming, viz [21], kdy odtrháváme několik posledních znaků slova.

Lemmatizace v lingvistice je převedení slova (slovní formy) na její základní tvar, tzv. *lemma*. Z pohledu systému vyhledávání informací můžeme lemmatizaci chápat jako prvně jmenovaný, lingvisticky zdůvodněný způsob stemmingu.

Základní rozdíl mezi stemmingem a lemmatizací spočítá v tom, že lemmatizace převádí slova na lingvisticky oprávněné, korektní tvary, kdežto stemming pouze odtrhává koncovky, a tak může dospět k nesmyslným tvarům.

V této práci budeme používat pojem *stemming* v jeho konkrétnější podobě, čili stemming jako Porter-style stemming neboli odtrhávání koncovek, a pojem *lemmatizace* jako lingvistické převedení slova na jeho základní podobu. V kapitole 5 uvádíme porovnání výsledků při použití Porter-style stemmeru převzatého z článku [4], který budeme jako autoři dále nazývat light+, a lemmatizátoru [8] pro češtinu. Protože použití stemmingu je již dobře prozkoumáno a popsáno např. v [4], zaměříme se v této práci spíše na použití lemmatizace.

Výhoda Porter-style stemmeru spočívá v tom, že jej lze vytvořit bez znalosti daného jazyka. Například stemmer light+ odtrhává nejčastější koncovky v češtině, takže ze slova `pěknému` vznikne `pěkn` a ze slova `kočkách` vznikne

kočk. Naopak lemmatizátor vytvořený se znalostí daného jazyka může dosáhnout vyšší úspěšnosti. Příkladem lemmatizace slova pěknému je pěkný a lemmatizace slova kočkách je kočka.

4.2 Použité části tématu

Jak je vidět na obrázku 3, témata se skládají ze tří částí: <title>, <desc> a <narr>. Čím více částí použijeme, tím delší dotaz získáme. Použití delšího dotazu může přidat více informací pro vyhledávání, ale může také přidat více zbytečných zavádějících slov. Například část <narr> často obsahuje věty typu „Najděte jen takové dokumenty, které pojednávají…“

4.3 Důležitost termů v dotazu

Jak jsme naznačili v předchozí kapitole, při vyhodnocování dotazu je třeba odhadnout důležitost jednotlivých částí (termů) dotazu. V této práci jsme použili několik přístupů k ohodnocení důležitosti termů, které můžeme rozdělit do tří skupin:

1. žádné odhadování důležitosti termů v dotazu, čili všechny termy mají stejnou váhu při vyhodnocování (všechny mají stejnou důležitost)
2. systém *stopwords*
3. snaha vypočítat důležitost slova například z jeho frekvence výskytu v celém korpusu nebo z počtu dokumentů, ve kterých se slovo objevilo

Důležitost termu jsme při výpočtu pravděpodobnosti zapojili tak, že se pravděpodobnost daného termu vynásobila jeho důležitostí. Přesně vzato už při výpočtu nepracujeme s pravděpodobnostním rozložením.

4.3.1 Stopwords

Skoro všechny systémy vyhledávání informací odstraňují ještě před samotným vyhledáváním tzv. *stopwords*, což jsou většinou funkční slova (např. spojky) či slova s velmi vysokou frekvencí v korpusu jako např. budeme, on a podobně. To

většinou vede k zlepšení výkonu systému, i když samozřejmě existují protipříklady, kdy odstranění stopwords vede ke ztrátě smyslu dotazu, jako například když je dotaz kolokace nebo ustálené slovní spojení. My jsme v této práci použili seznam 256 slovních forem volně přístupných z CLEF, [1]. Navíc v dotazu ignorujeme slova jako „relevantní“, „dokumenty“, atd. a používáme pouze slova slovních druhů přídavná jména, číslovky, příslovce, podstatná jména, zájmena, slovesa a předložky. Odstraněny jsou tedy zejména interpunkce a spojky.

Co se týče bigramů, je třeba rozhodnout, kdy prohlásíme bigram za stopword. Můžeme to udělat tehdy, když je alespoň jedna z jeho částí stopwords, nebo tehdy, když jsou obě jeho části stopwords. My jsme se rozhodli pro druhý přístup, protože první přístup příliš striktně zahazoval většinu bigramů a především zahazoval bigramy obsahující důležité informace, jako například v příkladu „volby v Zimbabwe“. V tomto případě by z dotazu nezbylo vůbec nic, protože jak bigram „volby v“, tak bigram „v Zimbabwe“ obsahuje stopword „v“.

4.3.2 Výpočet důležitosti termu z informací v kolekci

Při tomto přístupu se snažíme důležitost termu odstupňovat podle charakteristik termu, které získáme na celé kolekci. V této práci jsme jako důležitost termu vyzkoušeli dva výpočty:

- Převrácená frekvence termu v kolekci, tj. $\frac{1}{C_C(t)}$. Čím je term v kolekci vzácnější, tím je důležitější a naopak.
- Známá míra idf: $\log \frac{ndocs}{|\{D:t \in D\}|}$. Důležitost termu je dána jako logaritmus převrácené hodnoty počtu dokumentů, ve kterých se term vyskytuje. V čím více dokumentech se term vyskytuje, tím má menší schopnost rozlišit dva dokumenty a tím méně je tedy důležitý.

4.4 Kombinace modelů

V jazykovém modelování se kromě jednotlivých modelů s výhodou používají i jejich kombinace. V našem systému vyhledávání informací se také pokusíme jednotlivé jazykové modely – unigramový, bigramový povrchový a bigramový syntaktický – zkombinovat. Doufáme, že rozšířením samotného unigramového modelu získáme systém s vyšší úspěšností, protože i když jsou bigramové modely všeobecně méně úspěšné, mohou najít jiné relevantní dokumenty, které se pomocí pouze unigramového modelu najít nepodařilo.

Pro jednoduchost budeme modely skládat lineární kombinací.

V průběhu práce se ukázalo, že najít lineární koeficienty pro kombinaci jednotlivých modelů je náročné. Naším cílem je maximalizovat funkci MAP (viz kapitola 2.6 o evaluaci). Funkce MAP je tedy naší cílovou funkcí, ale nezaručuje takové vlastnosti, abychom pro její maximalizaci mohli uplatnit nějaký známý maximalizační algoritmus, kromě triviálního *grid search*, který popíšeme v následujícím odstavci.

4.4.1 Grid search

Grid search je triviální metoda vyzkoušení všech kombinací hodnot v zadaném prostoru. Pro každou zkoušenou kombinaci hodnot spočteme cílovou funkci (v našem případě MAP) a vybereme optimální kombinaci. Jedná se pouze o heuristiku, která nezaručuje dosažení optimálního výsledku. Časová náročnost této metody stoupá s počtem proměnných, pro které hledáme optimální hodnotu, s rozsahem jejich intervalů a s podrobností prohledávání. Přesto, jak ukážeme v kapitole 5, jsme pomocí této jednoduché metody získali dobré výsledky. Zároveň s výsledky uvádíme i lineární koeficienty jednotlivých modelů.

4.4.2 EM algoritmus

Vycházíme-li z předpokladu, že kvalitu jazykového modelu dobře popisuje entropie, můžeme si za cílovou funkci zvolit entropii a pokusit se ji minimalizovat

pomocí *expectation-maximization (EM) algoritmu*. EM je algoritmus využívaný ve statistickém zpracování přirozeného jazyka pro odhadnutí parametrů jazykových modelů, viz např. [9]. My jsme pomocí tohoto algoritmu odhadli lineární koeficienty pro kombinaci jednotlivých modelů tak, aby entropie výsledného jazykového modelu byla co nejnižší. Jako held-out data jsme použili 600 novinových článků z webové kolekce Lidových novin, viz kapitola 3. Výsledné váhy jednotlivých modelů a výsledky kombinovaného modelu uvádíme v kapitole 5.

4.4.3 Triviální kombinace modelů

Pro úplnost jsme se pokusili modely zkombinovat ještě jednoduchým způsobem:

- kombinované ohodnocení relevance dokumentu vzniká prostým sečtením ohodnocení daných jednotlivými modely
- kombinované ohodnocení relevance dokumentu vzniká sečtením ohodnocení daných jednotlivými modely, kde modely byly sčítány s vahami danými jejich spolehlivostí, tj. model, který v dotazu rozpoznal větší množství termů, byl započítán s větší vahou než méně spolehlivý model

4.4.4 Normalizace (škálování) výsledků

Před kombinováním pravděpodobností $P_D(Q)$ jednotlivých modelů je vhodné tyto pravděpodobnosti přeškálovat do stejných velikostí:

- lineární normalizace: $\frac{P_D(Q)-min}{max-min}$, čili $P_D(Q) \mapsto \langle 0,1 \rangle$. Tento způsob normalizace není vhodný kvůli exponenciálnímu rozložení pravděpodobnosti (při počítání s pravděpodobnostmi sčítáme jejich logaritmy).
- normalizace průměrem: $\frac{P_D(Q)-mean}{stdev}$, čili všechna $P_D(Q)$ jsou převedena do rozdělení s $\mu = 0$ a $stdev = 1$.

4.5 Pseudo relevance feedback

Pro nalezení více relevantních dokumentů může být vhodné rozšířit dotaz o sémanticky příbuzná slova. Technika rozšiřování dotazu o nová slova se nazývá *Query expansion*. Příkladem této techniky je použití tezauru, např. WordNet ([5]), či relevance feedback.

Relevance feedback je metoda používaná v některých systémech vyhledávání informací. Myšlenka spočívá v tom, že použijeme k prvních dokumentů vrácených systémem v prvním, inicializačním běhu a od uživatele získáme informaci o tom, které dokumenty jsou relevantní. Termy obsažené v několika prvních relevantních dokumentech vrácených v inicializačním běhu systému použijeme ke zpřesnění vyhledávání. Doufáme přitom, že první relevantní dokumenty by mohly obsahovat synonyma či rozšíření termů obsažených v původním dotazu a že tedy rozšíření dotazu o tyto nové termy pomůže v dalším běhu najít nové relevantní dokumenty.

Pseudo relevance feedback je automatická metoda pro analýzu výsledků z prvního běhu v případě, kdy nechceme vyžadovat uživatelskou interakci. Metoda spočívá v tom, že po prvním inicializačním běhu prohlásíme prvních k dokumentů za relevantní a stejně jako při uživatelsky řízeném feedbacku použijeme jejich termy při vyhledáváním v dalším běhu.

V této práci jsme použili velmi jednoduchý automatický způsob rozšíření původního dotazu o nové termy – pseudo feedback. Termy obsažené v prvních k dokumentech vrácených při prvním běhu jsme přidali k původnímu dotazu.[5] Tím se samozřejmě délka dotazu až několikanásobně zvětšila, což jsme kompenzovali tak, že jsme pro každý přidaný term vypočítali jeho důležitost pomocí idf, a tak snížili důležitost funkčních či příliš častých termů a posílili důležitost vzácných termů. Odhad velikosti k i počtu iterací pseudo feedbacku jsme provedli na trénovacích dotazech popsaných v kapitole 3 prostým vyzkoušením několika možností. Výsledky při použití pseudo feedbacku jsou uvedeny v kapitole 5.

[5] Technicky jsme zřetězili původní dotaz a obsah prvních k dokumentů za sebe.

4.6 Implementace

Morfologická analýza byla provedena pomocí taggeru J. Hajiče ([8]). Tento tagger vrací jednak morfologickou značku daného slova, jednak jeho lemma[6]. Syntaktická analýza byla provedena McDonaldovým parserem ([16]). Morfologické i syntaktické předzpracování dat bylo provedeno v systému TectoMT ([26]).

Ukázku dotazu po provedení syntaktické analýzy lze najít na obrázku 4. Kromě `XML` značek vidíme, že původní text (2. sloupec) byl obohacen o lemma (3. sloupec), morfologickou značku (4. sloupec), číslo slova ve větě, na kterém dané slovo závisí (5. sloupec) a konečně typ syntaktického vztahu (6. sloupec).

Samotný systém pro indexování a vyhledávání relevantních dokumentů byl implementován v `C++`. Dokumenty byly indexovány v invertovaném indexu, tj. pro každý term jsme udržovali setříděný seznam dokumentů, ve kterém se term vyskytuje a počet výskytů v daném dokumentu. Obraz paměti indexu jsme uložili na disk a při vyhledávání pouze namapovali do paměti.

Pro měření výsledků této práce jsme použili standardní nástroj *trec_eval*, [2].

Obr. 4: Ukázka dotazu

```
<top lang="cs">
<num>
10.2452/432-AH
</num>
<title>
1   Prezidentské    prezidentský    AAFP1----1A----    2    Atr
2   volby           volba           NNFP1-----A----    0    ExD
3   v               v-1             RR--6----------    2    AuxP
4   Zimbabwe        Zimbabwe_;G     NNNXX-----A----    3    Atr
</title>
...
```

[6]Lemma může být obohaceno o různé technické či lingvistické informace, jako je vidět na lemmatu `Zimbabwe_;G` na ukázce 4. My jsme lemmata používali celá přesně ve tvaru vráceném taggerem.

5 Výsledky a diskuze

5.1 Zkratky modelů

Modely, jejich parametry a kombinace uvedené v tabulkách jsou značeny následujícími zkratkami.

5.1.1 Model

Pro definice použitých modelů viz 2.4. Jednotlivé modely jsou popsány kombinací značek {U,B}×{P,S}×{F,L}, které po řadě popisují zda se jedná o unigramový (U) nebo bigramový (B) model, povrchové (P) nebo syntaktické (S) n-gramy a zda se indexují slovní formy (F) nebo lemmata (L).

- **UPF** – unigramový model, kde jsou jako termy použita slova v původní podobě v textu (slovní formy)
- **UPL** – unigramový model, kde jsou jako termy použita slovní lemmata
- **BPF** – bigramový model, kde jsou jako termy použity dvojice slov v původní podobě v textu (slovní formy) bezprostředně následující za sebou v povrchovém slovosledu
- **BPL** – bigramový model, kde jsou jako termy použity dvojice slovních lemmat bezprostředně následujících za sebou v povrchovém slovosledu
- **BSF** – bigramový model, kde jsou jako termy použity dvojice slov v původní podobě v textu (slovní formy) ve vztahu slovo a jeho syntaktický rodič
- **BSL** – bigramový model, kde jsou jako termy použity dvojice slovních lemmat ve vztahu slovo a jeho syntaktický rodič

5.1.2 Ohodnocovací funkce

- **cond** – podmíněná pravděpodobnost, definice viz 2.4
- **joint** – sdružená pravděpodobnost, definice viz 2.4

5.1.3 Vyhlazování

- **Jelinek-Mercer** – Jelinek-Mercer, viz 2.5.1
- **Dirichlet** – Dirichlet, viz 2.5.2

5.1.4 Použité části dotazu

- **t** – `<title>`
- **td** – `<title>` a `<desc>`
- **tdn** – `<title>`, `<desc>` a `<narr>`

5.1.5 Důležitost termu v dotazu

- **equal** – Všechny termy jsou stejně důležité a jsou brány s váhou 1.
- **stopwords** – Pokud term je term stopword, má váhu 0, což odpovídá ignoraci termu, jinak 1. Z CLEF [1] bylo převzato 256 stopword forem. Navíc v dotazu ignorujeme slova jako "relevantní", "dokumenty", atd., a používáme pouze slova slova slovních druhů `ACDNPVR`.
- **col_freq** – Důležitost termu t je dána $\frac{1}{C_C(t)}$.
- **idf** – Důležitost termu t je dána $\log \frac{ndocs}{|\{d_j : t \in d_j\}|}$.

5.1.6 Stemming

- **nostem** – Je-li uvedeno **nostem** nebo není-li uvedeno jinak, žádný stemmer nebyl použit.
- **light+** – Stemmer light+ z článku [4].

5.1.7 Kombinace modelů

- **sum_by_rel** – Pro každý dotaz se určí pořadí dokumentů podle relevance v každém ze šesti modelů, přičemž se určí, jak spolehlivé jsou jednotlivé modely. Míra spolehlivosti je poměr neznámých termů k počtu termů v dotazu (v daném modelu). Výsledné ohodnocení dokumentu je součtem

ohodnocení dokumentu v jednotlivých modelech vynásobených spolehlivostí daného modelu.

- **norm_sum_by_rel** – Jako sum_by_rel, ale s normalizací.
- **sum_by_1** – Prosté sečtení ohodnocení dokumentů od všech modelů.
- **norm_sum_by_1** – Jako norm_sum_by_1, ale s normalizací.
- **em_surface** – Lineární kombinace povrchových modelů s vahami odhadnutými pomocí EM algoritmu.
- **em_syntax** – Lineární kombinace syntaktických modelů s vahami odhadnutými pomocí EM algoritmu.
- **grid_search** – Lineární kombinace modelů, váhy nalezeny pomocí grid search.
- **grid_search_surface** – Lineární kombinace povrchových modelů, váhy nalezeny pomocí grid search.
- **grid_search_syntax** – Lineární kombinace syntaktických modelů, váhy nalezeny pomocí grid search.
- **norm_grid_search** – Lineární kombinace modelů, váhy nalezeny pomocí grid search, normalizace.
- **norm_grid_search_surface** – Lineární kombinace povrchových modelů, váhy nalezeny pomocí grid search, normalizace.
- **norm_grid_search_syntax** – Lineární kombinace syntaktických modelů, váhy nalezeny pomocí grid search, normalizace.

5.1.8 Pseudo feedback

- **none** – Není-li uvedeno jinak, žádný pseudo relevance feedback nebyl použit.
- **pseudo_feedback** – Pseudo relevance feedback byl použit, uvádí se s počtem iterací a použitých dokumentů.

5.2 Výsledky

5.2.1 Ohodnocovací funkce a vyhlazování

V každém vyhledávacím systému je nejdůležitější volba dobré funkce, která ohodnocuje relevanci dokumentu vzhledem k danému dotazu. V tabulce 4 uvádíme výsledky při použití sdružené pravděpodobnosti a podmíněné pravděpodobnosti (definice viz kapitola 2.4). Sdružená pravděpodobnost zjevně dává lepší výsledky než podmíněná, což jsme neočekávali.

Tento výsledek se můžeme pokusit vysvětlit následující hypotézou: Při vyhledávání není tak zajímavá skutečnost, jak se k sobě mají slova navzájem ve smyslu které následuje za kterým, čili které je podmíněno kterým, jak to počítá podmíněná pravděpodobnost, ale spíš, jaký vztah má daná dvojice slov k celému dokumentu, zda se v něm vyskytuje často nebo méně často v poměru k jeho velikosti. Například informace, že $P_D(b|a) = 1$ sice značí, že se slovo b vždy vyskytuje za slovem a, ale informace $P_D(a,b) = 0{,}25$ je užitečnější, protože znamená, že dvojice (a,b) tvoří čtvrtinu dokumentu. Toto pozorování vyplývá z toho, že při vyhledávání informace pracujeme s mnohem menšími a omezenějšími jazykovými modely danými krátkými dokumenty – krátkými vzhledem k jazykovému modelu vytvořenému na základě celého jazyka. Především je to ale dáno typem úlohy: Zadání úlohy jazykového modelu při vyhledávání informace bychom mohli přeformulovat spíše jako „zjištění důležitosti daného unigramu (bigramu) v dokumentu“ než jako „vztah jednotlivých slov navzájem“.

Jak jsme očekávali, použití samostatných bigramových modelů přináší horší výsledky než použití unigramových modelů. Kombinacím modelů se budeme dále věnovat v kapitole 5.2.7.

Tabulka 4 také ukazuje rozdíl při použití vyhlazování pomocí metody Jelinek-Mercer (definice viz kapitola 2.5.1) a pomocí metody Dirichlet (viz 2.5.2). Pro sdruženou pravděpodobnost dávají obě metody srovnatelné výsledky, pro podmíněnou pravděpodobnost nepřináší použití Dirichletovy me-

Tab. 4: MAP pro kombinace sdružené a podmíněné pravděpodobnosti a různá vyhlazování, λ pomocí grid-search, části **tdn**, **stopwords**, stemmer **light+**, tučně vyznačen nejlepší výsledek pro daný model

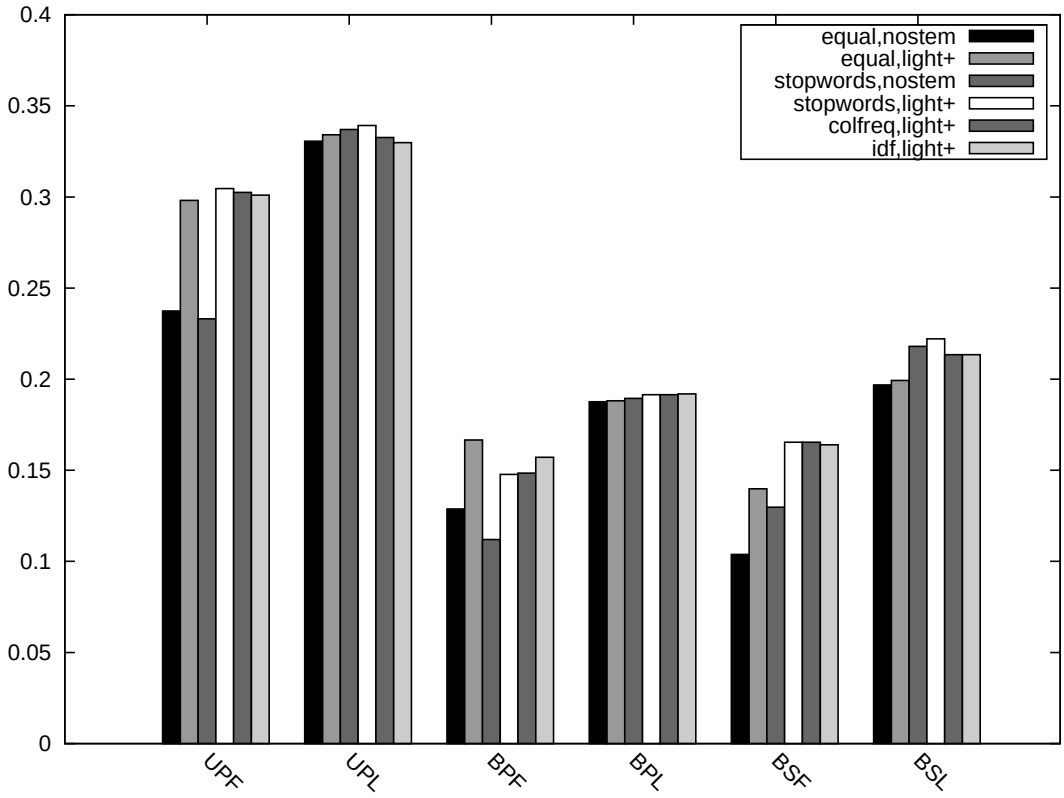
	joint		cond	
	Jelinek-Mercer	Dirichlet	Jelinek-Mercer	Dirichlet
UPF	**0,3046**	0,2941	–	–
UPL	0,3392	**0,3404**	–	–
BPF	**0,1477**	0,1456	0,1256	0,0767
BPL	0,1915	**0,2205**	0,1464	0,0903
BSF	**0,1654**	0,1588	0,1302	0,0879
BSL	0,2211	**0,2254**	0,1612	0,1248
λ	$0{,}01 - 0{,}32$	$600 - 8000$	0,99	$4 - 9$

tody dobré výsledky, pravděpodobně proto, že je obtížné toto vyhlazování matematicky rozšířit na bigramy.

5.2.2 Stemming a lemmatizace

Zajímal nás vliv stemmingu a lemmatizace na výsledné MAP. Při našich experimentech jsme potvrdili, že určitá forma lemmatizace nebo stemmingu velmi výrazně zlepšuje výsledky, jak je vidět na obrázku 5 ve velmi patrných rozdílech mezi prvním a druhým sloupcem u modelů, které používaly pouze slovní formy (UPF, BPF, BSF). První sloupec znázorňuje výsledky s použitím původních slovních forem v textu, druhý sloupec ukazuje výsledky po použití stemmingu. Stejného zlepšení jsme dosáhli při použití lemmatizace, tedy u modelů UPL, BPL, BSL. Z obrázku 5 vyplývá, že výsledky při použití stemmingu a lemmatizace jsou srovnatelné. Důležité je tedy provést nějaké mapování na základní tvary, zvlášť u češtiny jako jazyka s bohatou morfologií, aby se zamezilo přílišné řídkosti dat.

Obr. 5: Vliv stopwords, stemmingu a lemmatizace na MAP pro **joint** funkci, vyhlazování **Jelinek-Mercer**, λ pomocí grid-search, části **tdn**



důležitost termu	stem	UPF	UPL	BPF	BPL	BSF	BSL
equal	nostem	0,2374	0,3306	0,1288	0,1876	0,1038	0,1968
equal	light+	0,2981	0,3341	0,1666	0,1881	0,1398	0,1993
stopwords	nostem	0,2331	0,3370	0,1120	0,1895	0,1297	0,2180
stopwords	light+	0,3046	0,3392	0,1477	0,1915	0,1654	0,2211
col_freq	light+	0,3025	0,3326	0,1484	0,1915	0,1654	0,2134
idf	light+	0,3010	0,3298	0,1571	0,1919	0,1640	0,2134

5.2.3 Stopwords a důležitost termu v dotazu

V obrázku 5 na jednotlivých řádcích uvádíme také rozdíly při použití různých metod pro určení důležitosti termů v dotazu. Zde můžeme uvést, že použití stopwords dává přibližně stejné výsledky jako výpočet důležitosti termů pomocí frekvence v celé kolekci (col_freq) a pomocí míry idf.

5.2.4 Použití různých částí tématu při tvorbě dotazu

Při převodu tématu zadaného třemi částmi `<title>`, `<desc>` a `<narr>` na dotaz zpracovatelný vyhledávacím systémem se jednoznačně ukazuje výhodným použít všechny tři části, ne například jen titulek. I když titulek většinou obsahuje nejstručnější shrnutí daného tématu bez zbytečných pokynů a téměř bez stopwords, dokáží si metody určování důležitosti termů poradit se zbytečnými slovy v rozsáhlejších částech `<desc>` a `<narr>` a vyextrahovat z nich užitečné informace pro vyhledávání. Delší dotazy jsou také navíc i lepší pro jazykové modelování. V tabulce 5 vidíme vzestupnou tendenci MAP při použití delších dotazů.

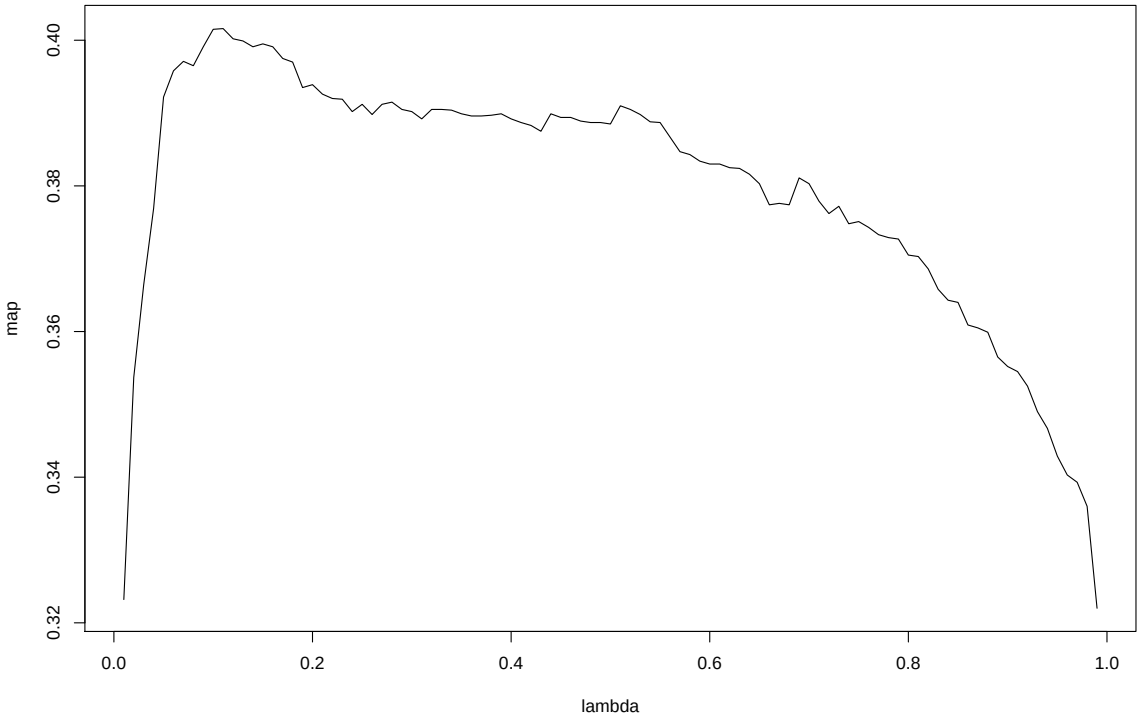
Tab. 5: MAP při použití různých částí dotazu `<title>`, `<desc>`, `<narr>`, nastavení modelů: **joint** funkce, **stopwords**, stemmer **light+**

vyhlazování	části	UPF	UPL	BPF	BPL	BSF	BSL
Jelinek-Mercer	t	0,1661	0,2198	0,1221	0,1892	0,1209	0,1922
	td	0,2627	0,3212	0,1532	0,2116	0,1495	0,2090
	tdn	0,3046	0,3392	0,1477	0,1915	0,1654	0,2211
Dirichlet	t	0,1821	0,2994	0,1257	0,1898	0,1292	0,1856
	td	0,2625	0,3292	0,1546	0,2102	0,1504	0,2311
	tdn	0,2941	0,3404	0,1456	0,2205	0,1588	0,2254

5.2.5 Odhadnutí parametrů vyhlazování

Při vyhlazování bylo potřeba také odhadnout parametry pro metodu Jelinek-Mercer (λ, viz kapitola 2.5.1) a metodu Dirichlet (μ, viz kapitola 2.5.2). To jsme provedli prostým průchodem intervalu $\langle 0{,}1 \rangle$ s krokem po 0,01. Vliv hodnoty parametru λ na MAP pro model UPL ukazuje obrázek 6. Volba správné velikosti parametru má zjevně velký vliv na výsledek systému, protože výsledky MAP se různí od hodnot 0,32 až do hodnot 0,40, a to můžeme potvrdit pro všechny modely, nejen znázorněný UPL.

Obr. 6: Závislost MAP na parametru λ pro model **UPL**, **joint** funkce, vyhlazování **Jelinek-Mercer**, části **tdn**, **stopwords**, stemmer **light+**



5.2.6 Pseudo relevance feedback

Při použití pseudo relevance feedback, zpětné vazby, je vždy třeba rozhodnout, kolik dokumentů použijeme pro rozšíření dotazu a kolikrát tento proces zopakujeme. Tabulka 6 ukazuje vliv těchto parametrů pro MAP modelu UPL. Implementace zpětné vazby, kterou jsme zavedli pro naše experimenty, upřednostňuje menší počet použitých dokumentů $(3-5)$ a menší počet iterací $(2-3)$.

Obrázek 7 a tabulka 7 ukazují výhodnost použití zpětné vazby (pseudo relevance feedback) pro všechny modely se sdruženou pravděpodobností a vyhlazováním Jelinek-Mercer. Zlepšení se různí od 0,0066 do 0,0529 a v průměru činí 0,0232.

Tabulka 8 také uvádí nejlepší výsledky bez zpětné vazby a se zpětnou vazbou dosažené pomocí sdružené pravděpodobnost a vyhlazování Dirichlet.

Tab. 6: Vliv parametrů pseudo feedbacku na MAP pro model UPL, **joint** funkce, vyhlazování **Jelinek-Mercer**, $\lambda = 0{,}11$ pomocí grid-search, části **tdn**, **stopwords**, stemmer **light+**

dokumenty ↓, běhy →	–	2	3	4
–	0,3392	–	–	–
1	–	0,3592	0,3340	0,3131
3	–	0,3731	0,3604	0,3526
5	–	0,3549	0,3761	0,3724
10	–	0,3549	0,3474	0,3445
50	–	0,3018	0,2900	0,2707

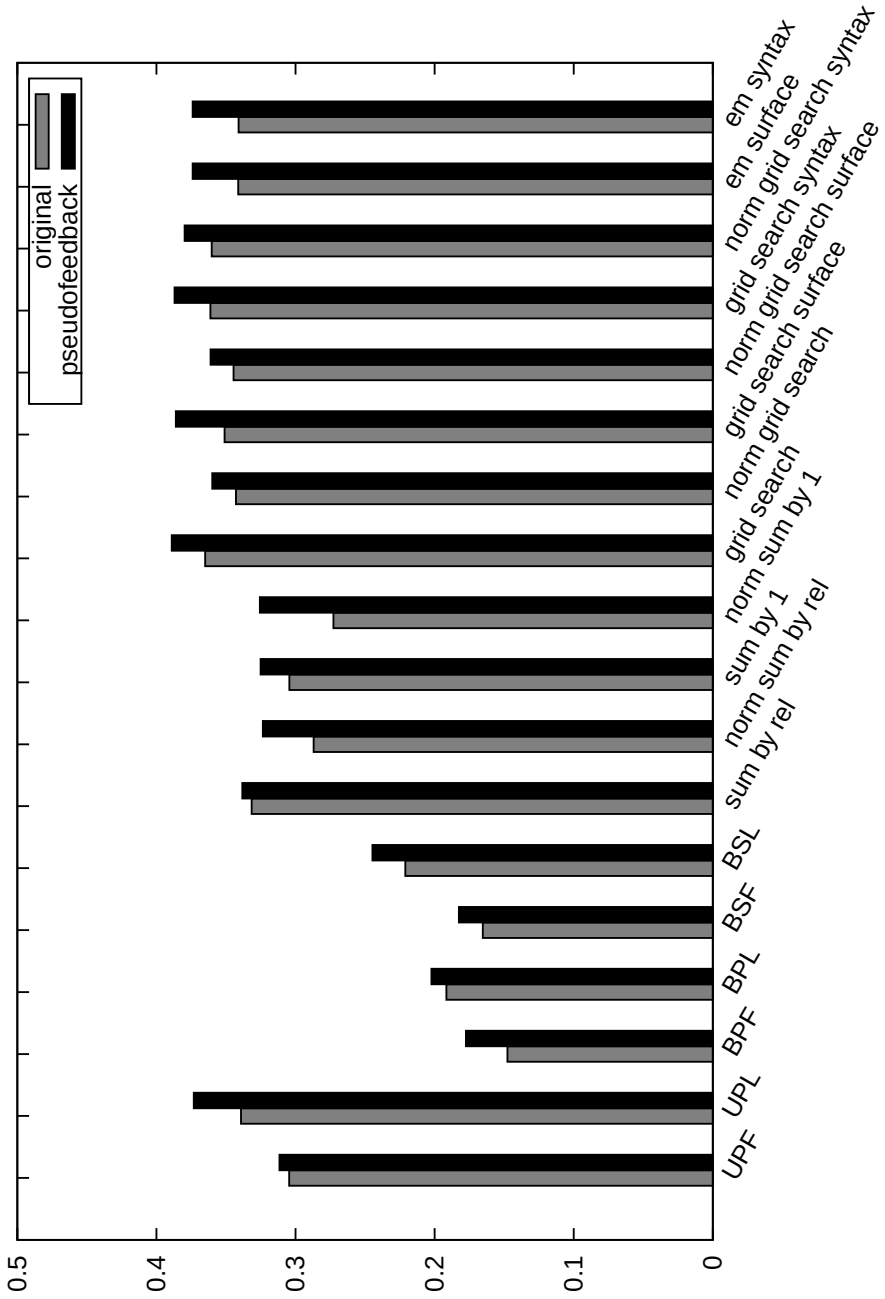
Pro toto vyhlazování nepřináší zpětná vazba (pseudo relevance feedback) žádné zlepšení, naopak zhoršuje výsledky.

5.2.7 Kombinace modelů

Obrázek 7 a tabulka 7 obsahují nejlepší dosažené výsledky této práce, a to jsou modely vzniklé lineární kombinací základních modelů UPF, UPL, BPF, BPL, BSF, BSL, kdy váhy jednotlivých modelů byly nalezeny pomocí grid search na deseti trénovacích dotazech (viz kapitola 3). Nejlepší výsledky se pohybují nad 0,38, přičemž nejlepší model, kombinace všech modelů, dosáhl MAP **0,3890** a je vyznačen tučně. Tento výsledek je získaný na testovací množině 40 témat (viz kapitola 3). Pro srovnání např. s [19] či s [4] uvádíme výsledek pro všech 50 témat: MAP = **0,4102**. Tímto výsledkem jsme pouze s použitím jazykovému modelu dosáhli lepšího výsledku než většina výsledků v [19]. Lepšího výsledku MAP = 0,4225 dosáhli autoři [4], ale i tomuto výsledku jsme se přiblížili na srovnatelnou úroveň.

Při srovnání modelů em_syntax, em_surface a grid_search na obrázku 7 nebo v tabulce 7 se ukazuje, že jednoduchá metoda hledání koeficientů pro maximalizaci MAP pomocí grid search dává stejné, dokonce lepší výsledky než EM algoritmus hledající optimální koeficienty pro minimalizaci entropie.

Obr. 7: Porovnání výsledků všech modelů bez pseudo feedbacku a s pseudo feedbackem (2 běhy, 3 dokumenty). Modely: **joint** funkce, vyhlazování **Jelinek-Mercer**, části **tdn**, **stopwords**, stemmer **light+** – grafické znázornění



Tab. 7: Porovnání výsledků všech modelů bez pseudo feedbacku a s pseudo feedbackem (2 běhy, 3 dokumenty). Modely: **joint** funkce, vyhlazování **Jelinek-Mercer**, části **tdn**, **stopwords**, stemmer **light+** – přesné hodnoty

model	původní	pseudo_feedback
UPF	0,3046	0,3116
UPL	0,3392	0,3731
BPF	0,1477	0,1775
BPL	0,1915	0,2023
BSF	0,1654	0,1826
BSL	0,2211	0,2447
sum_by_rel	0,3316	0,3382
norm_sum_by_rel	0,2870	0,3235
sum_by_1	0,3044	0,3251
norm_sum_by_1	0,2728	0,3257
grid_search	0,3650	**0,3890**
norm_grid_search	0,3428	0,3598
grid_search_surface	0,3510	0,3861
norm_grid_search_surface	0,3446	0,3611
grid_search_syntax	0,3612	0,3870
norm_grid_search_syntax	0,3603	0,3797
em_surface	0,3412	0,3739
em_syntax	0,3410	0,3740

Tab. 8: Porovnání výsledků všech modelů bez pseudo feedbacku a s pseudo feedbackem (2 běhy, 3 dokumenty). Modely: **joint** funkce, vyhlazování **Dirichlet**, λ pomocí grid-search, části **tdn**, **stopwords**, stemmer **light+**

model	původní	pseudo_feedback
UPF	0,2941	0,2441
UPL	0,3404	0,2964
BPF	0,1456	0,1018
BPL	0,2205	0,1850
BSF	0,1588	0,1540
BSL	0,2254	0,1614
grid_search	0,3578	0,2957
norm_grid_search	0,3516	0,3006
em_surface	0,2938	0,3235
em_syntax	0,2741	0,3132

5.3 Porovnání jednotlivých modelů

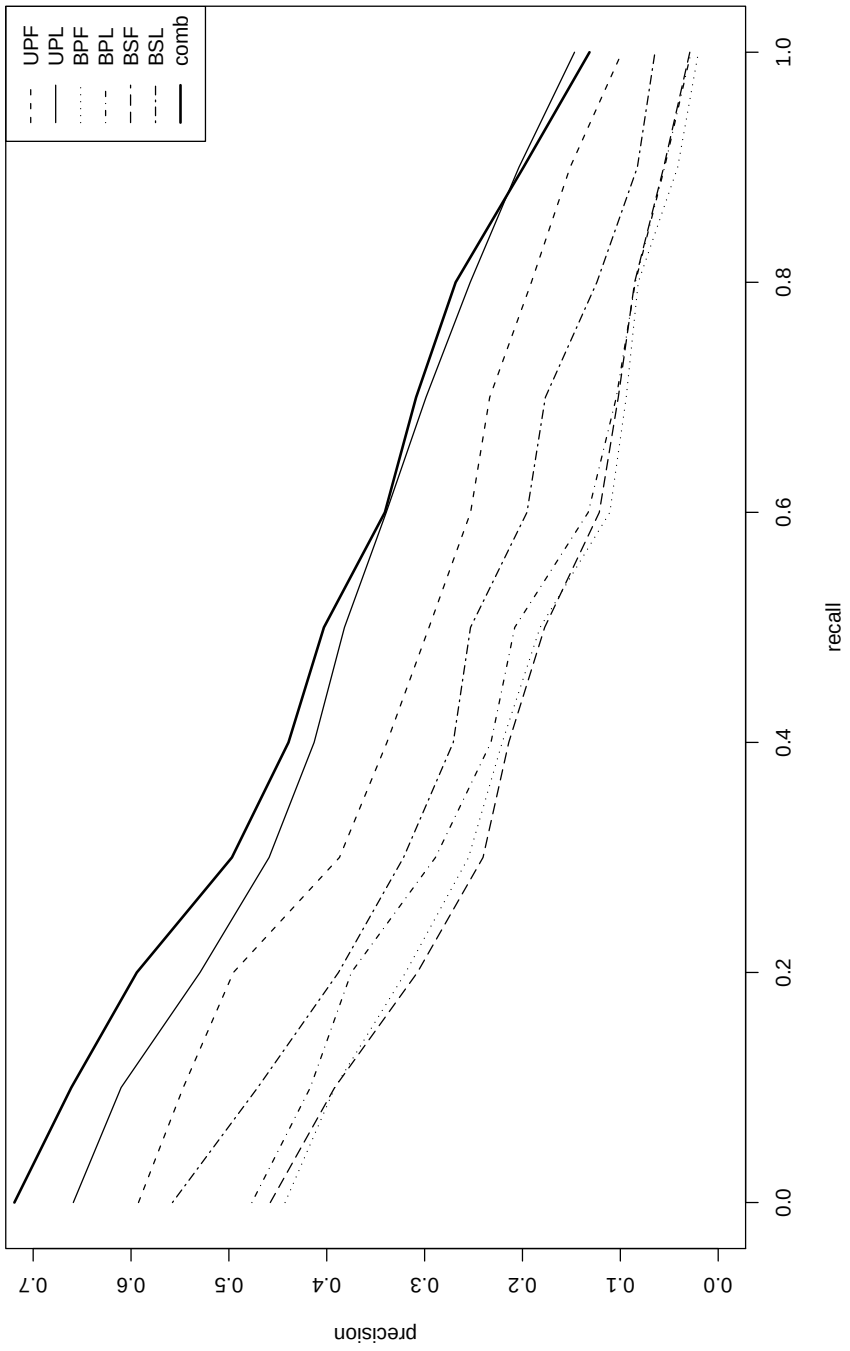
5.3.1 Precision-recall křivky

Pro zajímavost ukazujeme precision-recall křivky jednotlivých modelů na obrázku 8. Kombinovaný model vytvořený lineární kombinací všech modelů je znázorněn tučnou čarou. Zajímavé je, že vlastnosti jednotlivých modelů – precision – jsou konzistentní přes celý rozsah recall, tedy lepší modely jsou skutečně lepší na celém intervalu recall.

5.3.2 Rozdíly v AP pro jednotlivá témata

Na obrázcích 9, 10 a 11 ukazujeme podrobně rozdíly v AP pro vybrané dvojice modelů. Obrázek 9 ukazuje rozdíl modelů UPF a UPL, tedy unigramových modelů bez použití lemmatizace a s použitím lemmatizace. Obrázek 10 ukazuje rozdíly lemmatizovaných modelů unigramového a bigramového. Nejvíce

Obr. 8: Precision-recall křivky jednotlivých modelů UPF, UPL, BPF, BPL, BSF, BSL a jejich kombinace pomocí lineární kombinace a vah nalezených pomocí grid search. Nastavení modelů: **joint** funkce, vyhlazování **Jelinek-Mercer**, části **tdn**, **stopwords**, stemmer **light+**



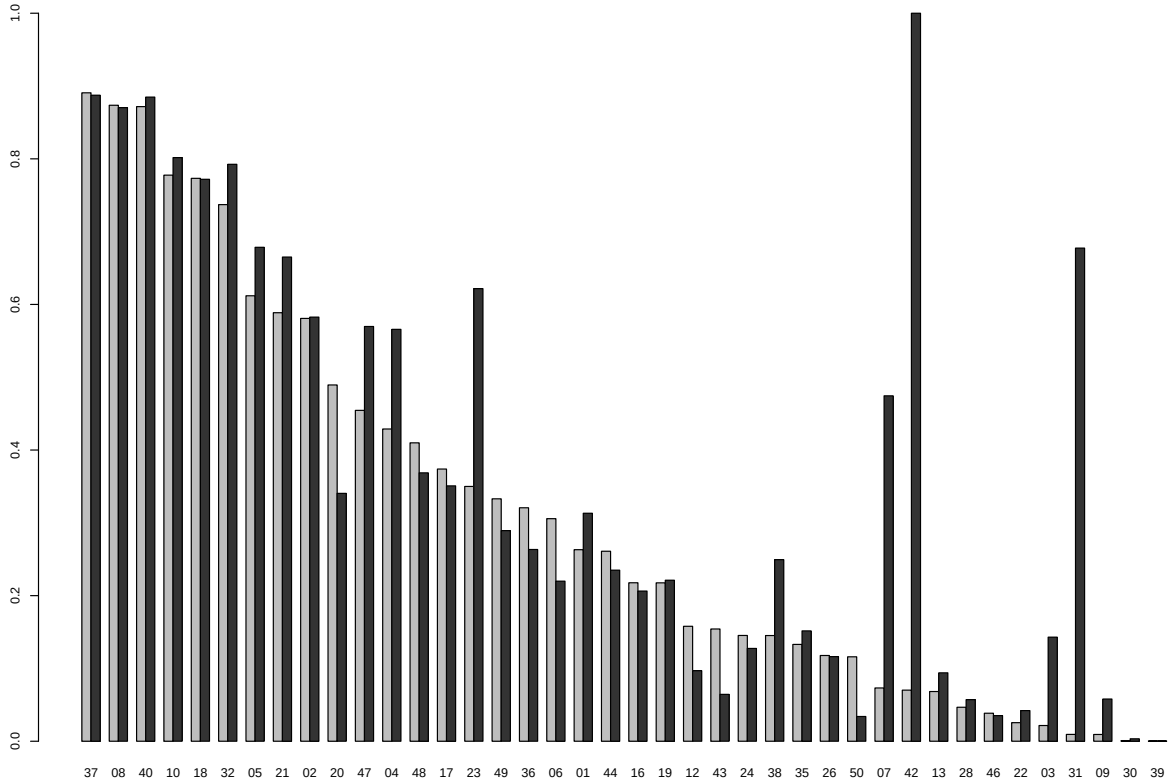
nás zajímá obrázek 11, který analyzuje rozdíl mezi povrchovým bigramovým modelem a syntaktickým bigramovým modelem.

Nejzajímavějším pozorováním je fakt, že i když se modely liší v MAP, tedy celkovém průměrném hodnocení, tento rozdíl nevznikl tak, že by se stejnou měrou lišila AP na jednotlivých dotazech. Např. u modelů BSF a BSL, u kterých se MAP liší o 0,0424, tj. o 20% (BPL = 0,2023 a BSL = 0,2447), se na některých dotazech BPL a BSL liší velmi výrazně. Podívejme se tedy, jaké jsou vlastnosti dotazů, na kterých dopadly jednotlivé modely lépe. Provedeme následující rozbor: Pro každé téma vezmeme k němu relevantní dokumenty a spočítáme, jak jsou v těchto relevantních dokumentech zastoupeny termy příslušné danému modelu, čili jak se v relevantních dokumentech vyskytujíc unigramové formy, unigramová lemmata, bigramy povrchové a bigramy syntaktické. Můžeme spočítat i jejich přispění k celkovému ohodnocení relevance dokumentu. Tyto výsledky pak musíme ručně vyhodnotit. Analýzu jsme prováděli na modelech se sdruženou pravděpodobností, s vyhlazováním Jelinek-Mercer, s použitím lemmat, se stemmerem `light+`, se stopwords, jednoduše s nejlepším možným nastavením z obrázku 7, resp. tabulky 7.

5.3.3 Model s formami vs. model s lemmaty

Graf 9 ukazuje velké rozdíly v AP mezi unigramovým modelem s formami (UPF, tmavě) a unigramovým modelem s lemmaty (UPL, světle) pro jednotlivé dotazy. Například pro téma číslo 38 s titulem „Výzkum rakoviny“ přispěly k ohodnocení relevance dokumentu lemmata „léčba“, „rakovina“, „látka“, „medikament“, „zhoubný“, „přístroj“. Tato slova se však v zadání tématu objevovala ve formách, které se v relevantních dokumentech neobjevily, např. „léčby“, „rakoviny“, „rakovinou“, „látek“, „zhoubného“, „přístrojů“.

Obr. 9: Porovnání modelů UPF (tmavě) a UPL (světle) na jednotlivých dotazech



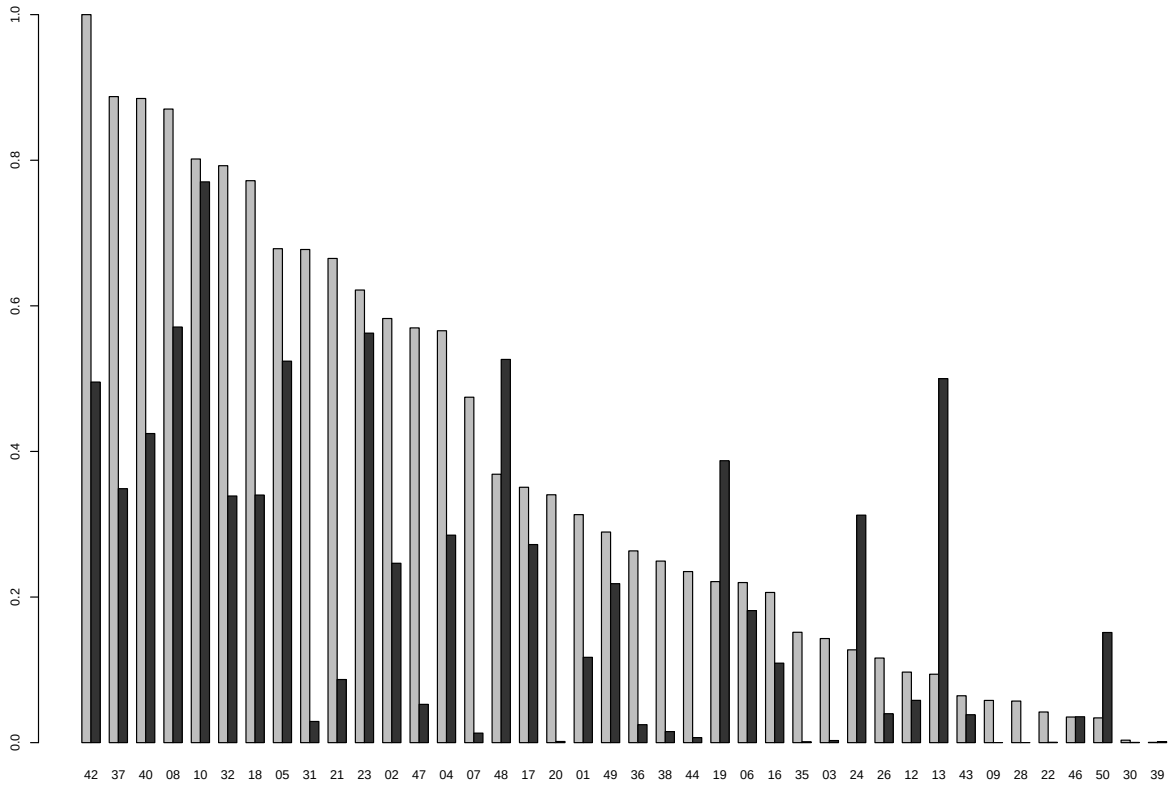
5.3.4 Unigramový vs. bigramový model

Při analýze výsledků bigramového modelu se ukazuje, že jeho největší problém je velké množství bigramů, které nemají pro vyhledávání žádný smysl. Například v tématu číslo 37 s titulkem „Neregulérní audity v Enronu“ se v bigramech používaných pro vyhledávání objevují bigramy jako „audit, v“, „v, Enron“, „zabývat, se“ (pracujeme s lemmaty). Najdeme samozřejmě i bigramy subjektivně „smysluplné“, jako „Enron bankrot“ či „účetní firma“, ale většina bigramů obsahuje stopwords či jsou pro vyhledávání zbytečné. Naopak uni-

gramový model, který používá jednotlivá slova, správně použije samostatné unigramy „firma“, „Enron“, „účetní“, „odpovědnost“.

Naopak v několika dotazech pomáhá použití bigramu najít souvislosti mezi slovy. Například v tématu číslo 13 s titulkem „Snižování rizika onemocnění cukrovkou“ dává bigramový model správně do souvislosti slova „onemocnění“ a „cukrovka“ a vyhledává dokumenty obsahující bigram „onemocnění cukrovka“.

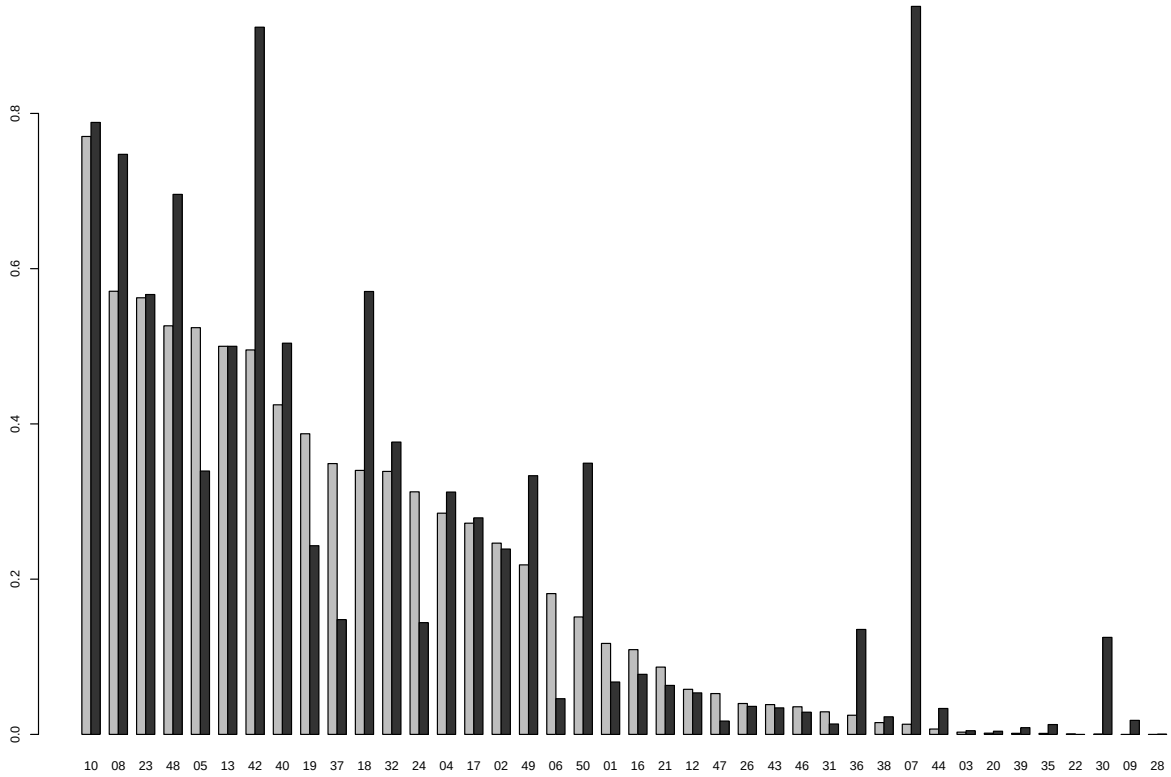
Obr. 10: Porovnání modelů UPL (tmavě) a BPL (světle) na jednotlivých dotazech



5.3.5 Povrchový vs. syntaktický model

Například pro téma číslo 7 s titulkem „Australský premiér“, na kterém vznikl největší rozdíl: Ohodnocení relevance dokumentů se účastnily povrchové bigramy „být v“, „být kdo“, „, který“, „australský premiér“, „být který“, kdežto v syntaktickém modelu pouze bigramy „být v“, „být kdo“, „australský premiér“. V této souvislosti je třeba poznamenat, že stopwords pracují na slovních formách, čili například nevhodný bigram „, který“ se do výběru dostal proto, že v původní formě vypadal „, kterých“ a forma „kterých“ na seznamu 256 stopwords z CLEF ([1]) nebyla. Podobně ostatní nevhodné povrchové bigramy. Našli jsme tedy jednu výhodu syntaktického modelu oproti povrchovému. V povrchovém modelu se vedle sebe ocitají slova, která k sobě (sémanticky, subjektivně) nepatří a velmi často se jedná o stopwords. Najít perfektní seznam stopwords je náročné a vyžaduje mnoho ruční práce, kdežto syntaktický model implicitně dává dohromady bigramy, které k sobě jednak patří, jednak se model vyhýbá stopwords bigramům. Jestliže se v syntaktickém modelu objeví dvě slova jako bigram, máme větší naději, že tato slova k sobě patří, než v povrchovém modelu. To lze vysledovat i na výsledcích v obrázku 5 v prvním řádku, kde nebyla použita žádná metoda určování důležitosti termu. Zde syntaktický model BSL dosáhl mírně lepšího výsledku (MAP = 0,1986) než povrchový model BPL (MAP = 0,1876).

Obr. 11: Porovnání modelů BPL (tmavě) a BSL (světle) na jednotlivých dotazech



6 Závěr

V předložené práci jsme matematicky ukázali, jak jednoduše zavést syntaxi do jazykového modelu a jak používat syntaktický jazykový model s lemmatizací, stemmingem a metodami pro rozšiřování dotazu (pseudo relevance feedback). Tento model jsme implementovali a experimentálně jsme jej porovnali s unigramovým a bigramovým povrchovým modelem.

Porovnali jsme také výsledky modelů při použití stemmingu a lemmatizace. Potvrdili jsme užitečnost lemmatizace a ukázali, že funguje stejně dobře jako stemming. Věnovali jsme se i použití různých metod vyhlazování a použití pseudo relevance feedback jako metody rozšiřování dotazu.

Vyzkoušeli jsme několik přístupů ke kombinování jednotlivých modelů. Pouze s použitím kombinací jazykových modelů jsme dosáhli výsledku srovnatelného s výsledky účastníků Cross Language Evaluation Forum 2007 Ad-Hoc track ([1], [19]), ve většině případů dokonce výsledku lepšího. Lepší výsledky uvedli pouze autoři [4], ale i těmto výsledkům jsme se přiblížili na srovnatelnou úroveň. Porovnáváme-li pouze jazykový model, pak jsme jazykový model oproti nejlepším známým výsledkům na této kolekci uvedeným v [4] výrazně zlepšili.

Použité modely jsme podrobně popsali a analyzovali jejich výsledky. Ukázali jsme, že modely, které se vzájemně liší v celkovém průměrném hodnocení přes všechny dotazy, mohou vykazovat velmi rozdílné výsledky pro jednotlivé dotazy. Ukázali jsme takové příklady pro vybrané dvojice modelů a tyto rozdílné výsledky jsme okomentovali. Konečně jsme našli příklady, kdy použití syntaktického jazykového modelu bylo výhodnější než použití klasického bigramového modelu.

Literatura

[1] Cross Language Evaluation Forum (CLEF), http://clef-campaign.org.

[2] trec_eval, http://trec.nist.gov/trec_eval.

[3] Berger, A.; Lafferty, J.: Information retrieval as statistical translation. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: ACM, 1999, ISBN 1-58113-096-1, s. 222–229, doi: http://doi.acm.org/10.1145/312624.312681.

[4] Dolamic, L.; Savoy, J.: Stemming Approaches for East European Languages. 2008: s. 37–44.

[5] Fellbaum, C.: *WordNet: An Electronical Lexical Database*. Cambridge, MA: The MIT Press, 1998.

[6] Fuhr, N.: Probabilistic models in information retrieval. *Comput. J.*, ročník 35, č. 3, 1992: s. 243–255, ISSN 0010-4620, doi: http://dx.doi.org/10.1093/comjnl/35.3.243.

[7] Gao, J.; Nie, J.-Y.; Wu, G.; aj.: Dependence language model for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: ACM, 2004, ISBN 1-58113-881-4, s. 170–177, doi:http://doi.acm.org/10.1145/1008992.1009024.

[8] Hajič, J.: *Disambiguation of Rich Inflection (Computational Morphology of Czech)*, ročník 1. Prague: Charles University Press, 2004.

[9] Jelinek, F.: *Statistical Methods for Speech Recognition*. The MIT Press, January 1998, ISBN 0262100665.

[10] Jelinek, F.; Mercer, R. L.: Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands: North-Holland, Květen 1980.

[11] Lease, M.; Charniak, E.: A Dirichlet-Smoothed Bigram Model for Retrieving Spontaneous Speech. 2008: s. 687–694.

[12] Lee, C.; Lee, G. G.: Probabilistic information retrieval model for a dependency structured indexing system. *Inf. Process. Manage.*, ročník 41, č. 2, 2005: s. 161–175, ISSN 0306-4573, doi: http://dx.doi.org/10.1016/j.ipm.2003.11.001.

[13] Mackay, D. J. C.; Petoy, L. C. B.: A Hierarchical Dirichlet Language Model. *Natural Language Engineering*, ročník 1, 1995: s. 1–19.

[14] Manning, C. D.; Raghavan, P.; Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, 2008, ISBN 9780521865715.

[15] Manning, C. D.; Schutze, H.: *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[16] McDonald, R.; Pereira, F.; Ribarov, K.; aj.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Human Language Technologies and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, Vancouver, Canada, 2005.

[17] Miller, D. R. H.; Leek, T.; Schwartz, R. M.: A hidden Markov model information retrieval system. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: ACM, 1999, ISBN 1-58113-096-1, s. 214–221, doi:http://doi.acm.org/10.1145/312624.312680.

[18] Nallapati, R.; Allan, J.: Capturing term dependencies using a language model based on sentence trees. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, New York, NY, USA: ACM, 2002, ISBN 1-58113-492-4, s. 383–390, doi:http://doi.acm.org/10.1145/584792.584855.

[19] Nunzio, G. M.; Ferro, N.; Mandl, T.; aj.: CLEF 2007: Ad Hoc Track Overview. 2008: s. 13–32.

[20] Ponte, J. M.; Croft, W. B.: A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*,

New York, NY, USA: ACM, 1998, ISBN 1-58113-015-5, s. 275–281, doi: http://doi.acm.org/10.1145/290941.291008.

[21] Porter, M. F.: An algorithm for suffix stripping. 1997: s. 313–316.

[22] van Rijsbergen, C. J.: A non-classical logic for information retrieval. 1997: s. 268–272.

[23] Robertson, S. E.; van Rijsbergen, C. J.; Porter, M. F.: Probabilistic models of indexing and searching. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, Kent, UK, UK: Butterworth & Co., 1981, ISBN 0-408-10775-8, s. 35–56.

[24] Salton, G.; Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Technická zpráva, Ithaca, NY, USA, 1987.

[25] Salton, G.; Buckley, C.: Improving retrieval performance by relevance feedback. 1997: s. 355–364.

[26] Žabokrtský, Z.; Ptáček, J.; Pajas, P.: TectoMT: Highly Modular MT system with tectogrammatics used as a transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, Association for Computational Linguistics, June 2008, s. 167–170.