AXBOROT IZLASH TIZIMLARIDA INDEKSLAR: TUZILISHI, SAMARADORLIGI VA TAJRIBAVIY TAHLILI

Mirzaaxmedov MuhammadBobur Karimberdiyevich[1], Jo‘rayev Muhammadjon Durbek ogli[2]

[1]Zahiriddin Muhammad Bobur nomidagi Andijon davlat universiteti,
Kompyuter injiniringi kafedrasi dotsenti., t.f.f.d, Andijon, O‘zbekiston
[2]Zahiriddin Muhammad Bobur nomidagi Andijon davlat universiteti,
Kompyuter injiniringi kafedrasi o‘qituvchisi, Andijon, O‘zbekiston
E-mail: muhammadzohir1110@gmail.com, jmd@gmail.com

K E Y W O R D S

axborot izlash, teskari indeks, pozitsion indeks, tokenizatsiya, Boolean qidiruv, BM25, indeksni siqish, normalizatsiya

A B S T R A C T

Maqolada axborot izlash tizimlarida teskari va pozitsion indekslarning vazifasi, tuzilishi hamda samaradorligi nazariy va tajribaviy jihatdan o‘rganildi. Tadqiqotning maqsadi indeks turini tanlashning qidiruv vaqti, indeksning kodlangan hajmi va aniq ibora talabini bajarishga ta’sirini ajratib ko‘rsatishdir. Python yordamida sakkiz mavzuli, 6 000 hujjat va 852 277 tokendan iborat sun’iy korpus yaratildi. Uning 1 000, 3 000 va 6 000 hujjatli qismlarida 32 ta Boolean hamda 8 ta iboraviy so‘rov tekshirildi. Eng katta korpusda teskari indeks bo‘yicha qidiruv medianasi 0,0167 ms, hujjatlarning atamalar to‘plamini ketma-ket tekshirishda esa 0,3893 ms bo‘ldi; medianalar nisbati 23,25 martani tashkil etdi. Pozitsion indeks iboraviy qidiruvda 26,21 martalik tezlashishni ko‘rsatdi. Hujjat identifikatorlarini farqlar orqali Variable Byte usulida kodlash postings ma’lumotlari hajmini 74,66 foizga kamaytirdi. Qo‘shimcha misollarda apostrof variantlarini birxillashtirish va BM25 orqali tartiblash ko‘rsatildi. Natijalar nazoratli sun’iy korpus hamda bir dasturiy muhitga tegishli bo‘lib, real foydalanuvchi relevantligi yoki sanoat tizimining unumdorligini bevosita ifodalamaydi.

KIRISH (INTRODUCTION)

Elektron kutubxona yoki ta’lim platformasida hujjatni saqlashning o‘zi yetarli emas: foydalanuvchi talabiga mos yozuvlarni ajratish mexanizmi ham kerak. Har bir so‘rovda barcha hujjatlarni qayta tekshirish o‘rniga, atama qaysi hujjatlarda uchrashini oldindan qayd etish mumkin. Teskari indeks aynan shunday bog‘lanishni tashkil etadi; uning asosiy qismlari atamalar lug‘ati va hujjat identifikatorlari ro‘yxatidir [1, 1-bob; 2].

Indeks samaradorligini faqat tezlik bilan baholash yetarli emas. Masalan, “axborot izlash” iborasini qidirish so‘zlarning alohida mavjudligini emas, ketma-ket kelishini talab qiladi. Shuningdek, bir atamaning turlicha yozilishi uni bir nechta mustaqil kalitga ajratishi mumkin. Demak, qidiruv sharti, matnni tayyorlash qoidalari va indeksda saqlanadigan ma’lumot o‘zaro mos bo‘lishi kerak.

Mazkur ish uch savolga javob izlaydi: indeks qidiruv vaqtini qanchaga qisqartiradi; pozitsiyalarni saqlash qanday imkoniyat va xarajat keltiradi; normalizatsiya bilan indeks tuzilmasining ta’sirini qanday ajratish mumkin? Ishning hissasi yangi qidiruv algoritmini taklif etish emas, balki ushbu savollarni yagona takrorlanadigan tajribada va hisoblash misollarida tekshirishdir.

Indekslarning nazariy asoslari va tasnifi

(cc) BY

Indeks matnning o‘zi emas, undan hosil qilingan qidiruv ma’lumot tuzilmasidir. To‘g‘ri indeksda hujjat kalit bo‘lib, unga tegishli atamalar qayd etiladi. Teskari indeksda esa atama kalit vazifasini bajaradi va shu atama uchragan hujjatlar ro‘yxatini ko‘rsatadi. Bitta atama-hujjat bog‘lanishi postings yozuvi deb yuritiladi. Yozuvga uchrash chastotasi yoki matndagi pozitsiyalar ham qo‘shilishi mumkin [1, 2].

1-jadval.

Indekslar va ular qo‘llab-quvvatlaydigan asosiy vazifalar

Indeks turi	Saqlanadigan axborot	Qo‘llanishi va chegarasi
To‘g‘ri indeks	Hujjat -> atamalar yoki chastotalar	Hujjatni tavsiflash; barcha hujjatlar bo‘ylab tekshirish zarur bo‘lishi mumkin.
Teskari indeks	Atama -> hujjat identifikatorlari	AND/OR qidiruvi; pozitsiyasiz holda aniq iborani tasdiqlamaydi.
Pozitsion indeks	Atama -> hujjat va pozitsiyalar	Ibora va yaqinlik qidiruvi; qo‘shimcha saqlash xarajati mavjud.
Belgili k-gram indeks	Belgi bo‘laklari -> atamalar	Imlo yoki shablon bo‘yicha nomzod atamalarni topish
Maydonli indeks	Sarlavha, matn va boshqa maydonlar	Qidiruvni maydon bilan cheklash yoki vaznlash
Vektor indeksi	Vektorlar va ularning qidiruv tuzilmasi	Yaqin vektorlarni topish; HNSW shunday usullardan biri [4].

Bu tasnifdagi ayrim belgilar bir-birini inkor etmaydi. Masalan, indeks bir vaqtning o‘zida teskari, pozitsion va maydonli bo‘lishi mumkin. Shuning uchun “eng yaxshi indeks qaysi?” savoli o‘rniga, “qaysi ma’lumotlar qaysi so‘rovni bajarish uchun zarur?” savolini qo‘yish aniqroqdir. Belgili k-gram indeksini so‘zlarning ketma-ket juftliklarini saqlovchi biword indeks bilan aralashtirmaslik kerak.

Tartiblangan ikki postings ro‘yxatini birlashtirib tekshirishning oddiy xarajati ularning uzunliklari yig‘indisiga bog‘liq. Kam uchraydigan atamadan boshlash nomzod hujjatlar sonini erta qisqartiradi [1]. Biroq “tadqiqot” yoki “tizim” kabi juda ko‘p hujjatda uchraydigan kalitlarda ham katta ro‘yxatlar tekshiriladi. Demak, indeks barcha so‘rovni doimiy vaqtda bajaradi, degan qarash asosli emas.

Indekslash va relevantlik bo‘yicha tartiblash ham alohida tushunchalardir. Indeks hisoblash uchun zarur nomzodlar va statistikani yetkazadi, BM25 kabi model esa ularga ball beradi [3]. Vektor indeksining mavjudligi ham mazmuniy moslikni o‘z-o‘zidan kafolatlamaydi: natija vektorlashtirish modeliga va baholash vazifasiga bog‘liq. BEIR tadqiqoti turli domenlarda tekshirish zarurligini ko‘rsatadi [5].

MATERIALLAR VA METODLAR

Tajriba korpusi va nazorat shartlari

Tajriba uchun mavjud tashkilot hujjatlari olinmadi. O‘zbekcha atamalar va texnik belgilardan tashkil topgan sun’iy token ketma-ketliklari yaratildi. Sakkiz mavzu tanlandi: axborot izlash, mashinali o‘rganish, ma’lumotlar bazasi, kompyuter tarmoqlari, bulutli hisoblash, axborot xavfsizligi, raqamli ta’lim va dasturiy injiniring. Har bir mavzu eng katta korpusda 750 tadan hujjat bilan ifodalandi. Hujjatlar tabiiy maqola sifatida emas, algoritmlarni nazoratli taqqoslash uchun yaratilgan birliklar sifatida qaraldi.

Tasodifiy generatorning boshlang‘ich qiymati 20260924 etib belgilandi. Dastlabki hujjat uzunligi 100-180 token oralig‘ida tanlandi. Har bir token uchun mavzuviy lug‘atdan tanlash ehtimoli 0,65, umumiy atamalardan 0,20, 800 ta “atama0000” shaklidagi sun’iy kalitdan 0,15 bo‘ldi. So‘ng

(cc) BY

mavzuning ikki tayanch atamasi yonma-yon, bir so‘z bilan ajratilgan yoki teskari tartibda qo‘shildi. Uch holat navbat bilan takrorlandi; bu ibora nazoratini aniq qurishga imkon berdi.

2-jadval.

Tajribaning asosiy parametrlari

Parametr	Qiymat yoki belgilangan tartib
Korpus hajmlari	1 000; 3 000; 6 000 hujjat; bitta korpusning ichma-ich qismlari
Eng katta korpus	852 277 token; 976 noyob atama; o‘rtacha 142,05 token/hujjat
Hujjat uzunligi	Tayanchlar qo‘shilgach 102-183 token
Boolean so‘rovlar	32 ta: tayanch, sohaviy, umumiy va siyrak guruhlarning har birida 8 tadan
Ibora va xato nazorati	8 ta ibora; vaqt hisobidan tashqari 4 ta mavjud bo‘lmagan kalitli so‘rov
Muhit	CPython 3.13.5; Linux x86-64; AMD EPYC 9V74 ko‘rsatilgan umumiy hisoblash muhiti
Takrorlash	Qidiruv: 7 davra; Boolean uchun har o‘lchovda 5 chaqiruv; indeks qurish: 5 marta

Matn va so‘rovga bir xil tayyorlash qo‘llandi: kichik harfga o‘tkazish, NFC normalizatsiyasi va apostrof variantlarini yagona belgiga xaritalash. NFC kanonik ekvivalent belgilarni birxillashtiradi [7]; tipografik apostroflar uchun esa alohida xaritalash ishlatildi. Stop-so‘zlar olib tashlanmadi, stemming va lemmatizatsiya qo‘llanmadi. Tokenizator lotin harflari, raqamlar va so‘z ichidagi apostroflarni qamrab oldi; kirill yozuvini qayta ishlash sinov doirasiga kirmadi.

Korpus va so‘rovlar tajriba boshlanishidan oldin belgilandi. Unumdorlik natijasiga qarab so‘rov chiqarib tashlanmadi. Har uch hajmda bir xil 32 so‘rov bajarildi. To‘rtta manfiy nazorat faqat to‘g‘rilik uchun tekshirildi, shuning uchun bo‘sh natijali sun’iy kalitlar median tezlikni yaxshilash vositasi bo‘lmadi.

Qidiruv algoritmlari va o‘lchash protokoli

Boolean bazaviy usul har hujjat uchun oldindan tayyorlangan atamalar to‘plamida so‘rov atamalari mavjudligini tekshiradi. U har safar matnni qayta tokenizatsiya qilmaydi va “indekssiz xom matn skaneri” deb talqin qilinmaydi. Teskari indeksli usul atamaga tegishli Python set to‘plamlarini uzunligi bo‘yicha tartiblab kesishadi. Har ikkala usulning natijasi hujjat identifikatorlari to‘plamidir; ularni ekranga chiqarish va tartiblash o‘lchanmadi.

$$R(q) = P(t_1) \cap P(t_2) \cap \ldots \cap P(t_k) \quad (1)$$

Bu yerda P(t) atama uchragan hujjatlar to‘plamidir. Ibora nazoratida bazaviy usul barcha token ketma-ketliklarini tekshiradi. Pozitsion usul esa avval nomzod hujjatlarni ajratib, ketma-ket atamalar pozitsiyalari orasidagi farq 1 ga tengligini tekshiradi. Har bir so‘rovda ikki usul javoblari avtomatik assert sharti bilan solishtirildi. Bu 144eying144n relevantlik emas, aynan belgilangan qidiruv sharti bo‘yicha to‘g‘rilik nazoratidir.

O‘lchashdan oldin har bir so‘rov bir marta bajarildi. Yetti davrada so‘rov tartibi 144eying144nistic aralashtirildi, usullarni o‘lchash navbati almashtirildi. Boolean so‘rov besh chaqiruvli paketda, ibora esa bir chaqiruvda o‘lchandi. Perf_counter_ns() vaqt farqi millisekundga aylantirildi [8]. Avval har so‘rovning 144eyi natijasi medianasi, so‘ng so‘rovlar bo‘yicha mediana, kvartillar va P95 hisoblandi. Qidiruv o‘lchovlarida Pythonning davriy xotira tozalagichi vaqtincha o‘chirildi. P95 xizmatdagi real so‘rovlar oqimining emas, ushbu test to‘plamining tavsifidir.

$$S = T(scan)/T(indeks); \quad C = 100[1 - B(VB)/B(32)] \quad (2)$$

S tezlashish koeffitsiyenti, C baytlardagi hajm kamayishidir. Siqishda tartiblangan docID farqlari Variable Byte bilan kodlandi [1]. Taqqoslash 32

(cc) BY

bitli butun sonlar bilan ifodalangan postings yuklamasiga nisbatan bajarildi. Lug‘at, fayl sarlavhasi va Python obyektlarining operativ xotira xarajati bu hajmlarga kiritilmadi. Kodlashdan keyingi qiymatlar qayta ochilib, dastlabki qiymatlar bilan tekshirildi.

BM25 va aniqlik mezonlari

$$K(d) = k_1[1 - b + b|d|/avgdl] \quad (3)$$

$$IDF(t) = ln[1 + (N - df(t) + 0.5)/(df(t) + 0.5)] \quad (4)$$

$$BM25(d,q) = \sum_{t \in q} IDF(t) \cdot tf(t,d)(k_1 + 1)/(tf(t,d) + K(d)) \quad (5)$$

BM25 misolida k1 = 1,2 va b = 0,75 olindi; musbat IDF shakli Lucene hujjatida berilgan [3; 6]. N hujjatlar soni, df atama uchragan hujjatlar soni, tf uning hujjatdagi chastotasi, avgdl o‘rtacha uzunlikdir. P = TP/(TP + FP), R = TP/(TP + FN), F1 = 2PR/(P + R) formulalari nazorat misollariga qo‘llandi [1]. Ularda haqiqiy foydalanuvchi baholari emas, oldindan belgilangan aniq moslik etaloni ishlatildi.

NATIJALAR

Boolean qidiruv tezligi va so‘rov tarkibining ta’siri

Teskari indeks va ketma-ket tekshirish 32 asosiy hamda 4 manfiy so‘rovning har birida bir xil natija qaytardi. Uch korpus hajmida jami 108 ta to‘plam tengligi tekshiruvi muvaffaqiyatli o‘tdi. 3-jadval barcha asosiy so‘rovlar bo‘yicha vaqt medianalarini ko‘rsatadi. Barcha sonlar ushbu ishda bajarilgan hisoblash tajribasidan olindi; tashqi tizimning e’lon qilingan ko‘rsatkichlari emas.

3-jadval.

Boolean so‘rovlar bo‘yicha vaqt natijalari

Hujjat soni	Skan mediana, ms	Teskari mediana, ms	Skan P95, ms	Teskari P95, ms	Tezlashish
1 000	0,0441	0,0039	0,0768	0,0144	11,35 marta
3 000	0,1654	0,0115	0,3919	0,0507	14,32 marta
6 000	0,3893	0,0167	0,9685	0,1074	23,25 marta

Eng katta korpusda vaqt 0,3893 ms dan 0,0167 ms gacha, ya’ni 95,70 foizga kamaydi. Skan usulining 25-75 foizlik kvartil oralig‘i 0,3509-0,5866 ms, teskari usulniki 0,0140-0,0379 ms bo‘ldi. Tezlashish jadvaldagi yaxlitlangan sonlardan emas, to‘liq aniqlikdagi medianalar nisbatidan hisoblandi. Mediana bilan P95 orasidagi farq ayrim so‘rovlar boshqalariga qaraganda ancha ko‘p hujjatni qamrab olishiga mos keladi.

4-jadval.

6 000 hujjatda so‘rov guruhlari kesimidagi natijalar

So‘rov guruhi	Natija soni medianasi	Skan, ms	Teskari, ms	Tezlashish
Tayanch atamalar	750	0,3764	0,0166	22,63 marta
Sohaviy atamalar	731,5	0,3883	0,0168	23,09 marta
Umumiy atamalar	4 063,5	0,9250	0,1013	9,13 marta
Siyrak aralash kalit	19,5	0,3201	0,0064	49,88 marta

(cc) BY

Izoh: har bir guruhda 8 ta so‘rov bor. Juft sondagi qiymatlar medianasi kasrli bo‘lishi mumkin; 731,5 hujjat alohida so‘rov javobi emas.

Siyrak guruhda qisqa postings to‘plamlari sababli tezlashish 49,88 martaga yetdi. Umumiy atamalarda esa natijalar medianasi korpusning qariyb uchdan ikki qismini qamrab oldi va tezlashish 9,13 marta bo‘ldi. Bu tafovut “indeks 23 marta tez” degan umumlashtirishdan ko‘ra muhimroq: indeks foydasi so‘rov selektivligiga, ya’ni u ajratadigan hujjatlar ulushiga bog‘liq.

Korpus hajmi olti marta oshganda skan vaqti 8,84 marta, teskari indeks vaqti 4,31 marta o‘sdi. Ushbu nisbatlarni algoritmning aniq darajali o‘sish qonuni sifatida qabul qilib bo‘lmaydi. Uch nuqta va bitta hisoblash muhiti bunday xulosa uchun yetarli emas; natijaga obyekt joylashuvi, protsessor keshi va umumiy muhit yuklamasi ham ta’sir qilishi mumkin.

Iboraviy qidiruv va yozuv variantlarini birxillashtirish

Pozitsion indeks sakkiz iboraning har birida to‘liq token tekshiruvi bilan bir xil hujjatlarni topdi. Uch hajmda jami 24 ta ibora to‘plami tengligi tekshirildi. 6 000 hujjatda har bir tayanch juftlik 750 hujjatda uchradi, ammo ulardan faqat 250 tasida so‘zlar so‘ralgan tartibda yonma-yon joylashdi. Bu nisbat korpus generatorida oldindan qurilgan uch holatning natijasidir.

5-jadval.

Aniq iboralarni qidirish vaqti va tekshiruv natijasi

Hujjatlar	To‘liq skan, ms	Pozitsion, ms	Tezlashish	Har iborada AND / aniq ibora
1 000	4,2678	0,1421	30,03 marta	125 / 42
3 000	13,2433	0,5080	26,07 marta	375 / 125
6 000	26,7395	1,0202	26,21 marta	750 / 250

Masalan, “axborot izlash” uchun oddiy AND kesishuvi qaytargan 750 hujjatning 500 tasi ibora shartini bajarmaydi. Aniq ibora etaloniga nisbatan uning precision qiymati 250/750 = 0,3333, recall qiymati esa 1 ga teng. Pozitsion tekshiruvda ushbu etalon bo‘yicha ikkala mezon ham 1 ga teng bo‘ldi. Bu AND algoritmidagi xato emas: AND va ibora turli so‘rov semantikasidir. Teskari indeksdan keyin asl matnni tekshirish ham to‘g‘ri javob berishi mumkin, lekin uning vaqti ushbu tajribada alohida o‘lchanmadi.

Eng katta korpusda pozitsion usul medianasi 1,0202 ms, P95 qiymati 1,1116 ms bo‘ldi; to‘liq skanda tegishli qiymatlar 26,7395 va 28,3795 ms ni tashkil etdi. Pozitsion qidiruvni Boolean jadvalidagi tezlik bilan bevosita solishtirish to‘g‘ri emas, chunki u nomzodlardan tashqari atamalar tartibini ham tekshiradi.

Normalizatsiya alohida olti hujjatli nazoratda tekshirildi. Dastlabki to‘rtta hujjatda o‘qituvchi va ta’lim atamalari U+0027, U+2018, U+2019, U+02BB yoki U+02BC apostrof belgilarining turli kombinatsiyalarida yozildi. Qolgan ikkita hujjat to‘liq so‘rovga mos emas edi. Har ikki tajribada bir xil teskari indeks algoritmi ishlatilib, faqat apostrofni xaritalash qoidasi o‘zgartirildi.

(cc) BY

6-jadval.

Apostrof variantlari bo‘yicha nazoratli moslik natijalari

Tayyorlash holati	Topilgan mos hujjat	Precision	Recall	F1
NFC va kichik harf; xaritalashsiz	1 / 4	1,00	0,25	0,40
NFC, kichik harf va apostrof xaritalash	4 / 4	1,00	1,00	1,00

Recall 25 foizdan 100 foizga ko‘tarildi, chunki yozuv variantlari bitta kalitga birlashtirildi. Bu natija butun o‘zbek tili uchun aniqlik bahosi emas, aynan to‘rtta mos yozuvga oid nazoratdir. Apostrofni olib tashlash emas, almashtirish qo‘llangani muhim: “talim” shakli avtomatik ravishda “ta’lim” bilan tenglashtirilmadi. Morfologik shakllar va sinonimlar esa ushbu qoida bilan hal qilinmaydi.

Indeks qurish xarajati va postings hajmi

Indeks qidiruvni tezlashtirsa-da, uni oldindan qurish talab etiladi. 7-jadval tokenizatsiya tugagandan keyingi tuzilmalarni yaratish vaqtini ko‘rsatadi. Korpusni yaratish, faylni o‘qish va siqish vaqti bu natijalarga kiritilmagan. Har bir qiymat besh alohida qayta qurishning medianasidir.

7-jadval.

Ma’lumot tuzilmalarini qurish median vaqti

Hujjatlar	To‘g‘ri atamalar to‘plami, ms	Teskari indeks, ms	Pozitsion indeks, ms
1 000	2,551	8,995	24,554
3 000	11,571	27,466	72,189
6 000	23,613	60,031	337,707

6 000 hujjatda pozitsion indeksni qurish teskari indeksga nisbatan 5,63 marta uzoqroq davom etdi. Buning uchun nafaqat 332 755 ta atama-hujjat bog‘lanishi, balki 852 277 ta pozitsiya ham saqlandi. Pozitsion qurishning 3 000 dan 6 000 hujjatga o‘tishda keskinroq o‘sishi ma’lumot tuzilmalari va xotira ajratish xarajatlarini ham tekshirish kerakligini ko‘rsatadi; bu yagona o‘lchov asosida umumiy murakkablik qonuni chiqarilmadi.

8-jadval.

6 000 hujjatda postings ma’lumotlarini kodlash hajmi

Postings turi	32 bitli shakl, bayt	Farq + VB, bayt	Kamayish
Boolean: docID	1 331 020	337 217	74,66 %
Pozitsion: docID, tf, pozitsiyalar	6 071 148	1 544 900	74,55 %

Izoh: bu to‘liq indeks fayli yoki RAM o‘lchovi emas. Lug‘at, siljish ko‘rsatkichlari, sarlavhalar va obyekt xarajatlari hisobga olinmagan; 1 KiB = 1 024 bayt.

(cc) BY

Boolean postings yuklamasi 1 299,82 KiB dan 329,31 KiB gacha qisqardi. Siqilgan pozitsion yuklama esa 1 508,69 KiB bo‘lib, siqilgan Boolean yuklamadan 4,58 marta katta chiqdi. Taqqoslashda ikkala shaklda ham bir xil lug‘at chetlab o‘tilgan. Shunga qaramay, bu nisbat boshqa hujjat uzunligi yoki atama chastotasiga ega korpusga ko‘chirilmasligi kerak.

Siqish tamoyilini [3, 10, 11, 1000] docID ro‘yxati bilan ko‘rsatish mumkin. Farqlar [3, 7, 1, 989] bo‘ladi. 32 bitli shakl 16 bayt talab etsa, ushbu VB kodlashida farqlar 1 + 1 + 1 + 2 = 5 baytni egallaydi. Qayta yig‘ishda 3, 3+7, 10+1 va 11+989 qiymatlari dastlabki ro‘yxatni tiklaydi. Tajribadagi barcha ro‘yxatlar uchun yo‘qotishsiz tiklash nazorati bajarildi.

Bazaviy tuzilmani qurish xarajatini hisobga olgan taxminiy qoplanish chegarasi Q* = (60,031 - 23,613)/(0,3893 - 0,0167) bo‘yicha 98 so‘rov atrofida chiqadi. Bu faqat shu korpusdagi median xarajatlar bilan qurilgan soddalashtirilgan ssenariydir; yangilash, disk, parallel xizmat va xotira narxi kiritilmagan. Siqilgan postings qidiruv vaqtida ochib ishlatilmadi, shu sababli hajm tejashni jadvaldagi tezlashishning sababi deb bo‘lmaydi.

Kichik korpusda indeks va BM25 hisoblash misoli

Indeks ichidagi ma’lumotni bevosita tekshirish uchun oltita qisqa hujjat olindi. Bu korpus yuqoridagi 6 000 hujjatli tezlik tajribasidan mustaqildir. 9-jadvaldagi matnlarda tinish belgilari olib tashlanib, apostrof saqlangan; stop-so‘zlar tushirilmagan.

9-jadval.

Hisoblash misoli uchun qisqa hujjatlar

ID	Hujjat matni	Tokenlar
D1	axborot izlash tizimi indeks asosida ishlaydi	6
D2	indeks axborot izlash tezligini oshiradi	5
D3	axborot tizimi ma’lumot saqlaydi	4
D4	izlash algoritmi hujjatlarni topadi	4
D5	axborot va izlash alohida mavzu	5
D6	indeks indeks axborot izlash	4

Teskari ro‘yxatlar quyidagicha hosil bo‘ldi: axborot -> {D1, D2, D3, D5, D6}; izlash -> {D1, D2, D4, D5, D6}; indeks -> {D1, D2, D6}. “Axborot AND izlash” natijasi {D1, D2, D5, D6}; aniq “axborot izlash” iborasi esa {D1, D2, D6} bo‘ldi. D5 ikkala atamani saqlaydi, lekin ular orasida “va” tokeni bor.

BM25 uchun so‘rov {axborot, indeks}, N = 6, avgdl = 28/6 = 4,6667 deb olindi. df(axborot) = 5, df(indeks) = 3; (4)-formula bo‘yicha IDF qiymatlari mos ravishda 0,2412 va 0,6931 chiqdi. Hisoblashda barcha olti hujjat ko‘rsatildi, ammo amaliy nomzodlarni OR bo‘yicha ajratganda D4 kirmaydi.

10-jadval.

BM25 ballarini hisoblash natijasi

ID	Uzunlik	tf(axborot)	tf(indeks)	BM25 ball	O‘rin
D1	6	1	1	0,8365	3

(cc) BY

ID	Uzunlik	tf(axborot)	tf(indeks)	BM25 ball	O‘rin
D2	5	1	1	0,9078	2
D3	4	1	0	0,2561	4
D4	4	0	0	0,0000	6
D5	5	1	0	0,2343	5
D6	4	1	2	1,2491	1

D6 hujjati qisqa va “indeks” atamasini ikki marta saqlagani uchun eng katta ball oldi. D1 va D2 da so‘rov atamalari chastotasi bir xil, ammo D2 qisqaroq bo‘lgani uchun uning balli yuqoriroq chiqdi. Demak, hujjat identifikatorlari ro‘yxati nomzodlarni ajratadi; chastota va uzunlik statistikasi ularni tartiblaydi. Bu misol BM25 formulasini tekshiradi, lekin D6 inson uchun eng foydali hujjat ekanini isbotlamaydi: bunday xulosa uchun mustaqil relevantlik baholari zarur.

MUHOKAMA

Natijalarning mazmuni va amaliy qarorlar

Natijalar indeksning ikki mustaqil afzalligini ko‘rsatdi: teskari tuzilma tekshiriladigan hujjatlar doirasini qisqartirdi, pozitsiyalar esa ibora shartini asl hujjatning barcha tokenlarini ko‘rmasdan tekshirishga imkon berdi. Bunda pozitsion ma’lumot Boolean natijasini “yaxshilamadi”, balki murakkabroq so‘rovni bajarish uchun qo‘shimcha axborot berdi. Apostrof misolida esa natija indeks algoritmi almashtirilgani uchun emas, atama kalitlari izchil birxillashtirilgani uchun o‘zgardi.

Shu asosda universitet elektron kutubxonasi uchun quyidagi boshlang‘ich arxitektura taklif etiladi: sarlavha va asosiy matnni alohida maydonlarda indekslash; hujjat identifikatori, chastota va kerakli pozitsiyalarni saqlash; muallif, yil hamda hujjat turini alohida filtrlar orqali qayta ishlash. Kalit atamalar bo‘yicha natijalarni ajratishdan so‘ng BM25 bilan tartiblash, aniq ibora so‘ralganda esa pozitsion tekshiruvni qo‘llash mumkin. Bu amaliy loyiha tavsiyasi bo‘lib, maqolada tayyor platforma sinovdan o‘tkazildi, degan ma’noni anglatmaydi.

Indeksni yangilash ham alohida qarordir. Asosiy indeks bilan yangi hujjatlar uchun yordamchi indeksni birga yuritish va keyin ularni birlashtirish dinamik indekslash yechimlaridan biridir [1]. Ushbu ishda faqat statik holat o‘lchandi. Shuning uchun yangilash tezligi, o‘chirilgan hujjatlarni filtrlash va qidiruvga ko‘rinish kechikishi uchun alohida sinov rejalashtirilishi kerak. Indeksni qayta qurishda eski va yangi avlodlarni ajratish nazoratli taqqoslashni osonlashtiradi.

Ishonchlilik chegaralari va keyingi tekshiruvlar

Birinchi cheklov korpusning sun’iyligidir. Mavzular teng taqsimlangan, lug‘at hajmi 976 atama bilan cheklangan va ibora mavjudligi maxsus qurilgan. Tabiiy matndagi ko‘p ma’nolilik, imlo xatolari, grammatik shakllar hamda haqiqiy so‘rovlar taqsimoti bu yerda to‘liq ifodalanmagan. 33,33 foizlik ibora precision ko‘rsatkichi va normalizatsiyadagi 100 foizlik recall aynan nazorat konstruksiyasiga tegishli; ularni real kutubxona sifati sifatida talqin qilish mumkin emas.

Ikkinchi cheklov amalga oshirish muhitidir. Skan ham, teskari kesishuv ham Python set obyektlaridan foydalanadi, lekin ularning ichki bajarilish xarajatlari bir xil emas. Bitta umumiy hisoblash muhitidagi yetti takror mutlaq tezlikni universal qilmaydi. Parallel foydalanuvchilar, tarmoq, disk kiritish-chiqarishi va sovuq kesh o‘lchanmagan. Indeks o‘lchovi ham butun dastur RAM sarfidan farq qiladi; siqilgan oqimni ochish vaqti alohida tekshirilishi lozim.

Uchinchi cheklov baholash dizaynidir. Hajmlar bitta korpusning ichma-ich qismlari

(cc) BY

bo‘lgani uchun mustaqil korpuslar emas. Statistik ahamiyat testi yoki ishonch oralig‘i hisoblanmadi. Keyingi bosqichda kamida bir nechta boshlang‘ich qiymat, turli hujjat uzunligi va real annotatsiyalangan so‘rovlar bilan tekshirish taklif etiladi. P@10, Recall@100 va nDCG@10 kabi mezonlarni faqat tegishli relevantlik etaloni tayyorlangach qo‘llash kerak [1, 5].

XULOSA

Nazoratli tajriba indeksni tanlashda qidiruv sharti, vaqt va saqlash xarajatini birgalikda ko‘rish zarurligini ko‘rsatdi. 6 000 hujjatda teskari indeks Boolean so‘rov medianasini 0,3893 ms dan 0,0167 ms gacha qisqartirdi. Pozitsion indeks aniq iboralarni to‘liq skan bilan bir xil topib, 26,21 martalik median tezlashishni berdi. Biroq uning qurilish xarajati va postings hajmi kattaroq bo‘ldi. DocID farqlarini VB usulida kodlash Boolean postings yuklamasini 74,66 foizga kamaytirdi.

Apostrof nazorati matnni birxillashtirishning indeks tuzilmasidan mustaqil ahamiyatini, BM25 misoli esa chastota va uzunlik statistikasi orqali tartiblashni ko‘rsatdi. Amaliy tizim uchun teskari indeksni asos, pozitsiyalarni ibora talabi mavjud maydonlarda qo‘shimcha imkoniyat sifatida ko‘rish maqsadga muvofiq. Natijalar ishlab chiqarish tizimining tayyor kafolati emas; yakuniy tanlov real korpus, foydalanuvchi baholari va ekspluatatsiya cheklovlari bilan qayta tekshirilishi lozim.

FOYDALANILGAN ADABIYOTLAR

1. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008. 1-6 va 8-boblar. ISBN: 978-0-521-86571-5. Elektron nashr: nlp.stanford.edu/IR-book/.
2. Zobel J., Moffat A. Inverted files for text search engines. ACM Computing Surveys. 2006. Vol. 38, No. 2. Article 6. DOI: 10.1145/1132956.1132959.
3. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval. 2009. Vol. 3, No. 4. P. 333-389. DOI: 10.1561/1500000019.
4. Malkov Yu. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. arxiv:1603.09320v4, 2018. DOI: 10.48550/arxiv.1603.09320.
5. Thakur N., Reimers N., Rücklé A., Srivastava A., Gurevych I. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. NeurIPS, Datasets and Benchmarks Track, 2021. arXiv:2104.08663.
6. Apache Software Foundation. BM25Similarity. Apache Lucene 9.12.1, Core API: org.apache.lucene.search.similarities.BM25Similarity. Rasmiy hujjat.
7. Unicode Consortium. Unicode Standard Annex #15: Unicode Normalization Forms. Unicode 17.0.0, revision 57, 30.07.2025. unicode.org/reports/tr15/tr15-57.html.
8. Python Software Foundation. time - Time access and conversions: perf_counter_ns. Python 3.13 rasmiy hujjatlari. docs.python.org/3.13/library/time.html.

(cc) BY