ТАТУ ХАБАРЛАРИ
ВЕСТНИК ТУИТ
TUIT BULLETIN

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETINING
ILMIY-TEXNIKA VA AXBOROT-TAHLILIY JURNALI

JURNAL 2007 YILDA TASHKIL ETILGAN.
BIR YILDA TO‘RT MARTA NASHR QILINADI

1 (73)/2025

https://uzjurnal.uz

«TATU xabarlari» jurnali («Вестник ТУИТ», “TUIT Bulletin”) O‘zbekiston matbuot va axborot agentligida 2007 yil 22 yanvarda 0204 - son bilan ro‘yxatdan o‘tgan.
O‘zR OAK tomonidan doktorlik dissertatsiyalari yuzasidan ilmiy maqolalar chop etilishi lozim bo‘lgan ilmiy jurnallar ro‘yxatiga kiritilgan
(2008 yil 2 yanvardagi 001-I-sonli buyruq).
Tahririyat manzili:
100202, Toshkent sh., Amir Temur ko‘chasi, Ф408-xona. Tel.: (+99871) 207-59-41
E-mail: tuit_xabar@tuit.uz
Qo‘lyozmalar taqrizlanmaydi va qaytarilmaydi.

TOSHKENT - 2025

UO‘K 81.322

O‘ZBEK TILI MILLIY KORPUSINING ARXITEKTURASI, MODELI VA ALGORITMLARI

Elov B.B.

Ushbu maqolada O‘zbek tili milliy korpusining konseptual arxitekturasi, ma’lumotlar modeli va qayta ishlash konveyeri (normallashtirish, tokenlash, POS-teglash, morfolofik, sintaktik va semantik tahlil va koreferensiyani aniqlash) bo‘yicha yechim taklif etiladi. Tizim TEI kodlash va Universal Dependencies (UD) tamoyillariga tayangan holda matn va meta ma’lumotlarni izchil saqlash, shuningdek, KWIC konkordans, kollokatsiyalar hamda n-gram chastota tahlilini ta’minlovchi indeksatsiya va API qatlamlaridan iborat. Arxitektura modullari (tozalash/normalizatsiya, lingvistik teglash, indekslash/qidiruv va analitika) mikroxizmatlar sifatida loyihalangan bo‘lib, moslashuvchanlik va kengayuvchanlikni qo’llab-quvvatlaydigan qilib ishlab chiqilgan. Maqolada milliy korpuslar, TEI/UD standartlari, Sketch Engine/CWB kabi korpus vositalari va turkiy tillar (xususan, o‘zbek) bo‘yicha dolzarb ishlanmalar tahlil qilinadi. Taklif etilayotgan yechim O‘zbek tili milliy korpusi uchun izchil ma’lumot modeli va qayta ishlash algoritmlarini belgilab, ochiq standartlarga tayanuvchi ilmiy-tadqiqot infratuzilmasini shakllantirishni maqsad qiladi.

Kalit so‘zlar: O‘zbek tili milliy korpusi, TEI, Universal Dependencies, KWIC, kollokatsiya, n-gram, indekslash, korpus arxitekturasi, koreferensiya, korpusda qidiruv, korpusda analitika, mikroxizmatlar.

В данной статье предлагается решение, охватывающее концептуальную архитектуру, модель данных и конвейер обработки (нормализация, токенизация, разметка частей речи, морфологический, синтаксический и семантический анализ, а также разрешение кореференции) Национального корпуса узбекского языка. Система опирается на принципы TEI-кодирования и Universal Dependencies (UD) для согласованного хранения текстов и метаданных, а также включает слои индексации и API, обеспечивающие анализ KWIC-конкордансов, коллокаций и частотности n-грамм. Архитектура модулей (очистка/нормализация, лингвистическая разметка, индексирование/поиск и аналитика) разработана как микросервисы, поддерживающие гибкость и расширяемость. В статье анализируются современные разработки в области национальных корпусов, стандартов TEI/UD, корпусных инструментов, таких как Sketch Engine/CWB, и исследований по тюркским языкам (в частности, узбекскому). Предлагаемое решение направлено на формирование последовательной модели данных и алгоритмов обработки для Национального корпуса узбекского языка, а также создание исследовательской инфраструктуры на основе открытых стандартов.

***Ключевые слова*:** Национальный корпус узбекского языка, TEI, Universal Dependencies, KWIC, коллокации, n-граммы, индексирование, архитектура корпуса, кореференция, поиск по корпусу, аналитика в корпусе, микросервисы.

This paper proposes a solution covering the conceptual architecture, data model, and processing pipeline (normalization, tokenization, POS tagging, morphological, syntactic and semantic analysis, and coreference resolution) of the National Corpus of the Uzbek Language. The national corpus is considered a fundamental resource for linguistics, computational linguistics, and Natural Language Processing (NLP), offering functionalities for searching language units as well as performing morphological, syntactic, and semantic analyses. The system is based on TEI encoding and Universal Dependencies (UD) principles to ensure consistent storage of texts and metadata, and it includes indexing and API layers that support KWIC concordance, collocation, and n-gram frequency analysis. The proposed architecture adopts a modular structure, providing a user-friendly web interface and RESTful API access for developers. SQL Server is employed as the database solution to ensure efficient storage, indexing, and retrieval of large-scale texts. Annotation processes include morphological analysis, POS tagging, syntactic dependency parsing, Named Entity Recognition (NER), and Semantic Role Labeling (SRL). Furthermore, the study compares the Uzbek National Corpus with international projects such as BNC, RNC, and TNC, highlighting best practices and limitations for adaptation. The architecture modules (cleaning/normalization, linguistic annotation, indexing/search, and analytics) are designed as microservices to support flexibility and scalability. The paper reviews recent developments in national corpora, TEI/UD standards, corpus tools such as Sketch Engine/CWB, and research on Turkic languages (particularly Uzbek). The proposed solution aims to define a consistent data model and processing algorithms for the Uzbek National Corpus and to establish a research infrastructure based on open standards. The system's performance has been evaluated using F1-score, UAS, and LAS metrics. In conclusion, the Uzbek National Corpus is emphasized as a valuable digital resource not only for linguists but also for translators, educators, software developers, and the broader community.

***Keywords*:** Uzbek National Corpus, TEI, Universal Dependencies, KWIC, collocation, n-gram, indexing, corpus architecture, coreference, corpus search, corpus analytics, microservices.

I. KIRISH

O‘zbek tilining milliy korpusi – bu o‘zbek tilidagi matnlarning jamlangan elektron axborot-ma’lumotlar bazasi bo‘lib, tilni kompyuter yordamida chuqurroq o‘rganish va axborot texnologiyalari bilan integratsiyalashish uchun xizmat qiladi.

Milliy korpus mazmunan turli sohalar va janrlardagi matnlarni qamrab oladi va ularni lingvistik teglash amalga oshiriladi. Korpus dasturi foydalanuvchilarga *matnlar ichida izlash, til hodisalarini tahlil qilish, statistik ma’lumotlar olish* kabi imkoniyatlarni taqdim etadi. Bunday tizimni yaratishda zamonaviy arxitektura yondashuvlari hisoblangan, *modulli tuzilma* yoki *xizmatga yo‘naltirilgan arxitektura* – SOA qo‘llanilib, veb-interfeys hamda dasturiy interfeyslar (API) orqali foydalanish ta’minlanadi.

O‘zbek tili milliy korpusi uchun taklif etilayotgan arxitektura modelining asosiy komponentlari, ularning o‘zaro aloqasi va ishlash algoritmlari, shuningdek xorijiy milliy korpuslar tajribasi bilan qiyosiy tahlil keltiriladi.

Milliy korpuslar arxitekturasi va belgilash standartlari. TEI P5 – bu matnlarning raqamli shaklda kodlanishi va tahlil qilinishi uchun mo‘ljallangan keng qamrovli qo‘llanma va standartlar to‘plami bo’lib, korpus dizayni, meta ma’lumotlar va ko‘p darajali belgilash uchun asosiy tamoyillarni belgilaydi [12, 13]. UD v2 yo‘riqnomalari morfologik va sintaktik teglash hamda bog‘lanishlarni tillararo izchillikda berishni ta’minlaydi [14, 15]. O‘zbek tili milliy korpusi uchun UD asosidagi teglar to‘plami va konvensiyalar o‘zbek tilining agglutinativ xususiyatlarini inobatga olishi zarur.

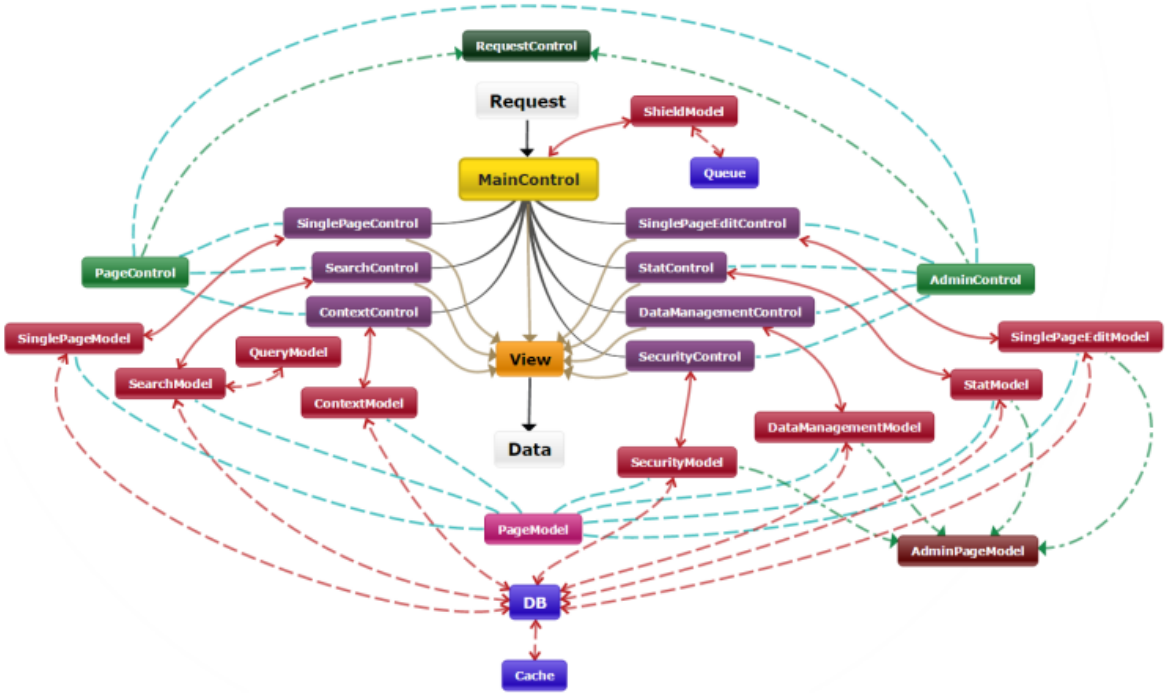
O‘zbek va turkiy tillar bo‘yicha resurslar. Yaqinda taqdim etilgan Uzbek-UT UD treebank (500 gap, 5850 token) o‘zbek tili uchun UD asosidagi birinchi ochiq resurslardan biri bo‘lib, lemma, POS, morfologik xususiyatlar va bog‘lanishlar bilan teglangan [21, 22]. UDPipe 2 va Stanza kabi vositalar UD treebanklarga tayangan holda tokenlash, lemmalash va sintaktik tahlilni amalga oshiradi [9-11]. Turkiy tillar doirasida Zemberek va boshqa ochiq manbali ishlanmalar morfologik tahlil va normallashtirishda amaliy yordam beradi [1, 23]. Shuningdek, o‘zbek-qozoq parallel korpusi kabi mintaqaviy resurslar ko‘p tilli modellashtirish uchun qulay hisoblanadi [24].

Korpus qidiruvi, KWIC va kollokatsiyalar. Sketch Engine va IMS Open Corpus Workbench (CWB/CQP) kabi tizimlar konkordans, kollokatsiya hamda atama izlash bo‘yicha asosiy standart hisoblanadi [5, 7]. Kollokatsiyalar statistikasi bo‘yicha klassik ishlarda PMI, t-test, log-likelihood va boshqa ko‘rsatkichlar qiyosiy tahlil qilinadi [3, 4]. N-gram va chastota taqsimotlarida Zipf qonuni kuzatiladi [20].

Indekslash va infratuzilma. Lucene oilasidagi invers indekslar tezkor qidiruv va “facet”lar uchun standartdir [18-19]. “Facet”lar – bu qidiruv natijalarini meta ma’lumot bo‘yicha guruhlab ko‘rsatish va tez filtrlash imkonini beradigan atributlar. CWB/CQP esa morfo sintaktik atributlar bo‘yicha murakkab so‘rovlarni qo‘llaydi [5]. O‘zbek tili milliy korpusi arxitekturasi uchun matnlarni yig‘ish→normallashtirish→teglash→indekslash→qidiruv→analitika jarayonlari mikroxizmatlarga bo‘linib, navbatlar va batch/stream shaklda amalga oshiriladi.

II. ASOSIY QISM

Arxitektura modeli va asosiy komponentlar. O‘zbek tilining milliy korpusi arxitekturasi modul tuzilmali bo‘lib, asosiy komponentlar mustaqil modullar yoki mikroxizmatlar tarzida tashkil etiladi. Dasturiy tizim MVC (Model-View-Controller) texnologiyasi asosida ishlab chiqilgan bo‘lib, bu yondashuv tizimni funksional qatlamlarga bo‘lish orqali kengaytiriluvchanlik va parvarishlash qulayligini ta’minlaydi.



1-rasm. O‘zbek tili milliy korpusi dasturiy ta’minotining MVC (Model-View-Controller) arxitekturasi

Asosiy komponentlar quyidagilardan iborat:

1. **Veb-server va foydalanuvchi interfeysi** – Korpus bilan ishlash uchun veb-ilova (Django framework) yaratilgan. Django Python tili asosida MVC arxitekturasini ta’minlaydi va veb-interfeys **“View” qatlami** sifatida foydalanuvchiga qulay tarzda ma’lumotlarni chiqaradi. Veb-interfeys orqali foydalanuvchi so‘rovlarni (kalit so‘z, so‘z shakli, lemma va h.k.) kiritadi va natijalarni ko‘radi. Django dasturida **Model qatlami** ma’lumotlar bazasi bilan ishlaydi, **Controller qatlami** so‘rovlarni qabul qilib tegishli modelga uzatadi.

2. **API xizmati** – Korpusdan tashqi dasturlar ham foydalana olishi uchun REST usulidagi API interfeyslar ta’minlanadi. Bu xizmat orqali masofaviy ilovalar yoki tadqiqotchilar maxsus so‘rovlar (masalan, JSON formatida) yuborib, korpus ma’lumotlarini olishi mumkin. Veb-ilova va API bir xil serverda integratsiyalashgan holda (monolit Django dasturi doirasida) ishlab, har ikkala interfeys bitta kod

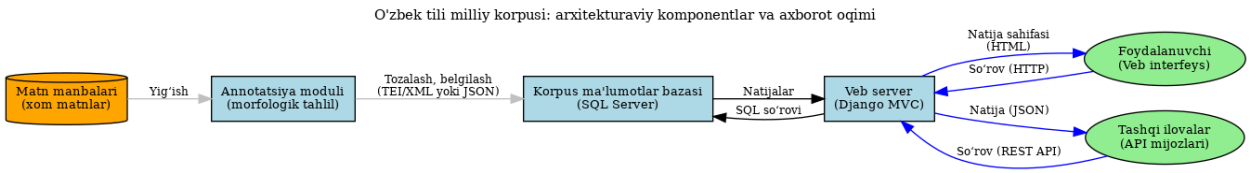
bazasiga asoslangan. Shu tariqa, foydalanuvchi veb-sahifa orqali ham, dasturchi to‘g‘ridan-to‘g‘ri API orqali ham korpus bilan ishlash imkoniyatiga ega bo‘ladi.

3. **Ma’lumotlar bazasi** – Korpusning yuragi sifatida katta hajmdagi matnlar va ularning lingvistik teglari saqlanadigan SQL Server ma’lumotlar bazasi ishlatildi. SQL Server tanlanishi korpusdagi so‘rovlarni optimallashtirish, murakkab filtrlar bilan tezkor qidiruvni ta’minlash va korxona darajasida ishonchlilik sabablaridadir. Ma’lumotlar bazasi tarkibida korpusdagi matnlar, ularning metadata (muallif, sana, janr, manba kabi ma’lumotlar), shu matnlardagi har bir so‘zning ajratilgan shakllari (tokenlar), lemmalari va morfologik belgilari kabi tuzilmalar mavjud bo‘ladi. Ma’lumotlar bazasida korpus uchun to‘liq matnli qidiruv indekslari shakllantirilgan. Bu esa yirik hajmdagi matnlarda so‘rovlar tez ishlashi uchun muhim.

4. **Teglash moduli** – Korpusga kelib tushayotgan strukturlanmagan matnlarni lingvistik jihatdan teglovchi dasturiy modullar. Bu modul odatda morfologik tahlilator va teglagichlardan iborat bo‘ladi. Morfologik tahlilator so‘zlarning lug‘aviy shaklini (lemmasini) va grammatik kategoriyalarini (masalan, so‘z turkumi, vaqt, son, kelishik kabilar) aniqlaydi. Teglash moduli o‘zbek tilining grammatik modeliga asoslanadi. Masalan, yaratilgan qoidalarga asoslangan yoki ML modeli asosida o‘qitilgan maxsus dastur (POS-tagger). Korpus tizimida bu modul yangi matnlarni bazaga kiritishdan oldin avtomatik ishlaydi.

5. **Boshqaruv vositalari** – Korpusni yangilab borish, yangi matnlar qo‘shish, sifati nazoratini qilish uchun alohida **admin moduli** mavjud. Bu modul orqali korpusga mas’ul shaxslar matnlarni yuklaydi, ularni tozalash va teglash jarayonlarini boshqaradi, foydalanuvchilar faoliyatini kuzatadi. Django framework’ining admin paneli bu jarayonlarni yengillashtiradi.

Yuqoridagi komponentlarning o‘zaro bog‘liqligi quyi 2-rasmda aks ettirilgan. Bu arxitekturada tizim modullarning mustaqilligini saqlagan holda ularni yagona tizimga birlashtirilgan.



2-rasm. O‘zbek tili milliy korpusining arxitekturaviy komponentlar va axborot oqimi modeli.

O‘zbek tili milliy korpusi arxitekturasi ikki asosiy jarayonni qamrab oladi:

1) Korpusni shakllantirish (ma’lumotlar oqimi).

2) Korpusdan foydalanish (so‘rovlar oqimi).

Bunday modulli arxitektura yangi komponentlarni qo‘shishda yoki mavjudlarini almashtirishda qulaylik yaratadi. Masalan, kelgusida morfologik

tahlilator o‘rniga yanada samaraliroq model paydo bo‘lsa, tizimga integratsiya qilish oson.

Tizim modullari va xizmatlararo integratsiya. Modullar o‘rtasida ma’lumot almashinuvi aniq belgilangan interfeyslar orqali amalga oshiriladi. **Veb-server qatlami (Django)** foydalanuvchi yuborgan so‘rovlarni qabul qilib oladi va ularni dasturiy API chaqiruvlariga aylantiradi. Masalan, foydalanuvchi qidiruv maydoniga biror so‘z kiritib *“Izlash”* tugmasini bossa, Django serveri ichida tegishli qidiruv funksiyasi chaqiriladi. Bu funksiya SQL so‘rovini generatsiya qiladi yoki ichki qidiruv xizmatiga murojaat qiladi. So‘rov tarkibida foydalanuvchi tanlagan filtrlar va parametrlar bo‘lsa (*masalan, faqat badiiy matnlardan izlash, muayyan muallif bo‘lgan matnlardan qidirish kabi*), ular ham hisobga olinadi va funksiyaga uzatiladi.

Django ilovasi doirasida ichki xizmatlararo aloqa funksiya chaqiruvlari va kutubxonalar orqali amalga oshiriladi. Masalan, morfologik tahlil xizmati integratsiya qilingan bo‘lib, Django model qismida unga murojaat qiluvchi funksiya bo‘ladi. Xizmatga so‘z yuborilganda, u javoban o‘sha so‘zning tahlil natijasini (*lemmasi, grammatik teglari*) qaytaradi va bu natija bazaga yoziladi. Shu tariqa, modul ichidagi APIlar (REST API chaqiriqlari) orqali o‘zaro aloqa amalga oshiriladi.

Xizmatlararo ma’lumot almashinuv formatlari sifatida **JSON** va **XML (TEI)** qo‘llanildi. Ma’lumotlarni teglash moduli alohida mikroxizmat ko‘rinishida bo‘lib, Django server HTTP so‘rov orqali unga matnni yuboradi va *JSON formatida tahlil natijalarini* oladi. Bunday yondashuv tilga oid qismlarni (*morfologik tahlil, sintaktik tahlil kabi*) asosiy serverdan mustaqil rivojlantirish va kerak bo‘lganda mustaqil ishga tushirish imkonini beradi. Bu xizmatga **yo‘naltirilgan arxitektura (SOA)**ning afzalliklaridan biridir.

Foydalanuvchi interfeysi va API o‘rtasidagi aloqa ham yuqoridagi 2-rasmda ko‘rsatilganidek, umumiy server orqali bir xil bazaga murojaat qiladi. Foydalanuvchi brauzeri orqali kelgan so‘rovlar uchun Django HTML sahifani natija bilan shakllantirib qaytaradi, API mijozlari uchun esa JSON ko‘rinishdagi natijalar qaytariladi. Har ikki holda ham so‘rovlarning protokoli HTTP/HTTPS bo‘lib, xavfsizlik va autentifikatsiya masalalari umumiy hal qilingan (masalan, API’dan foydalanish uchun token yoki kalitlar orqali identifikatsiya). Shu bilan birga, korpusga kirish huquqlari boshqaruvi ham joriy etiladi. Ayrim ma’lumotlar faqat tahlilchilar uchun yoki ma’muriyat uchungina ko‘rinadigan qilinishi mumkin. Ma’lumotlar almashinuvi jarayoni bosqichma-bosqich quyidagicha kechadi:

1) **Matnlarni yuklash:** Administrator yangi matnni korpusga qo‘shish uchun veb-interfeys yoki maxsus admin panel orqali faylni yuklaydi. Dastur bu matnni qabul qilib, dastlabki tozalash (boshlangi`ch qayta ishlash)dan o‘tkazadi.

2) **Teglash:** Yuklangan matn **teglash moduli**ga uzatiladi. Bu modul matnni tokenlarga ajratadi (gaplarga va so‘zlarga bo‘ladi), har bir so‘zga tegishli morfologik, sintaktik va semantik teglarni biriktiradi. Natija sifatida masalan, TEI XML (CONLL-2012) formatidagi teglangan matn hosil bo‘ladi. Bunda <*w*> tegi

ostida so‘z va uning atributlari (lemmasi, grammatikasi) ko‘rsatiladi. Modul natijani bazaga yozish uchun to‘g‘ridan-to‘g‘ri JSON obyektlari ham qaytarishi mumkin.

3) **Ma’lumotlar bazasiga saqlash:** Teglangan matnlar SQL so‘rovlar orqali tegishli jadvallarga saqlanadi. Bu jarayonda bir nechta jadvalga ma’lumot qo‘shiladi: *Matnlar jadvali* (yangi matn haqida yozuv), *Gaplar jadvali* (matndagi har bir gap alohida, qator raqami bilan), *So‘zlar jadvali* (har bir so‘zning o‘zi va uning begin-end pozitsiyasi, bog‘lanishlari), *Teglar jadvali* (teglarni saqlash uchun alohida normativ jadvallarga). Bazaga saqlash vaqtida tranzaksiya yaxlitligi va ma’lumotlar izchilligi ta’minlanadi.

4) **Foydalanuvchi so‘rovi:** Foydalanuvchi qidiruv formasiga masalan, muayyan (masalan, *“kitob”)* so‘zini kiritib izlashni boshladi. Django serveri bu so‘rovni olgach, qidiruv modeliga murojaat qiladi. Qidiruv modeli SQL ma’lumotlar bazasida tegishli jadvallardan kerakli ma’lumotlarni so‘raydi. Bizning misolda, so‘z bo‘yicha qidirilsa, avvalo **Word** jadvalidan *“kitob”* so‘ziga mos ID olinadi, so‘ng **Index** jadvalidan ushbu ID qatnashgan barcha qatordagi ma’lumotlar olinadi. Bunda har bir qatorda o‘sha so‘z uchragan biror joy (*matnID, gapID, so‘z pozitsiyasi va h.k.*) bo‘yicha ma’lumot mavjud. Agar foydalanuvchi qo‘shimcha filtrlar tanlagan bo‘lsa (masalan, faqat gazeta janridagi matnlardan qidirish), SQL so‘rov shu shart bilan optimallashtiriladi (ya’ni WHERE matn.janr = 'gazeta' kabi). Korpus arxitekturasi shu tarzda bir nechta kriteriyali qidiruvni qo‘llab-quvvatlaydi.

5) **Natijalarni qayta ishlash:** Bazadan olingan natijalar dastur tomonidan kerakli tartibda formatlanadi. Masalan, foydalanuvchiga konkordans shaklida ko‘rsatilishi uchun har bir topilgan misol kontekst bilan birga tuziladi: *<chap konteks> <kalit so‘z> <o‘ng konteks>*. Django bularni HTML shaklga keltirib, foydalanuvchiga yuboradi. Agar foydalanuvchi natijalarni batafsil ko‘rmoqchi bo‘lsa (masalan, butun gapni yoki paragrafni), u holda qo‘shimcha AJAX so‘rov orqali yoki alohida sahifa orqali batafsil ko‘rinish taqdim etiladi. API orqali so‘ralganda esa, natijalar strukturaviy ma’lumot sifatida (JSON massivlari ko‘rinishida) qaytariladi.

Shunday qilib, modullararo o‘zaro aloqa aniq protokollar va formatlar asosida quriladi. Bu tamoyil korpus tizimining shaffofligini va barqaror ishlashini ta’minlaydi. Tatar tili milliy korpusi uchun yaratilgan tizim misolida ham modul va xizmatlarga bo‘lingan arxitektura samarali ekani ko‘rsatilgan [25]: ular MVC konsepsiyasiga asoslangan korpus-menejer tizimini ishlab chiqib, har bir vazifani alohida controller va modelga ajratganlar, natijada tizimga yangi funksiyalar qo‘shish tez va oson kechadi. O‘zbek milliy korpusi arxitekturasi ham xuddi shunday, mustaqil xizmatlar majmuasi sifatida ishlab, ularning o‘zaro bog‘liqligi aniq interfeyslar bilan belgilanadi.

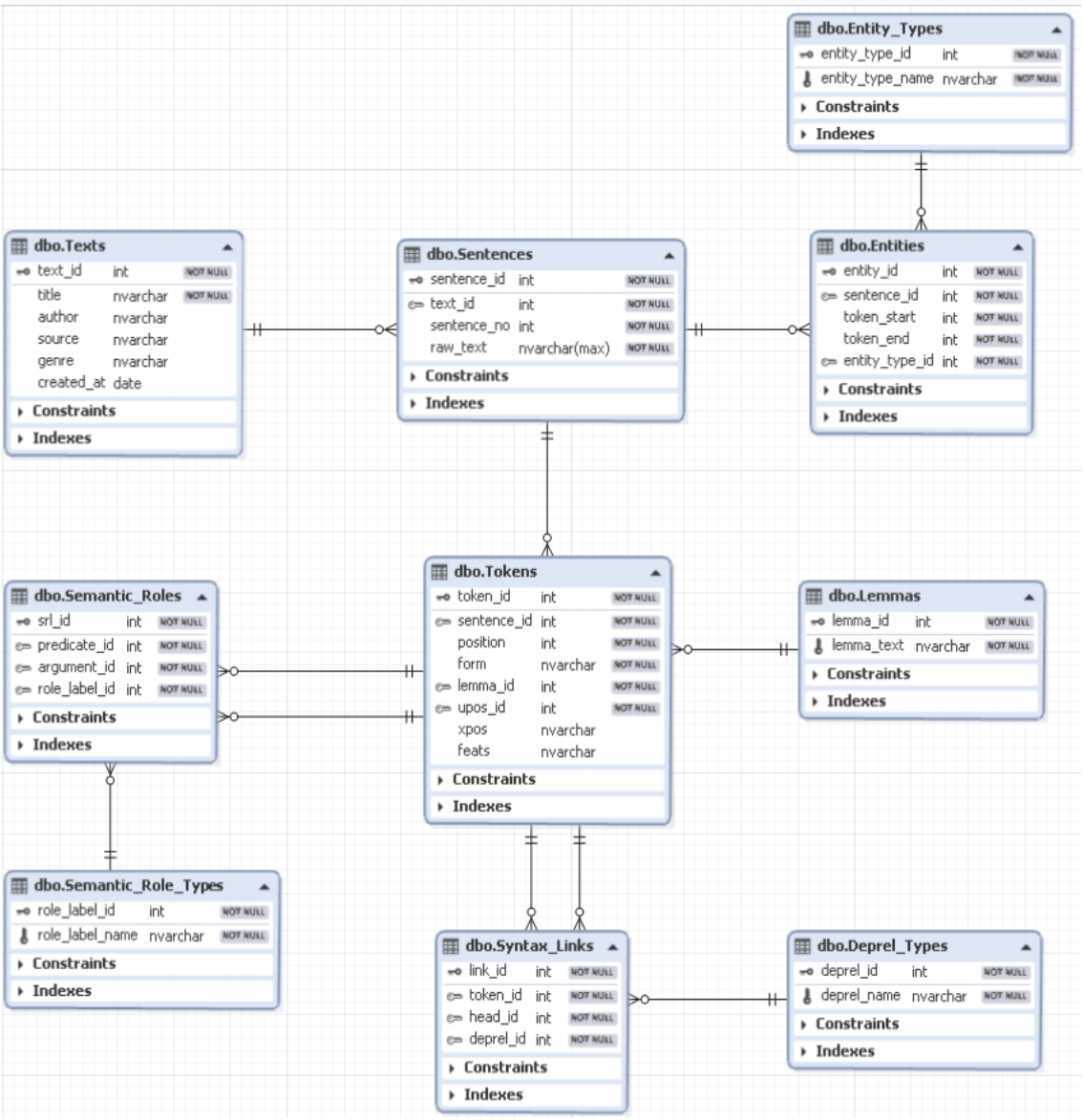
Ma’lumotlar bazasi tuzilishi va qidiruv algoritmlari. Korpusning ma’lumotlar bazasi relatsion modelga ega bo‘lib, bir-biri bilan bog‘langan bir necha

asosiy jadvallarni o‘z ichiga oladi. Quyidagi 1-jadvalda shunday tuzilmaning soddalashtirilgan ko‘rinishi keltirilgan (ayrim ustunlar tushunarli bo‘lishi uchun qisqartirib berilgan).

1-jadval

Korpus ma’lumotlar bazasining asosiy jadvallari va ularning maydonlari

№	Jadval nomi	Asosiy maydonlar (ustunlar)	Tavsif
1	**Texts**	text_id (PK), title, author, source, genre, created_at	Matnlar va ular haqida metama’lumotlar saqlanadi
2	**Sentences**	sentence_id (PK), text_id (FK), sentence_no, raw_text	Har bir matndagi gaplar to‘liq matn holida saqlanadi
3	**Tokens**	token_id (PK), sentence_id (FK), position, form, lemma_id (FK), upos_id (FK), xpos, feats	Har bir gapdagi token (so‘z shakli) haqida ma'lumotlar
4	**Lemmas**	lemma_id (PK), lemma_text	Har bir lemma (so‘zning asos shakli) alohida saqlanadi
5	**UPOS_Tags**	upos_id (PK), upos_name	Universal POS teglar (so‘z turkumlari) ro‘yxati saqlanadi
6	**Syntax_Links**	link_id (PK), token_id (FK), head_id (FK), deprel_id (FK)	Sintaktik bog‘lanishlarni saqlaydi (Dependency Parsing uchun)
7	**Deprel_Types**	deprel_id (PK), deprel_name	Sintaktik bog‘lanishlar turlari ro‘yxati
8	**Entities**	entity_id (PK), sentence_id (FK), token_start, token_end, entity_type_id (FK)	NER (nomlangan obyektlar) annotatsiyalari
9	**Entity_Types**	entity_type_id (PK), entity_type_name	NER ob'ektlarining turlari ro‘yxati (PER, LOC, ORG va h.k.)
10	**Semantic_Roles**	srl_id (PK), predicate_id (FK), argument_id (FK), role_label_id (FK)	Semantik rollar annotatsiyasi (SRL) saqlanadi
11	**Semantic_Role_Types**	role_label_id (PK), role_label_name	Semantik rollarning turlari ro‘yxati (agent, patient va h.k.)



3-rasm. Korpusning soddalashtirilgan ERD sxemasi

1. **Matnlar: Texts** (text_id, title, author, source, genre, created_at)
2. **Gaplar: Sentences** (sentence_id, text_id, sentence_no, raw_text)
3. **Tokenlar: Tokens** (token_id, sentence_id, position, form, lemma_id, upos_id, xpos, feats)
4. **Lemmalar: Lemmas** (lemma_id, lemma_text)
5. **Universal POS-teglar: UPOS_Tags** (upos_id, upos_name)
6. **Sintaktik bog‘lanishlar: Syntax_Links** (link_id, token_id, head_id, deprel_id)
7. **Sintaktik bog‘lanish turlari: Deprel_Types** (deprel_id, deprel_name)
8. **NER annotatsiyalari: Entities** (entity_id, sentence_id, token_start, token_end, entity_type_id)
9. **NER turlari: Entity_Types** (entity_type_id, entity_type_name)

10. **Semantik rollar: Semantic_Roles** (srl_id, predicate_id, argument_id, role_label_id)
11. **Semantik rollar turlari: Semantic_Role_Types** (role_label_id, role_label_name)

Bunday relatsion formatning afzalligi – *katta hajmdagi ma'lumotni samarali indekslash va qidirish:* masalan, SQL so‘rovi bilan ma'lum bir tegli so‘zlarni yoki ma'lum bir tuzilmani butun korpus bo‘ylab tezda topish imkoni. Bundan tashqari, korpusning ma'lumotlar yaxlitligini ta'minlash, takrorlarni yo‘qotish (normalizatsiya) kabi imkoniyatlar mavjud. Kamchiligi esa bunday bazani yaratish va undan foydalanish uchun katta resursli server(lar) talab qilinadi. Oddiy foydalanuvchi bunday formatni bevosita ko‘ra olmaydi (maxsus interfeys yoki SQL so‘rovlari kerak bo‘ladi). Shu sababli korpus avval CSV/CoNLL kabi formatlarda tayyorlanib, keyin qidiruv qulayligi uchun SQL Server yoki boshqa korpus manbasiga import qilinadi.

Yuqoridagi 3-rasmda ko‘rsatilgan ER tuzilmasida eng katta hajmga ega va qidiruvda eng ko‘p foydalaniladigani **Tokens** jadvalidir. Unda korpusdagi har bir so‘zning teglari haqida ma'lumotlar mavjud. Bunday yondashuv (inverted index) orqali qidiruv jarayoni ancha tezlashadi. Foydalanuvchi biror so‘z yoki lemma bo‘yicha izlaganda, tizim matnlarning o‘zini to‘liq ko‘zdan kechirmaydi, balki tayyor indeks jadvalidan kerakli qatorlarni tanlab oladi.

Zamonaviy korpus tizimlari (masalan, BNCweb/CQPweb) ham xuddi shunday uslubda *ikkinchi darajali indeks va metama'lumotlar bazasi kombinatsiyasidan* foydalanadi. Bunda IMS Corpus Workbench kabi maxsus matn qidiruv algoritmi indekslangan matn fayllarida qidiruvni amalga oshiradi, SQLServer kabi relatsion baza esa foydalanuvchi va so‘rovlar metadata'sini boshqaradi. Bizning holatda, SQL Serverning o‘zida indeks jadvali saqlangani uchun yagona jadval ichida ham metadata, ham matn qidiruvi hal bo‘ladi.

Qidiruv algoritmi. Korpusda qidiruvni *to‘g‘ridan-to‘g‘ri* va *teskari rejimlarda* amalga oshirish mumkin. To‘g‘ridan-to‘g‘ri qidiruv – bu aniq so‘z formasini yoki aniq lemmani izlash, teskari qidiruv – bu grammatik xususiyatlariga ko‘ra izlash (masalan, *"o‘tgan zamon fe'llari"* yoki *"ko‘plikda ishlatilgan otlar"* kabi).

3-rasmda ko‘rsatilgan ER tuzilmasiga ko‘ra:

1. So‘z bo‘yicha qidiruv: Foydalanuvchi kiritgan matn avvalo **Tokens** jadvalidan *token_id* topiladi. So‘ng shu *token_id* ga teng bo‘lgan barcha yozuvlar **Sentences** jadvalidan tanlanadi. Natijada ushbu so‘z uchragan barcha gaplar olinadi. Tizim bu ro‘yxatni foydalanuvchiga chiqishdan oldin saralashi va guruhlashi mumkin (masalan, matnlar bo‘yicha guruhlash, vaqt bo‘yicha tartiblash). Korpus dasturida ushbu jarayon uchun maxsus optimizatsiyalar qo‘llanadi: **Tokens** jadvalida *token_id* ustuniga indeks qo‘yilgan, shuningdek, to‘liq matn indeksi ham bo‘lishi mumkin. TNC (Turk milliy korpusi) misolida MySQL bazasida to‘liq matn

indeksi orqali qidiruv ishlangani va uni optimallashtirish jiddiy masala bo‘lgani haqida xabar berilgan. Bizning tizimda ham SQL Server’ning full-text indeksi so‘z bo‘yicha qidiruvni tezlashtiradi.

2. **Lemma bo‘yicha qidiruv:** bu holatda **Lemmas** jadvalidan *lemma_id* olinadi, so‘ng **Tokens** jadvalidan shu lemmega tegishli barcha qatorlar (ya’ni lemma_id ustuni bo‘yicha) tanlanadi. Amalda ko‘pincha foydalanuvchilar korpusda lemma bo‘yicha izlaydi, chunki bu barcha shakllarni qamrab oladi. Shuning uchun, qidiruv interfeysida *“Lemmaga ko‘ra izlash”* asosiy rejim bo‘lishi maqsadga muvofiq. Masalan, foydalanuvchi *“kelmoq”* lemmasini izlasa, natijada *“keldi”, “kelmoqda”, “kelar”* kabi barcha shakllar chiqadi[28]. Bu imkoniyat RNC (Rus milliy korpusi) va boshqa ko‘plab korpuslar interfeysida mavjud va foydali sanaladi.

3. **Morfologik xususiyatlar bo‘yicha (teskari) qidiruv:** bunda foydalanuvchi maxsus filtr orqali grammatik belgi tanlaydi (yoki so‘rov tilidan foydalansa, masalan, *pos="FE'L"* kabi). Tizim bunday so‘rovni **Tokens** jadvaliga *feats* yoki teglar orqali yozilgan shartga aylantiradi. Yuqoridagi jadval tuzilmasida *feats* maydoni morfologik belgilarni bitlar orqali kodlab saqlaydi. Masalan, 64 bitli ikki maydon sifatida saqlash varianti sinovdan o‘tkazilgan: bir maydon muayyan toifadagi belgilarni, ikkinchisi boshqa toifalarni ifodalaydi. Agar gram_code ikkilik niqob bilan solishtirilsa, masalan, *“fe’l”* va *“o‘tgan zamon”* bitlari 1 bo‘lsa, ushbu shartga mos barcha qatorlar chiqariladi[29]. Bu usul juda tezkor teskari qidiruvni ta’minlaydi. Tatar korpusi tizimida sinovlar natijasida kompleks morfologik shartlar bo‘yicha ham so‘rov bir soniyadan kam vaqtda bajarilishi erishilgan.

4. **Kombinatsiyalangan qidiruv:** foydalanuvchi bir vaqtning o‘zida so‘z/lemma va grammatik filtrlarni tanlashi mumkin. Bunda yuqoridagi mexanizmlar birgalikda ishlaydi: dastlab lemma yoki so‘z bo‘yicha potensial natijalar chiqib, so‘ng ular orasidan grammatik filtrlarga mos bo‘lmaganlari so‘raladi. Masalan, *“ot”* so‘z turkumiga mansub kitob so‘zini izlash uchun dastur avval kitob lemmasini topadi, so‘ng **Tokens** jadvalidan shu lemma bo‘yicha barcha yozuvlarni olib, ular orasidan *xpos = “N”* bo‘lganlargina qoldiriladi. Natijada, agar so‘z ba’zan fe’l yoki sifat sifatida ham belgilangan bo‘lsa, ular natijadan chetlanib, faqat ot sifatidagilari ko‘rsatiladi.

Korpus tizimi ko‘p foydalanuvchili rejimda ishlashi uchun, indeks va so‘rovlarni bajarish qismi optimallashtirildi. Bizning axborot tizimida SQL Server ichki kesh mexanizmlari va to‘g‘ri indekslash orqali yuqori samaradorlikni ta’minladi.

Qidiruv natijalari odatda konkordans ko‘rinishida – ya’ni kalit so‘z o‘rtada, chap va o‘ng tarafda ma’lum miqdor kontekst bilan – foydalanuvchiga taqdim etiladi. Veb-interfeys bunda maxsus HTML shablonlar orqali har bir natijani formatlaydi: masalan, kalit so‘zni qalin yoki rangli qilib belgilashi, ustiga bosganda batafsil ma’lumot ko‘rsatishi mumkin. TNC tizimida natijalar oynasida har bir konkordans qatoriga bosilganda shu so‘zning matndagi kengroq konteksti va

manbasi haqidagi ma’lumotlar alohida oynada ochilishi joriy etilgan. O‘zbek korpusi interfeysida ham shunday qulayliklar ko‘zda tutilgan. Foydalanuvchi natijani tahlil qilayotgan paytda shu joyning to‘liq paragrafini yoki butun hujjatni ochib ko‘rishi, kerak bo‘lsa shu matnda boshqa so‘zlarni izlashga o‘tib ketishi mumkin.

Korpus ma’lumotlar bazasi tuzilishi va qidiruv algoritmlarini loyihalashda asosiy maqsad keng qamrovli leksik-statistik so‘rovlarni eng qisqa vaqtda bajarishdan iborat. Hozirda dunyoning yirik korpus tizimlari bu borada turli yondashuvlarni qo‘llashadi. Britaniya milliy korpusi (BNC) uchun ishlab chiqilgan BNCweb tizimi IMS Corpus Workbench kabi C++ da yozilgan optimallashtirilgan qidiruv tizimini MySQL bilan birlashtirish orqali ham tezlik, ham qulaylikka erishgan. Rus milliy korpusida esa dastlabdan moslashtirilgan o‘z qidiruv mexanizmi yaratilgan bo‘lib, u rus tilining boy morfologiyasini hisobga olgan holda lemma va grammatik belgilarga ko‘ra tezkor izlashni imkonini beradi. Yangi platformalarda (masalan, NoSketch Engine va uning KonText interfeysi) esa ochiq kodli korpus izlash mexanizmi va moslashuvchan veb-interfeys uyg‘unlashgan. Chex milliy korpusi ham hozir KonText’dan foydalanadi va uni boshqa resurslar bilan integratsiya qilish imkoniyatiga ega.

Umuman, o‘zbek milliy korpusi bazasi va qidiruv algoritmlari yuqoridagi ilg‘or tajribalarni inobatga olgan holda ishlab chiqildi.

Teglash va ma’lumotlarni qayta ishlash modullarining ishlash algoritmlari. Korpusda lingvistik teglash – bu murakkab va ko‘p bosqichli jarayon. O‘zbek tilining xususiyatlarini hisobga olgan holda, milliy korpusni teglash algoritmlarini quyidagi bosqichlarga ajratish mumkin:

1. Oldindan ishlov (preprocessing): Korpushga yangi qabul qilingan matn dastlab normalizatsiya qilinadi. Bu bosqichda matndagi keraksiz belgilar tozalanadi, kodlash tizimi bir me’yorga keltiriladi. Shuningdek, matndagi xatoliklar, noto‘g‘ri ajratilgan so‘zlar aniqlansa, minimal avtomatik tuzatishlar qo‘llaniladi.

2. Tokenlash: Matnni kichik birliklarga – avvalo gaplarga, so‘ngra so‘z va belgilarga ajratish algoritmi bajariladi. Bu jarayon uchun odatda regulyar ifodalar va qoidalar to‘plami tuziladi. Masalan, gaplarni ajratishda nuqta, so‘roq, undov belgisidan foydalaniladi (abbreviaturalarda uchraydigan nuqtalarni e’tiborga olgan holda). So‘zlarni ajratishda oraliq (probel)lar va tinish belgilari asosiy chegara hisoblanadi. Tokenlash natijasida matn [so‘z, so‘z, ..., tinish belgisi, ...] ko‘rinishidagi ketma-ketlikka ajraladi. Ushbu bosqich sifatli amalga oshirilishi keyingi tahlillar samaradorligi uchun juda muhim.

3. **Morfologik tahlil** (lemmani aniqlash, teglash): Har bir ajratilgan so‘z uchun uning lug‘aviy shakli (lemma) va grammatik kategoriyalari aniqlanadi. O‘zbek tilida bu bosqich nisbatan murakkab, chunki qo‘shimchalar tizimi boy va bir so‘z turli ma’nolar berishi mumkin. Morfologik tahlil moduli bunday noaniqliklarni hal qilish uchun lug‘at va qoidalar majmuasidan foydalanadi. O‘zbek tili uchun asl

so‘z formalarini lug‘atdan izlab topish va qo‘shimchalar ketma-ketligini qoidalarga solishtirish qo‘llanildi. Natijada har bir so‘zga masalan, quyidagicha teglar to‘plami biriktiriladi: *“qilsa” – <qil, fe’l, shart mayl, 3-shaxs>.*

4. Sintaktik tahlil (so‘z va gap bo‘laklari): Bu bosqich murakkabroq bo‘lib, barcha korpuslarda ham to‘liq amalga oshirilavermaydi. Ko‘pincha faqat ilmiy-tadqiqot uchun kichik korpuslarda qilinadi. Ammo bizning o‘zbek milliy korpusida sintaktik bog‘lanishlarni ham teglash amalga oshirilgan. Sintaktik tahlil uchun odatda qora quti usulidagi modellar yoki maxsus qoidalar asosida ishlaydigan dasturlar kerak. Sintaktik tahlil natijasida har bir gap uchun sintaktik daraxt yoki bog‘lanishlar ro‘yxati hosil bo‘ladi. Bu ma’lumotlar korpus qidiruvida, masalan, *“S + P tuzilmasidagi gaplar”* yoki *“to‘ldiruvchi qaysi so‘z bilan kelgan”* kabi so‘rovlarni qayta ishlash imkonini taqdim etadi.

5. Semantik teglash: Bu hozircha eksperimental darajada bo‘lsa-da, ayrim korpuslarda qo‘llaniladi. O‘zbek tilida bu yo‘nalish endi rivojlanmoqda, shuning uchun milliy korpusning ilk talqinida semantik teglash tizimi ishlab chiqilmoqda.

Teglash modullarini qo‘llash natijasida tizim samaradorligini oshirish uchun ba’zan oraliq keshlash joriy etiladi. Misol uchun, agar bir xil so‘z formasi ko‘p marta tahlil qilinayotgan bo‘lsa, bir marta hisoblangan natijani vaqtinchalik xotirada saqlab, keyingi safar shu natijani qayta ishlatish mumkin. Bu **dinamik dasturlash tamoyili** katta korpuslarni teglashda qo‘llanadi.

Barcha lingvistik teglash natijalari korpus ma’lumotlar bazasida tegishli jadvallarda saqlab boriladi. Shuningdek, TEI XML formatidagi to‘liq teglangan matnlar ham alohida fayllarda saqlanishi mumkin. Bu yondashuv asosan zaxira va tashqi tizimlarga almashish uchun qulay. Ko‘pgina korpuslarda (masalan, RNC va BNC) matnlarning to‘liq teglangan nusxalari XML ko‘rinishida saqlanadi va foydalanuvchilarga yuklab olish uchun ham taklif etiladi.

Teglash jarayonida imkon qadar *noaniqliklar (disambiguatsiya)* bartaraf qilinadi. Masalan, yuqorida keltirilgan *“olma”* misolida bu so‘zning otmi yoki fe’lmi ekanini aniqlash uchun kontekstga murojaat qilinadi: “Qizim idishni olma” gapida *“qilsa”* fe’l, *“Olma juda qimmat”* iborasida esa *“olma”* ot kabi teglanadi. Bunday farqlarni aniqlash uchun maxsus qoidalar yoki ehtimollik modellari ishlatiladi. Masalan, HMM (Yashirin Markov modeli) asosida POS-teglash usullari so‘zlar ketma-ketligidan kelib chiqib eng ehtimolli teglar kombinatsiyasini tanlaydi. Yoki neyron tarmoqlar yordamida har bir so‘z atrofidagi vektorlarga asoslanib teg belgilashda xatolar kamaytiriladi. Hozirda o‘zbek tili uchun ochiq manbalarda katta teglangam korpuslar mavjud emasligi sabab, qoidalarga asoslangan yondashuv asosiy variant hisoblalanadi. Lekin korpusning yaratilishi o‘zi keyinchalik shu maqsadda – yangi statistik/mashinali o‘qitish modellarga ma’lumot bazasi bo‘lib xizmat qiladi. Milliy korpus hajmi oshgani sari, uning ichidan avtomatik ravishda teglangan ma’lumotlardan modelni o‘qitib, keyin yanada sifatliroq teglar qo‘yish orqaga bog‘lanish (feedback loop) usullar bajariladi.

Teglash moduli kiruvchi matnni olib, chiqaruvchi teglangan ma’lumotlarni beradi. Bu bir turdagi matnlarni **qayta ishlash konveyeri (pipeline)**dir. Har bir bosqich (tokenlash, POS-teglash, morfolofik, sintaktik va semantik tahlil) navbatma-navbat o‘tib, natijani boyitib boradi. Shuning uchun, modulning har bir qismi aniq algoritmik vazifani bajaradi va ularni alohida funksiyalar yoki xizmatlar sifatida tatbiq etish maqsadga muvofiq.

TNC misolida aynan shunday uyg‘unlashgan jarayon tashkil etilgan: matnlar boshida imlo tekshiruvidan o‘tkazilgan, so‘ngra SVN versiya nazorati orqali jamoa a’zolari tomonidan tozalangan, keyin NooJ yordamida morfologik teglangan. Jarayon davomida bir necha nazorat nuqtalari bo‘lib, sifatni tekshirish va xatolarni tuzatish qo‘lda amalga oshirilgan (har bir matn ekspert lingvistlar tomonidan ko‘rib chiqilgan). O‘zbek tili milliy korpusi uchun ham dastlabki davrda avtomatik teglash natijalarini lingvist mutaxassislar ko‘rib, xatolarni to‘g‘rilab borishi lozim. Ayniqsa, murakkab yoki ikkilanishli holatlar bo‘yicha qoidalar bazasini boyitish uchun. Bu ishlar bosqichma-bosqich korpus hajmi kengayishi bilan takrorlanadi.

Teglangan korpus foydalanuvchilar uchun bir necha *qatlamli filtrlarni taqdim etadi*: foydalanuvchi faqat so‘z shakli bo‘yicha emas, balki teglar bo‘yicha ham izlash imkoniyatiga ega bo‘ladi. Masalan, interfeysda *“Fe’l va sifatlarni topish”* uchun maxsus forma elementlar ishlab chiqiladi. Yoki *“[OT + sifat + OT] birikmalarini qidirish”* kabi murakkabroq so‘rovlar ham til birliklarini teglash orqaligina bajarilishi mumkin bo‘lgan vazifa hisoblanadi. Bunday murakkab qidiruvlar **CQP (Corpus Query Processor)** singari maxsus so‘rov tillari yordamida yoki foydalanuvchi uchun soddalashtirilgan interfeys elementlari orqali ta’minlanadi. RNC dagi maxsus *“Izlash tili”* oynasida foydalanuvchilar masalan, *[lemma="kitob" & pos="S"]* kabi ifodalar kiritishi mumkin. Bizning tizim esa buni foydalanuvchi uchun yashirin tarzda, oddiy “kitob (ot)” shaklida tanlash orqali bajarishi mumkin.

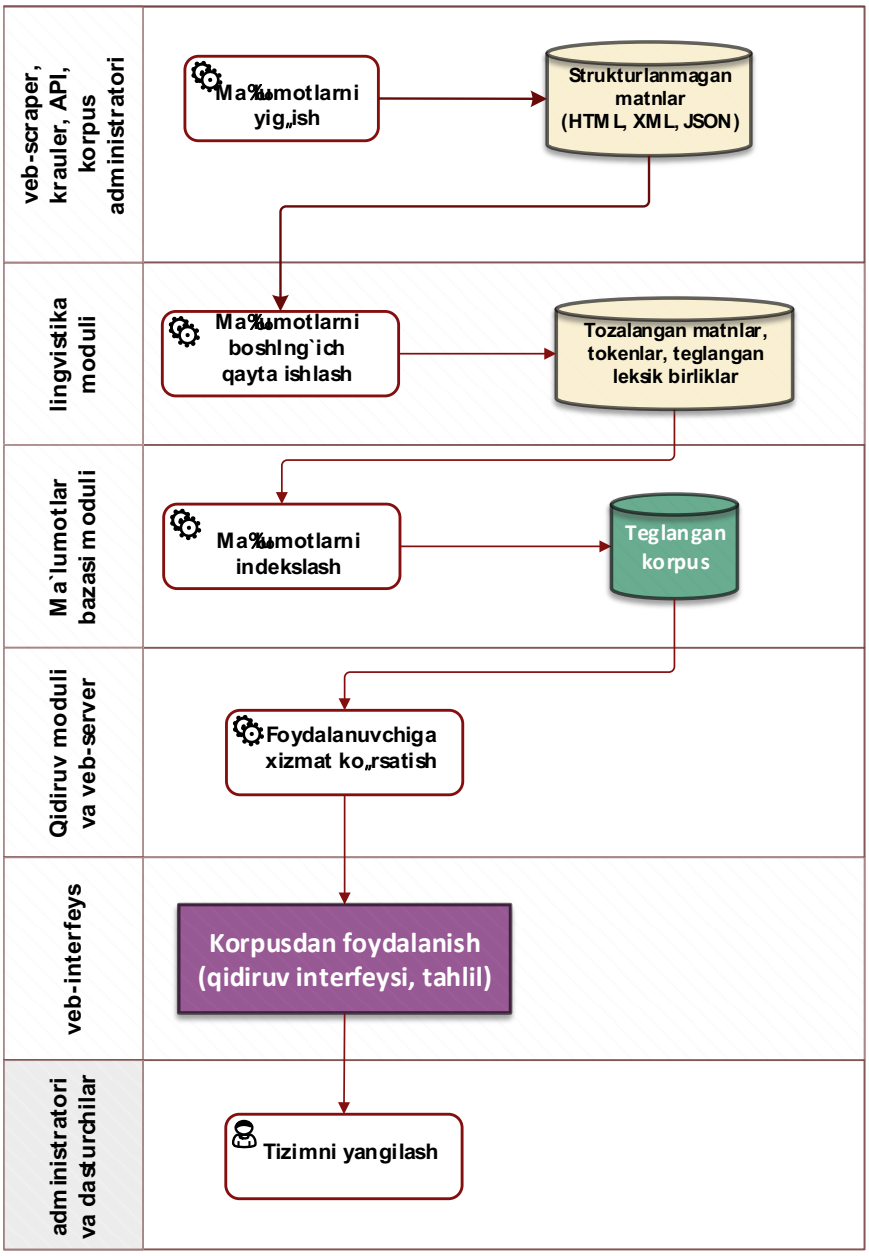
Yuqoridagi barcha algoritmik yechimlar o‘zbek tili milliy korpusini murakkab axborot tizimi sifatida tavsiflaydi. Unda axborot oqimi kirish (matnlar)dan chiqish (natijalar)gacha qat’iy teglangan qadamlar orqali oqadi, har bir qadamda ma’lumotlar shakli o‘zgaradi va boyiydi. Bu jarayonlarni formal tarzda ifodalash mumkin – masalan, funksional xaritalar yordamida:

$$\text{QidiruvNatijasi} = \text{F}(\text{qidiruv so‘rovi}, \text{filtrlar}) =$$

$$\{\text{x} \in \text{Korpus}: \text{shartlar}(\text{x}, \text{qidiruv so‘rovi}, \text{filtrlar})\}$$

bu yerda **shartlar()** – foydalanuvchi tanlagan so‘z yoki lemma va grammatik talablarga moslikni tekshiruvchi mantiqiy funksiya, **Korpus** – barcha matnlar to‘plami. Natijada **QidiruvNatijasi** – ushbu shartlarni qanoatlantiruvchi korpus elementlari to‘plamidan iborat bo‘ladi. 2-rasm ayni paytda ma’lumot oqimini tushuntiruvchi DFD rolini ham o‘ynaydi: unda ma’lumotlarning qayerdan qayerga oqayotganini va qanday o‘zgarishlar sodir bo‘lishini ko‘rish mumkin. O‘zbek milliy korpusining hayotiy sikli (biznes-jarayonlari) quyidagi 4-rasmda keltirilgan.

4-rasmda har bir bosqichda biznes-jarayon ishtirokchilari va ularning vazifalari aniq belgilangan. Masalan, foydalanuvchi qidiruv so‘rovi bilan murojaat qilganda, bu “Axborot izlash” biznes jarayonini ishga tushiradi. Axborot tizimida esa bunga mos jarayonlar zanjiri ketma-ketlikda amalga oshadi: so‘rovni qabul qilish, interpretatsiya, qidiruv, natijani shakllantirish, qaytarish.



4-rasm. O‘zbek milliy korpusining hayotiy sikli (biznes-jarayonlari)

Xorijiy korpuslar arxitekturasi bilan qiyosiy tahlil. Dunyodagi yirik milliy korpuslar (masalan, Britaniya milliy korpusi – BNC, Rus milliy korpusi – RNC, Turk milliy korpusi – TNC, shuningdek, Chex, Polsha, Amerikaning ingliz korpuslari va h.k.) yaratilib, ularning har biri o‘ziga xos arxitekturaviy yechimlarni sinovdan o‘tkazgan. O‘zbek tili milliy korpusi arxitekturasini loyihalash jarayonida

ushbu xorijiy tajribalar chuqur o‘rganilib, ularning yutuq va kamchiliklari hisobga olindi.

Britaniya milliy korpusi (BNC) – 100 milliondan ortiq so‘zdan iborat ingliz tili milliy korpusi bo‘lib, dastlab matnlarni SGML formatida teglangan holda CD disklar orqali tarqatilgan. Uning arxitekturasi va modeli zamonaviy ko‘plab korpuslarga asos bo‘lib xizmat qilgan [26]. BNCning asosiy tamoyili – muvozanatli (balanced) korpus yaratishdan iborat. BNC korpusida turli janr, uslub va mavzudagi matnlarni ahamiyatiga ko‘ra muayyan foizlarda kiritilgan. TNC korpusini yaratishda masalan, BNC ning janrlar bo‘yicha taqsimoti nisbatan kichikroq hajmga proyeksiya qilingan. BNC arxitekturasi dastlab foydalanuvchiga to‘g‘ridan-to‘g‘ri ochiq interfeys taqdim etmagan, lekin keyinchalik BNCweb deb nomlangan veb-interfeys ishlab chiqilgan. BNCweb arxitekturasi mijoz-server modeliga asoslangan bo‘lib, u ikki asosiy qismdan iborat: orqa fonda IMS Open Corpus Workbench (CWB) – yuqori samarali matn qidiruv moduli, MySQL ma’lumotlar bazasi va CGI skriptlari orqali veb-interfeys ishlagan. Bu yechim o‘z davrida innovatsion bo‘lib, foydalanuvchi uchun qulaylik (veb interfeys) va kuchli qidiruv imkoniyatlarini uyg‘unlashtirishga erishgan. CQPweb deb nomlangan keyingi tizim aynan BNCweb g‘oyasini umumlashtirib, har qanday korpusga moslangan [27].

BNCning afzalligi – aniq teglangan TEI SXEMAsi va metadata strukturasiga ega ekani bo‘lib, kamchiligi – dastlab onlayn qidiruvning murakkabligi (faqat maxsus dasturlar orqali) edi. Hozirgi vaqtda BNC CQPweb interfeysi orqali internetda ochiq, lekin u yerdagi so‘rovlar tilini o‘rganish biroz vaqt talab qilishi mumkin. Milliy korpuslar rivojida BNC namunasi shuni ko‘rsatdiki, korpus tuzilishi (mazmun va balans) va texnik arxitektura (foydalanish qulayligi) muhim jihatlardir.

Rus milliy korpusi (RNC) – 300 milliondan ortiq so‘zdan iborat rus tili uchun yaratilgan korpus arxitekturasi jihatidan juda qiziqarli obyekt hisoblanadi. RNC bir nechta mustaqil (sub) korpuslar majmuasidan iborat: *Asosiy korpus, So‘zlashuv korpusi, Poetik korpus, Tarixiy korpuslar, Dialektal korpus* va hokazo, barchasi yagona sayt orqali qidirish imkonini beradi. Bu multi-korpus arxitekturasi maxsus modul sifatida ko‘rilishi mumkin. Ya’ni, bitta tizim doirasida turli korpuslarni tanlash va qidirish. RNC texnik ko‘rinishda C++ va Perl dasturlash tillarida yozilgan bo‘lib, unda rus tilining boy morfologiyasi uchun maxsus qidiruv vositalari mavjud. Foydalanuvchi interfeysi bir necha tilda (rus, ingliz) taqdim etilgan, va unda morfologik filtrlash qudratli qidiruv shakllari orqali amalga oshadi. Masalan, foydalanuvchi oddiy rejimda so‘z kiritib izlaganda, tizim avtomatik lemmatizatsiya qilib qidiradi. Ya’ni, barhca grammatik shakllarni qamrab oladi. Kengaytirilgan rejimda esa CQL (Corpus Query Language)ga yaqin maxsus sintaksisdan foydalanish imkoniyati mavjud. Shungdek, ushbu korpusda sodda holda tushunarli interfeys elementlari bilan grammatik belgilar tanlash mumkin. RNC arxitekturasida dastlab qidiruv mexanizmi sifatida Tomita-parser va Yandex ishlab chiqqan MyStem analizatori integratsiya qilingan edi, keyinchalik yanada takomillashgan vositalar qo‘shildi.

RNCning afzallik tomonlari – juda katta hajmdagi teglangan (morfologiya 100% avtomatik, sintaksisli subkorpuslar ham bor) va qidiruv imkoniyatlari keng. Shuningdek, RNC parallel korpuslar (tarjima matnlari) va lug‘atlar bilan integratsiyani ham yo‘lga qo‘ygan. Foydalanuvchi qidirgan so‘zining lug‘aviy ma’nosini ko‘rishi yoki shu so‘zning boshqa tillardagi tarjimalar korpusini izlashi mumkin. Bu, albatta, arxitekturaviy jihatdan qo‘shimcha modullar *(lug‘at xizmati, parallel qidiruv moduli)* orqali amalga oshirilgan. RNCning kamchilik tomoni esa – uning dasturiy bazasi ancha eski va optimizatsiya qilingan qo‘lda yozilgan kod bo‘lib, zamonaviy mikroxizmatlar arxitekturasiga mos kelmaydi. 2022–2023 yillarda RNC saytining yangi interfeysi va tizimini yaratish ustida ish olib borilib, foydalanuvchilar uchun yanada qulay dizayn va tezkor qidiruv joriy qilindi. Yangi RNC arxitekturasi NoSketch Engine asosida ishlovchi KonText interfeysiga o‘xshash ochiq platformaga asoslangan.

Turk milliy korpusi (TNC) – 50 million so‘zli turk tili korpusi arxitekturasi haqida rasmiy maqolada batafsil yozilgan bo‘lib [26], u asosan BNC modeli asosida qurilganini qayd etish lozim. TNC korpusini ishlab chiqishda imkon qadar ochiq kodli dasturlardan foydalanilgan. Ularning ma’lumotlari MySQL bazasida saqlanib, foydalanuvchi interfeysi PHP/JavaScript yordamida yaratilgan. TNCda ham matnlar dastlab lingvistlar tomonidan tekshirilib, maxsus teglash qoidalari asosida NooJ lingvistik dasturi bilan morfologik teglangan. Qidiruv tizimi quyidagi arxitekturasida MySQLning MyISAM mexanizmidagi to‘liq matn indeksidan foydalangan holda turk tilidagi affikslarni qidirish uchun wildcard uslubidan foydalanilgan. Masalan, foydalanuvchi interface’da *“so‘zi ... bilan tugaydigan”* deb tanlasa, PHP kod orqa fonda %... kabi LIKE operatsiyasidan yoki to‘g‘ridan-to‘g‘ri full-text indeksning INFIX qidiruv imkoniyatidan foydalangan. Bu usul murakkabroq lemmatizatsiya o‘rnini bosishi ma’lum darajada mumkin. TNC foydalanuvchi interfeysi imkon qadar soddalashtirilgan. Foydalanuvchi SQL so‘rovi o‘rniga bir nechta forma elementlari bilan izlashni amalga oshira oladi. Masalan: “[iyor bilan tugaydigan so‘zlarni qidir]” kabi misolda “-iyor” affiksi bilan tugaydigan fe’llarni badiiy matnlardan qidirish misoli keltiriladi.

TNC arxitekturasi afzalliklari sifatida web asosidagi oddiy yechim, bepul vositalardan foydalanishni, kamchiligi sifatida esa ayrim qidiruvlar uchun to‘liq morfologik normalizatsiya mavjud emasligini (yani, lemmatizatsiya to‘liq emas, foydalanuvchi affiks bo‘yicha qo‘lda izlashi kerak) hamda MySQL bazasida juda katta hajmda matn saqlanganda samaradorlik muammolari chiqishi mumkinligini keltirish mumkin. TNC turk tilida ilk yirik korpus bo‘lib, ilmiy tadqiqotlar uchun muhim manba hisoblanadi va u internetda ochiq holda taqdim etilgan.

Xulosa qilib aytganda, xorijiy korpuslar arxitekturasini qiyoslasak, quyidagi umumiy tendensiyalarni ko‘rish mumkin.

2-jadvaldan ko‘rinadiki, har bir milliy korpus o‘z oldiga qo‘ygan maqsad va resurslardan kelib chiqib turli texnik yechimlarni tanlagan. BNC va uning vorisi CQPweb maxsus korpus mexanizmiga asoslangan bo‘lsa, RNC dastlab maxsus

dasturiy yechim bo‘lib, tilga moslashuv va hajmga urg‘u bergan. TNC esa imkon qadar soddalik va amaliylikni maqsad qilgan. TNCda mavjud vositalar yordamida tez fursatda milliy korpusni yaratish mumkin. O‘zbek tili milliy korpusi esa bu tajribalarni jamlab, servisga yo‘naltirilgan modul arxitektura asosida, zamonaviy veb texnologiyalar (Django/Python) yordamida, shu bilan birga tilimizga xos chuqur lingvistik modellarni integratsiya qilgan holda ishlab chiqilgan.

2-jadval

Milliy korpuslar arxitekturasi qiyosiy tahlili (asosiy xususiyatlar)

Xususiyat	BNC (Angliya)	RNC (Rossiya)	TNC (Turkiya)	O‘zbek milliy korpusi (loyiha)
Tuzilish (hajm/janr)	100 mln. so‘z; tekis taqsimotli janrlar	~300 mln. so‘z; subkorpuslarga bo‘lingan (asosan yozma matnlar, turli tarixiy va mintaqaviy)	50 mln. so‘z; BNC modeliga asoslangan balans	~100 mln so‘z; tekis taqsimotli, turli uslub/janrlarni qamrab olgan
Platforma va texnologiya	IMS CWB + MySQL; CGI-BIN asosida veb (BNCweb)	Maxsus C++ dastur + Perl CGI; 2023 yildan yangilangan	MySQL (MyISAM) + PHP + JS; morfologiya uchun NooJ	SQL Server + Python (Django); modul SOA arxitekturasi; morfologiya uchun maxsus analizator.
Qidiruv imkoniyatlari	CQP so‘rov tili; lemmatizatsiya; BNCweb/CQPweb interfeysi orqali kollokatsiya, chastota.	Murakkab morfologik va sintaktik qidiruv; parallel va diaxron qidiruv; grafik interfeys (grammatik filtrlar)	Oddiy web interfeys, lemma o‘rniga affiks bo‘yicha ham qidiruv; filtrlar (janr, sana, muallif)	Qidiruv turlari: so‘z/lemma, morfologik filtrlar, birikmalar; foydalanuvchiga sodda interfeys + ilg‘or rejim (CQL/REGEX) ko‘zda tutilgan.
Teglash darajasi	Morfologik (CLAWS5 POS tag), ba’zi semantik teglar.	To‘liq morfologik (Sharov tagset); ba’zi subkorpuslarda sintaksis; semantik teglash.	To‘liq morfologik teglangan (NooJ rule-based)	To‘liq morfologik teglangan; sintaksis jarayonda; NER teglash mavjud.
Jamoatchilikka ochiqlik	Ilk tarqatish offlayn (CD), hozir onlayn ro‘yxatdan o‘tish bilan bepul.	To‘liq ochiq onlayn; API cheklangan (faqat interfeys orqali).	Ilmiy foydalanish uchun bepul (onlayn veb).	To‘liq ochiq (veb + API); litsenziyalar qonun doirasida
Afzalliklar	Muvozanatli, standarti yuqori;	Murakkab lingvistik qidiruv;	O‘z tiliga moslashtirilgan qoidalar;	Zamonaviy arxitektura;

	ko‘plab vositalar bilan mos.	bir qancha til resurslari integratsiyasi.	sodda interfeys.	tilning o‘ziga xos jihatlariga mos maxsus modullar.
Kamchiliklar	Dastlab murakkab foydalanish (CQP o‘rganish zaruriyati).	Eski platforma (eskirgan texnologiyalar, yangilanmoqda).	Lemma qidiruvining chekligi; hajm oshganda MySQL cheklovlari.	Test rejimida ishlamoqda; katta hajmda sinovdan o‘tmagan.

O‘zbek tilining milliy korpusi nafaqat tilshunos olimlar, balki tarjimonlar, adabiyotshunoslar, pedagoglar va shu tilni o‘rganayotgan keng jamoatchilik uchun ham ulkan manba bo‘lib xizmat qiladi. Uning arxitekturasi puxta o‘ylangan holda ishlab chiqilsa, turli soha va ilovalarga integratsiya qilish oson bo‘ladi. Masalan, kelgusida mashina tarjimasi tizimlari, nutqni tanish dasturlari yoki intellektual til o‘rgatish platformalari milliy korpusning API orqali kerakli ma’lumotlarini olib foydalanishi mumkin bo‘ladi. Shu sababdan, korpusning ochiq API xizmatlarini rivojlantirish hamda ma’lumotlarni umumqabul qilingan formatlarda taqdim etish juda muhim.

Ayni paytda O‘zbekistonda ushbu yo‘nalishda qator ishlar boshlangan: 2018 yilda birinchi bor matbuotda milliy korpusni yaratish zarurati tilga olingan, 2019 yilgi Prezident farmonida davlat tilini raqamli qo‘llab-quvvatlash haqida alohida band kiritilgan. 2020–2030-yillarga mo‘ljallangan til siyosati konsepsiyasida elektron milliy korpus yaratish vazifasi belgilandi. 2021–2023-yillarda Alisher Navoiy nomidagi Toshkent davlat o‘zbek tili va adabiyoti universitetida “Kompyuter lingvistikasi: muammo, yechim va istiqbollar” anjumanlari doirasida o‘zbek milliy korpusi mavzusida tadqiqotlar e’lon qilindi.

Jumladan, Elov B.B. milliy korpusning til modeli haqida ma’ruzalar qildi. Shuningdek, 2022 yilda o‘zbek tili milliy korpusi uchun maxsus veb-sayt (https://uzschoolcorpara.uz/) ishlab chiqildi. Mavjud ayrim ochiq loyihalar (masalan, uzbekcorpus – kichik hajmli ochiq matnlar to‘plami, Uzbek Word Embeddings – korpus asosida vektor modellari va hokazo) paydo bo‘ldi. Bular milliy korpus uchun texnik asos hozirlashda va keyingi algoritmlarni sinashda qimmatli tajriba beradi.

O‘zbek milliy korpusi arxitektura, model va algoritmlari bo‘yicha yuqorida bayon etilgan ilmiy-texnik yechimlar, albatta, doimiy ravishda takomillashtirib boriladi. Xorijiy korpuslar tajribasi shuni ko‘rsatadiki, korpus hech qachon *“yakunlangan”* holatga kelmaydi. U doimo rivojlanuvchi tizim. Shu bois, arxitekturaning moslashuvchan va kengaytiriluvchan bo‘lishi eng asosiy tamoyillardan biri sifatida qaralgan. Servisga yo‘naltirilgan modul arxitektura aynan shunday moslashuvchanlikni beradi: yangi modul qo‘shish (masalan, dialektlar korpusi, parallel korpus) yoki xizmatni almashtirish (masalan, boshqa MBga o‘tish) tizimning qolgan qismiga minimal ta’sir ko‘rsatadi.

Kelgusida milliy korpus asosida “O‘zbek tilining ta’limiy korpusi” (masalan, maktab darsliklari va o‘quv materiallari korpusi) yoki ixtisoslashgan kichik korpuslar (masalan, she’riy uslub korpusi, rasmiy hujjatlar korpusi) kabi turdosh loyihalarni yaratish imkoni paydo bo‘ladi. Texnik jihatdan bunday yangi korpuslar mavjud infrastruktura bazasida modul sifatida qo‘shilishi mumkin. Bu ham korpus arxitekturasi servisga yo‘naltirilgan bo‘lishining foydasidir. Shuningdek, korpusni boyitish uchun xalqaro standartarga mos XML/JSON eksport funksiyalari joriy etiladi. Tadqiqotchilar istalgan biror matn to‘plamining belgilangan XML variantini yuklab olib, o‘z dasturlarida tahlil qilishlari mumkin bo‘ladi. Masalan, Sketch Engine kabi tijoriy korpus platformalariga o‘zbek korpusini import qilish, yoki aksincha, ulardagi tayyor vositalardan foydalanish uchun eksport qilish amaliy jihatdan ahamiyatli bo‘lishi mumkin.

Arxitekturaviy nuqtai nazardan, milliy korpusni yaratish jarayonida tanlangan yechimlarning afzallik va kamchiliklarini qisqa ko‘rib chiqamiz: Python/Django bazasi – bu tez prototiplash va ishlab chiqish uchun qulay (afzallik), ammo C/C++ da yozilgan maxsus qidiruv mexanizmlari darajasida tezlikni bermasligi mumkin (kamchilik). Biroq, hozirgi zamon hardware imkoniyatlari va yaxshi optimizatsiya qilingan algoritmlar (indekslar, keshlar) bu farqni minimal darajaga tushiradi. SQL Server – korporativ darajadagi kuchli MBBT, lekin ochiq emas (ya’ni bepul emas). Kelgusida, ehtimol, PostgreSQL kabi ochiq MBBT ga o‘tish masalasi ham ko‘rilishi mumkin. Lekin ayni damda ko‘p tashkilotlarda mavjud infratuzilmani hisobga olgan holda SQL Server tanlanishi yomon emas, chunki u o‘zbek tilidagi ma’lumotlarga Unicode ko‘llab-quvvatlashi, full-text qidiruv imkoniyatlari, katta hajmlarni boshqarish quvvati bilan mos keladi.

Xulosa o‘rnida aytish mumkinki, o‘zbek tili milliy korpusi uchun taklif etilgan arxitektura, modeli va algoritmlari ilmiy asoslangan, xorijiy tajribaga tayangan va amaliy amalga oshirishga yo‘naltirilgan yechimlar majmuasidir. Bu korpusni yaratish va rivojlantirish nafaqat tilshunoslikda, balki axborot texnologiyalari sohasida ham yirik qadam bo‘ladi. Natijada, ona tilimizning raqamli maydondagi nufuzi oshishiga, zamonaviy NLP (Natural Language Processing) ilovalarida o‘zbek tilining qo‘llanishi kengayishiga xizmat qiluvchi fundamental resurs shakllanadi.

III. XULOSA

Ushbu ishda o‘zbek tili milliy korpusi uchun mikroxizmatlarga ajratilgan, kengayuvchan va standartlarga tayanadigan arxitektura taklif etildi. Yechim TEI belgilash va Universal Dependencies (UD) tamoyillarini markazga qo‘yib, yig‘ish→ normallashtirish → teglash → indekslash → qidirish → analitika jarayonlari orqali matnlarni izchil qayta ishlashni ta’minlaydi. Amaliy jihatdan, lingvistik teglash (tokenlash, lemmalash, POS/morfologik xususiyatlar, sintaktik bog‘lanishlar) va korpus analitikasi (KWIC, kollokatsiya metrikalari, n-gram chastotalari) bir xil ma’lumot modeli va API-lar bilan uyg‘unlashtirildi. Qidiruv qatlamida invers indekslash va “facet”larga tayanuvchi yondashuv ko‘p mezonli filtrlashni tez va

barqaror bajaradi; bu esa foydalanuvchining tahlilini deyarli real vaqt rejimida amalga oshirishni ta’minlaydi.

Arxitekturaviy tanlovlar (mikroxizmatlarga ajratish, navbatlar, qayta ishlashni optimallashtirish) tizimni kengaytirish, parallel ishlov berish va ishonchlilik nuqtai nazaridan maqbul qiladi. Django asosidagi veb-interfeys va REST API tadqiqotchilar, dasturchilar hamda keng foydalanuvchi auditoriyasi uchun yagona foydalanish interfeysini beradi; SQL darajasidagi optimallashtirilgan tuzilma esa (lemma/teg bo‘yicha indekslar, hujjat meta ma’lumotlari, to‘liq matn indekslari) morfo sintaktik so‘rovlarni barqaror tezlikda bajarishga xizmat qiladi. Taklif etilgan yechim turkiy tillarning agglutinativ xususiyatini inobatga olgan holda, lemma markaziyligiga tayanadi va ko‘p qiymatli grammatik xususiyatlarni mashinaga qulay formatda saqlash modelini qo‘llaydi.

Taqdim etilgan arxitektura milliy korpusni barqaror, kengayuvchan va ilmiy jihatdan foydalanuvchan platforma sifatida shakllantiradi. Ushbu poydevor ustida korpus qamrovini kengaytirish, yuqori darajali teglashlarni joriy etish, interaktiv analitika va integratsiya imkoniyatlarini boyitish orqali o‘zbek tilining raqamli ekotizimidagi o‘rni sezilarli kuchayadi; natijada, tilshunoslik tadqiqotlaridan tortib ta’lim va sun’iy intellekt ilovalarigacha bo‘lgan keng ko‘lamdagi qo‘llanishlarda milliy korpus markaziy resursga aylanadi.

FOYDALANILGAN ADABIYOTLAR

1. A. Akın and M. D. Akın, “Zemberek: an opensource NLP framework for Turkic languages,” Structure, vol. 10, pp. 1–5, 2007. [Online]. Available: https://citeseerx.ist.psu.edu/document?doi=82b22dbb84c5bff7352fe93b80f66a1bbf4b9c80
2. Ç. Çöltekin, “A set of open-source tools for Turkish natural language processing,” in Proc. LREC, 2014, pp. 1079–1086. [Online]. Available: https://coltekin.net/cagri/papers/coltekin2014lrec.pdf
3. K. W. Church and P. Hanks, “Word association norms, mutual information, and lexicography,” Computational Linguistics, vol. 16, no. 1, pp. 22–29, 1990.
4. S. Evert, The Statistics of Word Cooccurrences: Word Pairs and Collocations. Ph.D. diss., Univ. Stuttgart, 2004.
5. S. Evert and A. Hardie, “Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium” in Corpus Linguistics 2011, 2011.

A. Hardie, “CQPweb—Combining power, flexibility and usability in a corpus analysis tool,” Int. J. Corpus Linguistics, 17(3), 380–409, 2012, doi: 10.1075/ijcl.17.3.04har.

6. Kilgarriff et al., “The Sketch Engine: Ten years on,” Lexicography, 1, 7–36, 2014.
7. J. Nivre, M.-C. de Marneffe, F. Ginter, et al., “Universal Dependencies v2: An evergrowing multilingual treebank collection,” in Proc. LREC 2020, pp. 4027–4036.

8. P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, “Stanza: A Python NLP toolkit for many human languages,” in Proc. ACL System Demos, 2020, pp. 101–108.
9. M. Straka and J. Straková, “UDPipe 2.0 prototype at CoNLL 2018 UD shared task,” in Proc. CoNLL 2018 Shared Task, 2018, pp. 197–207.
10. M. Straka, “Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe,” in Proc. CoNLL 2017 Shared Task, 2017, pp. 88–99.
11. TEI Consortium, Guidelines for Electronic Text Encoding and Interchange (P5), v4.9.0, 2025.
12. TEI Consortium, TEI P5 Guidelines (Full PDF), 2025.
13. Universal Dependencies, UD Guidelines v2 (Living document).
14. Universal Dependencies, “Executive summary of changes from v1 to v2,” 2016.
15. Universal Dependencies, “Enhanced dependencies in UD v2,” n.d.
16. Apache Lucene Project, “Apache Lucene 9.9.1 documentation—File formats,” 2024.
17. Apache Lucene Project, “Index file formats” (historical docs, 2.x), 2010.
A. Chilukuri and S. Akram, “Analyzing and improving the scalability of in-memory inverted indexes in Lucene,” in Proc. ISMM ’23, 2023, pp. 77–88, doi: 10.1145/3591195.3595272.
18. G. K. Zipf, Human Behavior and the Principle of Least Effort. Addison-Wesley, 1949.
A. Akhundjanova and L. Talamo, “Universal Dependencies Treebank for Uzbek,” in Proc. RESOURCEFUL 2025, 2025.
19. UniversalDependencies, “UD_Uzbek-UT (GitHub repository)” 2025.
20. O. T. Yıldız, O. Kuyrukçu, Ş. T. Özateş, and E. Yılmaz, “An open, extendible, and fast Turkish morphological analyzer,” in Proc. RANLP 2019.
21. Allaberdiev et al., “Parallel texts dataset for Uzbek–Kazakh machine translation” PeerJ Computer Science, 2024.
22. Сулейманов, Д. Ш., & Мухамедшин, Д. Р. (2018). Система корпус-менеджер: архитектура и модели корпусных данных. Программные продукты и системы, 31(4), 653-658.
23. Aksan, Y., Aksan, M., Koltuksuz, A., Sezer, T., Mersinli, Ü., Demirhan, U. U., ... & Kurtoglu, Ö. (2012, May). Construction of the Turkish National Corpus (TNC). In LREC, pp. 3223-3227.
24. Hardie, A. (2012). CQPweb—combining power, flexibility and usability in a corpus analysis tool. International journal of corpus linguistics, 17(3), 380-409.
25. https://uznatcorpara.uz/uz/Dictionary/Search?q=%5Cbkel&op=1&cs=1
26. https://uznatcorpara.uz/uz/Bigram/Search?q1=&p1=56&q2=&p2=168&op=1&cs=1