September, 2025

№ 003

Management and Future Technologies

SCIENTIFIC JOURNAL

journal.umft.uz

ISSN 3060-5008

Jurnal sohalari: menejment, dasturiy injiniring, sun'iy intellekt texnologiyalari, kompyuter injiniringi, infokommunikatsiya injiniringi, axborot xavfsizligi, raqamli iqtisodiyot, pedagogika va psixologiya, matematika va fizika

Jurnal O'zbekiston Respublikasi Prezidenti Administratsiyasi huzuridagi Axborot va ommaviy kommunikatsiyalar agentligidan 212842-raqamli guvohnoma bilan 26.01.2024 sanasida ro'yxatdan o'tkazilgan. Jurnal OAK Rayosatining 2025-yil 12-fevraldagi 367-son qarori bilan 05.00.00 – Texnika fanlari ixtisosliklari bo'yicha "Dissertatsiyalar asosiy ilmiy natijalarini chop etish tavsiya etilgan ilmiy nashrlar ro'yxati"ga kirgazilgan.

MUNDARIJA

TEXNIKA FANLARI

TEXNIKA FANLARI

O‘ZBEK MATNLARI UCHUN N-GRAM TIL MODELI

Botir Elov; Mastura Primova
Texnika fanlari falsafa doktori, dotsent. ToshDO’TAU
elov@navoiy-uni.uz

Annotatsiya: Ushbu maqolada, til modellari ya’ni, statistik usuli N-gram modeli keltirilgan. Til modellari bu tabiiy til (masalan, o‘zbek tili, ingliz tili) dagi so‘zlar ketma-ketligini ehtimolli tarzda ifodalovchi matematik yoki hisoblash modelidir. Ular berilgan kontekst asosida keyingi so‘zni bashorat qilish**,** matn yaratish, so‘zlar ehtimolini baholash yoki tilga xos statistik xususiyatlarni tahlil qilish uchun qo‘llaniladi. Quyida N-gram modelidagi silliqlash usullaridan foydalanilgan: Laplace silliqlash, Good-Turing diskontlash, Katz back-off, Interpolatsiya, Kneser_ney silliqlash. Maqolada, O’zbek matnlari uchun n-gram til modelini qurish bosqichlari: korpusni tayyorlash, N-gramlarni shakllantirish, silliqlashtirish, til modeli formatini tayyorlash, bashorat qilish mexanizmlari keltirilgan.

Keywords: NLP, Smoothing, Kneser–Ney, IRSTLM, KenLM, N-gram.

Kirish

N-gram va Markov cheklovi: Til modellashning eng sodda statistik usuli *N-gram modelidir*. N-gram – bu ketma-ket kelgan N ta belgidan (odatda so‘zlardan) tashkil topgan guruh bo‘lib, til modelida ushbu guruhning ehtimolini hisoblashga xizmat qiladi. Markovning faraziga ko‘ra, navbatdagi so‘zning paydo bo‘lish ehtimolini aniqlash uchun undan oldingi barcha kontekstni emas, balki faqat so‘nggi *(N-1)* ta so‘zni hisobga olish kifoya. Masalan, bigram (2-gram) modelda so‘z faqat undan oldingi bitta so‘zga bog‘liq deb faraz qilinadi. Umumiy holda, N-gram til modeli uchun N so‘zdan iborat gap ehtimoli quyidagicha yoziladi:

$$P(w_1, w_2 \dots w_N) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \dots P(w_n|w_{1:n-1}) = \prod_{k=1}^{N} P(w_k|w_{1:k-1})$$

Bu yerda w_i i-so‘zni, N gapdagi so‘zlar sonini ifodalaydi. $P(w_i|w_{i-n+1:i-1})$ esa w_i so‘zining undan oldingi (N-1) so‘zlar ketma-ketligidan keyin kelish ehtimoli. N-gram modelda bu shartli ehtimollar korpusdagi chastotalar asosida maksimum haqqoniylik bahosi (Maximum Likelihood Estimate) bilan aniqlanadi:

$$P(w_i|w_{i-n+1:i-1}) \approx \frac{C(w_{i-n+1}, \dots, w_{i-1}, w_i)}{C(w_{i-n+1}, \dots, w_{i-1})}$$

bu yerda $C(\cdot)$ operatori qavs ichidagi so‘zlar ketma-ketligi korpusda nechta marta uchrashini (chastotasini) hisoblaydi. Masalan, o‘zbek tilidagi matnda *“juda ajoyib”* bigrammasi uchun quyidagi formula ishlatiladi:

$$P(\text{“}juda\text{”} \mid \text{“}ajoyib\text{”}) = C(\text{“}juda\ ajoyib\text{”}) \,/\, C(\text{“}juda\text{”})$$

Agar korpusda *“juda ajoyib”* ketma-ketligi biror marta ham uchramagan bo‘lsa, C(“juda ajoyib”) = 0 bo‘ladi va model bu kontekst uchun ehtimolni nolga teng deb baholaydi. Bu esa til modelida nol ehtimollar muammosini keltirib chiqaradi – ya’ni korpusda uchramagan har qanday so‘z ketma-ketligi model nazarida mumkin emas (ehtimoli 0).

Adabiyotlar sharhi

Hozirgi vaqtda jahon miqyosida til korpuslari va til modellari bo‘yicha keng qamrovli tadqiqotlar amalga oshirilgan. Til hodisalarini ehtimolliy model sifatida tahlil qilish g‘oyasi ilk bor Claude Shannon [1] tomonidan axborot nazariyasi doirasida ilgari surilgan bo‘lib, bu nazariy yondashuv keyingi tadqiqotlar uchun tamal toshi bo‘lib xizmat qildi. Masalan, Michel va boshqalar [2] Google Books N-gram korpusidagi 1500–2000-yillar oralig‘idagi millionlab kitoblar matnini tahlil qildi. Ushbu yondashuv natijasida asrlar davomida til va madaniyatdagi o‘zgarishlarni statistik usulda kuzatish mumkinligi ko‘rsatib berildi. O‘zbek tilining o‘ziga xos agglutinativ tuzilishi, ya’ni so‘z yasovchi qo‘shimchalarga boyligi, mavjud N-gram modellashtirish usullarini qo‘llaganda ba’zi muammolarni keltirib chiqaradi. Shuningdek, o‘zbek tilida dastlabki til modellarini yaratish bo‘yicha ayrim ilmiy maqolalar paydo bo‘la boshladi. Xususan, B. Elov va hammualliflar tomonidan o‘zbek tilida N-gram model qurish va uni baholashga doir izlanishlar 2024 yilda e’lon qilindi. Bu ishlarda korpus matnlari asosida unigram, bigram, trigram modellar tuzilib, ularning o‘rtacha log ehtimolliklari (perplexity) va boshqa ko‘rsatkichlari tahlil qilingan.

N-gram modellarda maxsus *silliqlash* usullari

Smoothing (silliqlash) usullari

Nol ehtimollar va kam kuzatilgan holatlarning ehtimollarini yaxlitlash uchun N-gram modellarda maxsus *silliqlash* usullari qo‘llaniladi. Silliqlashning maqsadi – korpusda kuzatilmagan ketma-ketliklarga kichik, musbat ehtimollar berish va kuzatilgan ketma-ketliklarning ehtimollarini biroz kamaytirib, taqsimotni muvozanatlashdir. Quyida N-gram modelidagi silliqlash usullari keltirlgan.

- Laplace (add-one) silliqlash: Har bir chastotaga 1 qo‘shiladi.
- Good–Turing diskontlash: Kichik chastotalarni diskontlab, ko‘rilmagan ketma-ketliklarga ehtimol beradi.
- Katz back-off: Kuzatilmagan yuqori tartibli ketma-ketliklar uchun past tartibli modellar ehtimoli ishlatiladi.
- Interpolatsiya: Har bir kontekst uchun turli tartibli modellarning ehtimollari λ og‘irliklar bilan birlashtiriladi.
- Kneser–Ney silliqlashi: Eng samarali usullardan biri bo‘lib, so‘zlarning yangi kontekstda uchrash imkonini baholaydi.

Eng sodda usul – Laplas (add-one) silliqlash bo‘lib, har bir chastotaga 1 qo‘shish orqali nol chastotalarni bartaraf etadi. Biroq, Laplas usuli katta so‘z boyligida ehtimollarni sun’iy oshirib yuborishi mumkin. Shu sabab, amaliyotda quyidagi samaraliroq silliqlash usullari qo‘llaniladi:

1. Good–Turing diskontlash usuli: statistikada kichik chastotalarni tuzatish uchun ishlab chiqilgan. Uning mohiyati – korpusda 0 marta uchragan ketma-ketliklarga ba’zi ehtimollarni taqsimlash uchun 1, 2, 3 va hokazo marta uchragan ketma-ketliklarning hisoblangan ehtimollarini biroz kamaytirib (diskontlash) qayta taqsimlashdir. Good–Turing formulasi ko‘rilmagan N-gramlarning ehtimolini ularning kutilayotgan chastotasi orqali baholaydi [3]. Bu yondashuv Katz tomonidan til modellashda tatbiq qilingan [4].

2. Katz back-off (chekinish) modeli: bu yondashuvda agar yuqori tartibli N-gram ketma-ketlik kuzatilmagan bo‘lsa, u holda model avtomatik ravishda bir pog‘ona pastroq (masalan, trigram kuzatilmasa – bigramga) murojaat qiladi va shu kontekst uchun mos ehtimolni oladi. Shu bilan birga, kuzatilgan ketma-ketliklar ehtimoli diskontlanib, ortgan massa pastroq modelga *“chekinishda”* ishlatiladi. Katz silliqlashi Good–Turing baholovini diskont koeffitsientlarini hisoblashda qo‘llaydi.

3. Interpolatsiya: bu usulda har bir mumkin bo‘lgan kontekst uchun yuqori va past tartibli modellarning ehtimollari muayyan λ og‘irliklar bilan aralashtiriladi. Masalan, uch so‘zli kontekst uchun:

- trigram modeli ehtimoli P_3,
- bigram modeli ehtimoli P_2,
- unigram modeli ehtimoli P_1 bo‘lsa, yakuni baho quyidahicha hisoblanadi:

$$\lambda_3 \cdot P_3 + \lambda_2 \cdot P_2 + \lambda_1 \cdot P_1$$

Bu yerda λ_i koeffitsientlar quyidagi shartga javob beradi:

$$\lambda_3 + \lambda_2 + \lambda_1 = 1$$

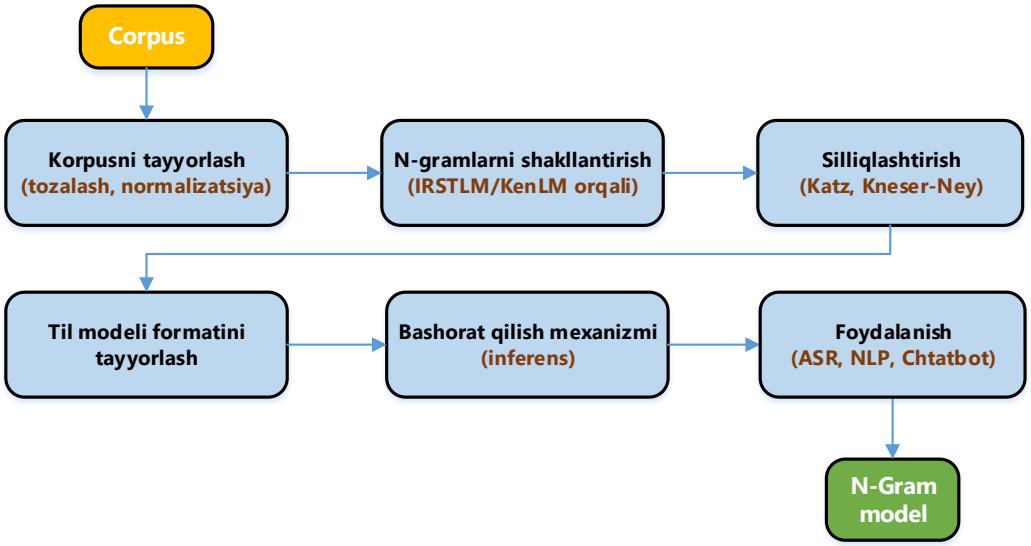
Interpolatsiya nol ehtimollar muammosini yumshatadi va turli tartibli modellarning afzalliklarini birlashtiradi.

4. Kneser–Ney silliqlashi: bugungi kunda eng samarali silliqlashlardan biri hisoblanadi. Kneser–Ney usuli har bir kuzatilgan N-gram chastotasidan ma’lum bir d diskont miqdorini ayirib (odatda 0<d<1 tanlanadi), ortgan ehtimollar massasini past tartibli modelga uzatadi. Eng muhim jihati – past tartibli modelda so‘zning *“yangi kontekstlarda uchrash ehtimoli”*ni hisobga oluvchi maxsus davomiylik ehtimoli (continuation probability) ishlatiladi. Bu esa kam uchragan so‘zlar uchun yaxshiroq baho beradi. Kneser–Ney algoritmi ko‘p hollarda til modellash masalalarida, xususan nutq tanish (ASR) tizimlarida, eng yaxshi natijalar bergani ma’lum.

N-gram modeli algoritmik yechimlari oddiy so‘z sanagich (counter) va ehtimollarni hisoblash formulalariga tayansa-da, uni samarali qurish va qo‘llash uchun texnologik asoslar talab etiladi. Ushbu maqolada ishlab chiqilgan N-gram til modelining texnik yechimlari va qurilish bosqichlari batafsil yoritiladi; xususan, *IRSTLM* va *KenLM* kabi maxsus toolkit’lar hamda Python dasturlari yordamida model yaratish jarayoni va natijalari tahlil qilinadi.

N-GRAM MODELINI QURISH BOSQICHLARI (IRSTLM, KENLM)

N-gram til modelini yaratish bir necha bosqichlar orqali amalga oshiriladi. Quyida IRSTLM, KenLM toolkit’lari va maxsus Python skriptlari yordamida bajarilgan asosiy bosqichlar ketma-ketligi keltirildi:



1-rasm. O‘zbek matnlari uchun n-gram til modelini qurish bosqichlari

1. Ma’lumotlarni oldindan tayyorlash (preprocessing). Modelni o‘qitish uchun avvalo katta hajmdagi matn korpusi yig‘ildi va tozalandi. Bu bosqichda matnlardagi keraksiz elementlar olib

tashlandi va matn normallashtirildi. Masalan, *HTML teglari, noto‘g‘ri kodlashdagi belgilar, ortiqcha probellar* va *tinish belgilari* tozalanadi, *barcha harflar kichik registrga keltiriladi*. Gap chegaralarini modellash uchun har bir gap boshiga maxsus <s> belgisi va oxiriga </s> belgisi qo‘shildi. Shuningdek, korpusda kam uchragan yoki nostandart so‘zlar (masalan, raqamlar ketma-ketligi, juda kam takrorlangan nomlar) umumlashtirib <unk> (unknown) belgisi bilan almashtirildi [5]. Bu usul *korpus lug‘atini qisqartirish* va *kam uchraydigan so‘zlarning model sifatiga salbiy ta’sirini kamaytirish* uchun qo‘llanadi. Natijada, tozalangan va normallashtirilgan matnlar tayyor bo‘lib, ulardan modelni o‘qitishda foydalanildi.

2. N-gramlarni hisoblash va lug‘at shakllantirish. Ma’lumotlarni oldindan tayyorlash bosqichidan o‘tgan matn korpusidan *N-gramlar statistikasini chiqarish* – model qurilishining asosiy texnik bosqichi. Bu bosqichda 1-gram (unigram) dan boshlab N-gramgacha bo‘lgan barcha ketma-ketliklar uchun chastotalar (kuzatilish soni) hisoblab chiqildi. Korpus hajmi katta bo‘lgani tufayli, barcha mumkin bo‘lgan n-gramlar soni ham juda katta bo‘ladi. Masalan, korpusdagi unikal so‘zlar lug‘ati o‘lchami V bo‘lsa, nazariy jihatdan N=3 (trigram) modell uchun eng ko‘pi bilan V^3 ta uchlik kombinatsiya bo‘lishi mumkin – bu juda ulkan miqdor. Amalda esa kuzatilgan n-gramlar soni ancha kichik bo‘ladi, lekin baribir millionlab sondagi n-gramlar darajasiga yetishi mumkin. Shuning uchun, bunday hisob-kitobni amalga oshirishda maxsus samarali dasturiy vositalar qo‘llanildi: IRSTLM va KenLM.

IRSTLM toolkit’i yirik korpuslardan til modelini qurishda resurslarni tejashga mo‘ljallangan bo‘lib, u o‘zining build-lm.sh skripti orqali korpusni avtomatik ravishda bir necha qismlarga bo‘lib hisoblash imkonini beradi. IRSTLM skripti matnni k=10 kabi bo‘laklarga ajratib, har birida n-gramlarni alohida sanab chiqadi va yakunda ularni birlashtiradi – bu usul hisoblash vaqtini biroz oshirsada, xotira bandligini sezilarli kamaytiradi. Misol uchun, IRSTLM da 3-gram model tuzish quyidagicha buyruq bilan amalga oshirilishi mumkin:

build-lm.sh -i "corpus.txt" -n 3 -o model.arpabo -k 10

(bu yerda -k 10 korpusni 10 qismga bo‘lishni belgilaydi).

KenLM toolkit’i esa buning aksi, maksimal tezlik va samaradorlikka yo‘naltirilgan: KenLM’da lmplz dasturi korpusni kiritma sifatida qabul qilib, ichki on-disk sortlash algoritmi yordamida n-gramlarni sanaydi. KenLM’da foydalanuvchi maksimal foydalaniladigan xotira hajmini (% ga yoki baytlar bilan) belgilashi mumkin, masalan

lmplz -o 5 -S 80% -T /tmp < corpus.txt > model.arpa

buyrug‘i korpusdan 5-gram modelni chiqaradi (80% operativ xotiragacha foydalanib, yetmasa vaqtinchalik /tmp diskiga yozadi). KenLM vositasi shu tariqa SRILM kabi boshqa vositalar eplay olmagan juda katta korpuslarni ham to‘liq qamrab oluvchi n-gram modelini sanashga qodir va buni amalga oshirishda IRSTLMdagidek taxminiy soddalashtirishlarga murojaat qilmaydi.

N-gramlarni hisoblash yakunida maxsus matn formatidagi ARPA fayl (Arpa Language Model format) hosil qilinadi – unda har bir n-gram uchun uning ehtimoli yoki keltirilgan holda log-ehtimoli hamda kerak bo‘lsa back-off vazn (izohlanadi) qiymatlari saqlanadi. Korpusdagi umumiy lug‘at (vocabulary) ham ushbu bosqichda aniqlanadi – masalan, bizning tajribamizda lug‘at kattaligi {V} ta so‘zni tashkil etdi (kam chastotali so‘zlarni <unk> ga birlashtirish natijasida). Keyingi bosqichlarda barcha hisob-kitoblar ana shu ARPA ko‘rinishdagi model asosida olib borildi.

3. Silliqlashtirish va ehtimollarni baholash. N-gramlar asosida tuzilgan cheksiz ko‘p ehtimoliy model parametrlari $P(w_i|w_{i-N+1\dots i-1})$ni korpusdagi chastotalardan to‘g‘ridan-to‘g‘ri baholash (maximum likelihood) yaxshi natija bermaydi, chunki ko‘plab n-gramlar *umuman uchramagan*

bo‘lishi tabiiy. Korpusda kuzatilmagan ketma-ketliklar uchun oddiy hisob-kitob ehtimolni nol deb baholaydi, bu esa til modeli nuqtai nazaridan muammo – nol ehtimolli hodisalar real til uchun mumkin emas (yoki juda kam ehtimolli xolos).

Shu bois, *silliqlashtirish (smoothing)* usullaridan foydalanildi. Silliqlashtirishning maqsadi – kuzatilmagan n-gram kombinatsiyalariga ham ma’lum ehtimol massasi ajratish va kuzatilgan n-gramlarning hisoblangan ehtimollarini biroz pasaytirib *“tekislash”*dir. Adabiyotlarda Laplas (Add-one), Gud-Tyuring, Kleyzer-Ney (Kneser-Ney), Vitten-Bell (Witten-Bell) kabi ko‘plab silliqlashtirish usullari mavjud [6]. Bizning ishlanmada zamonaviy til modellashda eng samarali deb tan olingan Modifikatsiyalangan Kneser-Ney silliqlashtirish usuli tanlandi. KenLM dasturi aynan shu usulni qo‘llaydi va ehtimollarni hisoblashda *chetlanish (discount)* va *back-off* kombinatsiyalangan modelini quradi. IRSTLM dasturida esa sukut bo‘yicha yengilroq *Witten-Bell silliqlashtirishi* ishlatiladi va u katta ma’lumotlarda samarali bo‘lsa-da, Kneser-Ney usulidan biroz pastroq sifat berishi mumkin. Darhaqiqat, mustaqil tadqiqotlar natijasida 5-gram model uchun Kneser-Ney silliqlashtirishi Witten-Bellga nisbatan perpleksiya ko‘rsatkichini yaxshiroq kamaytirgani kuzatilgan . Shuning uchun, biz Kneser-Ney usulidan foydalandik; IRSTLM imkon bergan holatlarda uning *“improved Kneser-Ney”* yaqinlashgan versiyasidan ham foydalandik. Silliqlashtirish jarayonida back-off sxemasidan foydalanildi: agar *berilgan N-1 kontekst + so‘z* kombinatsiyasi korpusda mavjud bo‘lmasa (uning chastotasi 0 bo‘lsa), u holda model pastroq darajaga *“chekinadi”* – ya’ni, so‘nggi N-2 so‘z konteksti uchun (N-1)-gram ehtimolga murojaat qiladi. Bunda ehtimollar yig‘indisi birligicha qolishi uchun har bir kichik modelga maxsus chekinish vazni (back-off weight, λ) koeffitsienti kiritiladi. Masalan, bigram modeli uchun Kneser-Ney formulasini yozsak:

Agar $w_i - 1$ oldingi so‘z kontekstida w_i so‘zi kuzatilgan bo‘lsa (ya’ni, $c(w_{i-1,i}) > 0$):

$$P(w_i|w_{i-1}) = \frac{c(w_{i-1,i}) - d}{c(w_{i-1})}$$

Bu yerda, d – discount (chegirma) konstanta.

Agar $w_i - 1$ kontekstida w_i so‘zi umuman uchramagan bo‘lsa ($c(w_{i-1,i}) = 0$):

$$P(w_i|w_{i-1}) = \lambda(w_{i-1}) \cdot P(w_i)$$

ya’ni, bigram modeli kichrayib unigram model ehtimoliga bog‘lanadi (λ – back-off vazn, $P(w_i)$ esa unigram ehtimol) [5].

Yuqoridagi formulalar Kneser-Ney back-off silliqlashtirishining soddalashtirilgan ko‘rinishi bo‘lib, undan yuqori darajadagi N-gramlar uchun ham xuddi shunday rekursiv qo‘llaniladi. Silliqlashtirish natijasida ARPA til modeli faylida har bir n-gram uchun ehtimoliyat (yoki log10 ehtimol) va kerakli joylarda *back-off vaznlar yozib chiqildi*. E’tibor qilish lozimki, IRSTLM yaratuvchilari Kneser-Neyning to‘liq realizatsiyasini xotira tejamkorligi maqsadida biroz soddalashtirganlar (bu usulni *“modified shift-beta”* deb atashadi)[7], KenLM esa to‘liq Kneser-Neyni realizatsiya qilgani tufayli, ayni bir korpusda KenLM chiqargan ehtimollar IRSTLM’ga nisbatan biroz ishonchliroq bo‘lishi mumkin. Shu sababdan, biz yakuniy model qurishda KenLM’dan olingan ehtimollarni asos qilib oldik, lekin *ikki xil toolkit* yordamida qurilgan modellarni ham qiyoslab ko‘rdik (quyida batafsilroq keltiriladi).

4. Til modeli formatini tayyorlash. N-gramlar va ularning silliqlashtirilgan ehtimollari tayyor bo‘lgach, til modelini samarali saqlash va ishlatish masalasi turadi. Ko‘pgina LM tizimlari matn ko‘rinishidagi ARPA fayllarni bevosita yuklab ishlay oladi. Biroq ARPA fayllar hajmi katta bo‘ladi (ayniqsa, yuqori darajali n-gram ko‘p bo‘lsa) va undagi qatorma-qator ma’lumotni har safar qidirish

sekinlik keltirib chiqaradi. Shu bois, amaliyotda til modelini binar formatga kompilyatsiya qilish afzal sanaladi. IRSTLM tarkibida bu uchun maxsus compile-lm yordamchi dasturi mavjud bo‘lib, u ARPA formatdagi modelni ixcham .bin faylga aylantiradi. KenLM ham xuddi shunga o‘xshash build_binary dasturiga ega – u ARPA faylni binar formatga konvert qiladi. KenLM’ning muhim jihati shundaki, u binar formatni shakllantirishda turli ma’lumot tuzilmalarini tanlash imkonini beradi: trie (daraxt), probing (hash-jadval) yoki sorted (saralangan massiv) variantlardan birini.

Standart variant – probing (bu eng tezkor qidiruvni ta’minlaydi, lekin xotirani ko‘proq ishlatadi), trie esa bir oz sekinroq bo‘lsa-da, xotira tejamkor formatdir[1]. Bizning loyiha doirasida modelni integratsiya qilishda asosan KenLM’ning trie-binar formati qo‘llandi – bu holda LM fayl hajmi sezilarli darajada kichrayadi va uni operativ xotiraga to‘liq yuklash shart bo‘lmaydi. Misol uchun, KenLM trie formatidagi model resident xotira hajmi IRSTLM eng ixcham varianti hajmining atigi 58% ini tashkil etdi (ya’ni deyarli yarmiga teng), shu bilan birga so‘rovlarni qayta ishlash tezligi IRSTLM eng tez variantining 81%iga teng bo‘ldi, ya’ni deyarli tezlikdan yutqazmagan holda xotira samaradorligi oshgan. Bunday binar modelni IRSTLM bilan ham kvantlash orqali erishish mumkin: IRSTLM’da ehtimollarni 4 bayt float o‘rniga 1 bayt qilib saqlash rejimi mavjud bo‘lib, quantize-lm yordamida ARPA faylni kvantlashtirilgan formatga o‘tkazish mumkin, natijada model hajmi 4 baravar kichrayadi. Biz tajribada KenLM modelini trie ko‘rinishida qurib, undan foydalanishda mmap (lazy loading) rejimini ham sinab ko‘rdik[8]: bunda model fayli diskdan to‘g‘ridan-to‘g‘ri xotiraga xaritalanadi va kerakli bo‘limlari dinamik yuklab ishlatiladi. Memory mapping usuli juda katta modellarga ega tizimlarda foydalidir – bir nechta jarayon bir model faylidan foydalanganda ham diskdan bir marta yuklab, birlik manzillar maydonida baham ko‘rishi mumkin. Bizning model hajmi unchalik ulkan bo‘lmagani bois (taxminan bir necha yuz MB atrofida) aksar hollarda modelni to‘liq xotiraga yuklab ishlatdik, bu esa yuklanish vaqtini qisqartirdi va so‘rovlarni bajarish tezligini oshirdi.

5. Bashorat qilish mexanizmi (inferens). N-gram til modeli qurib bo‘lingach, uni test qilish va turli ilovalarda qo‘llash uchun kerakli funksionallik ishlab chiqildi. Til modelining ishlash prinsipi shundan iboratki, u berilgan kontekst (ya’ni, oldingi N-1 ta so‘z) uchun navbatdagi so‘zning ehtimollar taqsimotini qaytaradi. Bizning holatda, masalan, KenLM modeli uchun maxsus Python wrapper yozilib, berilgan matn satrlarini tezkor skoring (baho olish) funksiyasi yaratildi. Ushbu funksiya gapdagi har bir so‘z ketma-ketligi ehtimolini modeldan oladi va natijada butun gapning log-ehtimoli hamda perpleksiya qiymatini hisoblab beradi. Xuddi shu mexanizm asosida keyingi paragrafda tavsiflangan perpleksiya ko‘rsatkichlari olindi. Til modelidan bashorat qilish (next word prediction) maqsadida foydalanish uchun esa quyidagicha yondashuv qo‘llanildi: <s> boshi berilgan holda, modelning ehtimollar taqsimoti ichidan eng yuqori ehtimolli so‘z tanlab olindi – bu birinchi so‘z bashorati bo‘ldi. So‘ngra tanlangan so‘z kontekstga qo‘shilib, yana modelga so‘rov berildi va navbatdagi so‘z tanlandi, bu jarayon </s> (gap oxiri) tokeni chiqqunga qadar davom ettirildi. Natijada, model mustaqil ravishda gaplar generatsiya qila oladigan dasturiy vosita yaratildi. Bunda, albatta, har doim eng yuqori ehtimolli so‘zni tanlash deterministik tekst hosil qilishi (ko‘p takrorlar va bir xil strukturadagi gaplar) mumkin. Shu sabab, generativ rejimda ba’zan ehtimollar taqsimotidan tasodifiy namuna olish (masalan, Boltzmann sampling yoki top-k random sampling usullari) orqali so‘z tanlash mexanizmi ham sinovdan o‘tkazildi. Umuman olganda, N-gram modelining bashorat qilish mexanizmi oddiy daraxt ko‘rinishidagi qidiruv jarayoni bo‘lib, yuqori tartibli (masalan, 5-gram) kontekst topilmasa, pastga chekinish va oxir-oqibat unigram ehtimolga tayanish bilan amalga

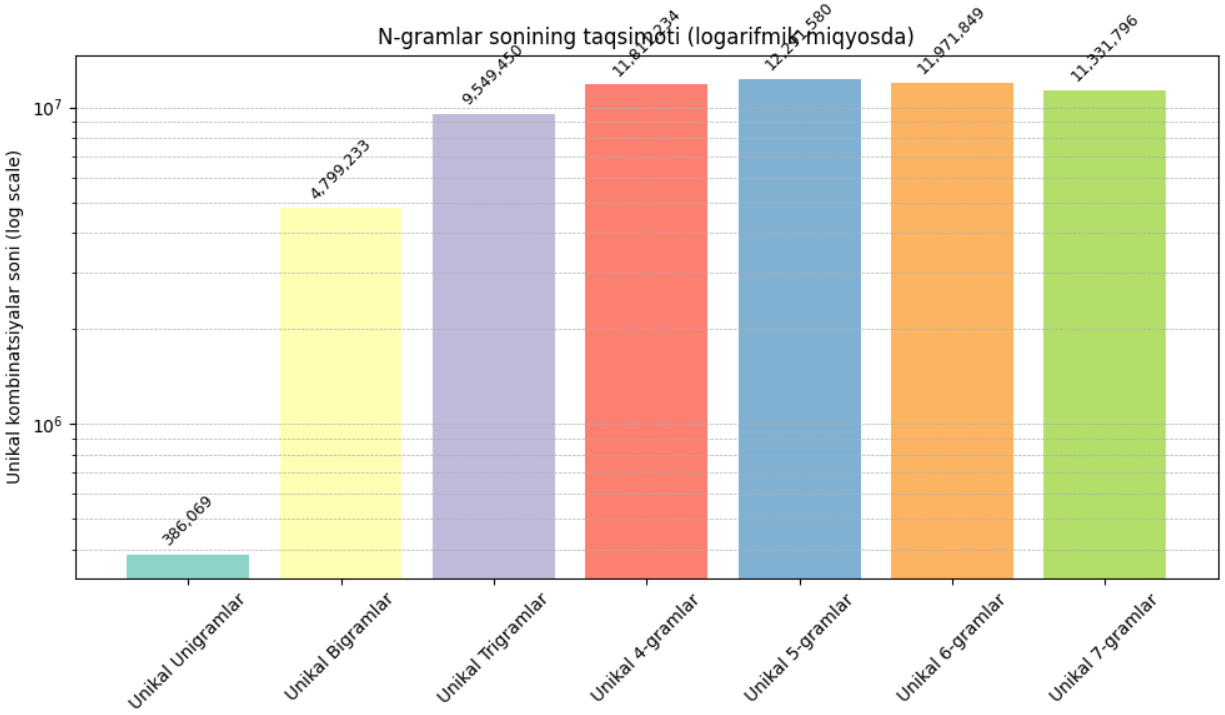
oshadi. Shu tariqa, qurilgan model yordamida turli kirish matnlariga ehtimoliy baho berish, keyingi so‘zni taxmin qilish, yoki so‘zlar ketma-ketligini to‘liq generatsiya qilish funksiyalari qo‘lga kiritildi.

Korpus chastotalari va kam uchraydigan holatlar

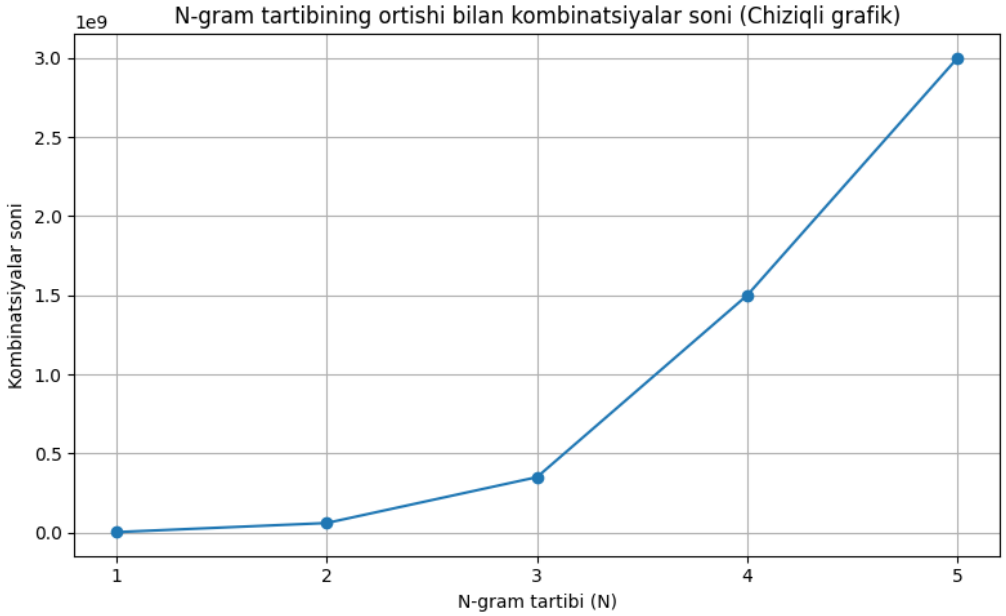
N-gram model sifatida qurilgan til modellari to‘liq statistik ma’lumotga tayanadi. Agar biror so‘z birikmasi katta korpusda ko‘p marotaba uchrasa, model uning ehtimolini yuqori deb belgilaydi, aksincha kam uchragan yoki uchramagan birikmalar ehtimoli juda kichik bo‘ladi. Tilning xususiyatlariga ko‘ra, hatto juda katta korpuslarda ham ma’lumotlar siyrakligi (data sparsity) muammosi yuzaga keladi – ya’ni mumkin bo‘lgan so‘z kombinatsiyalarining ulkan ko‘pligidan amalda korpusda ularning faqat kichik qismi kuzatilgan bo‘ladi. Masalan, o`zbek tilidagi veb-korpusdan (gaplar soni: 1,120,458) olingan statistikaga qarasak (1-jadval), 386.069 ta unikal so‘z (unigram) mavjud bo‘lganda bigramlar soni 4.799.233, trigramlar soni esa 9.549.450 ga teng.

1-jadval. N-gramlar sonining taqsimoti (logarifmik miqyosda)

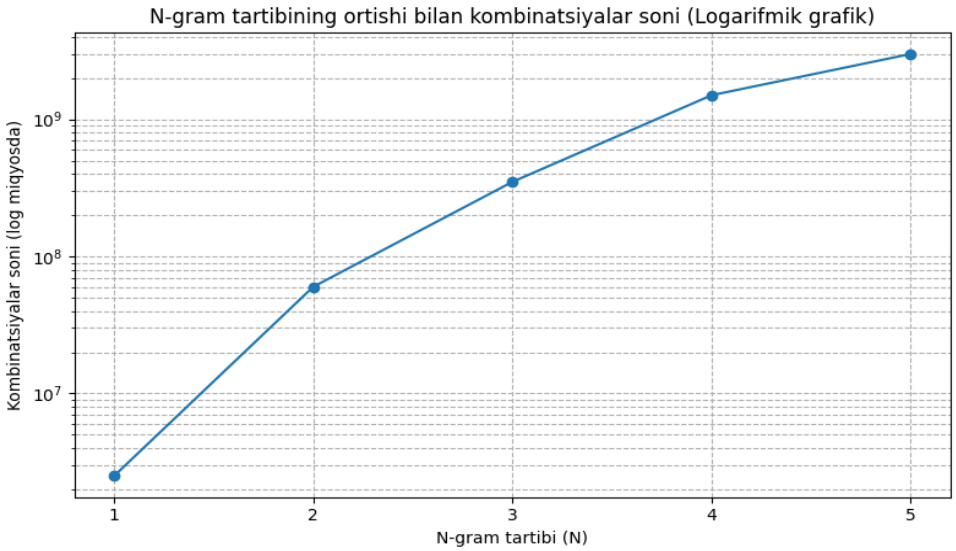
Gaplar soni	**1,120,458**
Tokenlar soni	19,984,549
Unikal Unigramlar	386,069
Unikal Bigramlar	4,799,233
Unikal Trigramlar	9,549,450
Unikal 4-gramlar	11,811,234
Unikal 5-gramlar	12,291,580
Unikal 6-gramlar	11,971,849



2-rasm. N-gramlar sonining taqsimoti (logarifmik miqyosda)



3-rasm. N-Gramlarning chiziqli grafigi



4-rasm. N-Gramlarning logarifmik grafigi

1) Chiziqli grafik — N-gram tartibi ortishi bilan kombinatsiyalar sonining qanday o‘sishini aniq ko‘rsatadi. 4-gramdan boshlab keskin eksploziv o‘sish kuzatiladi.

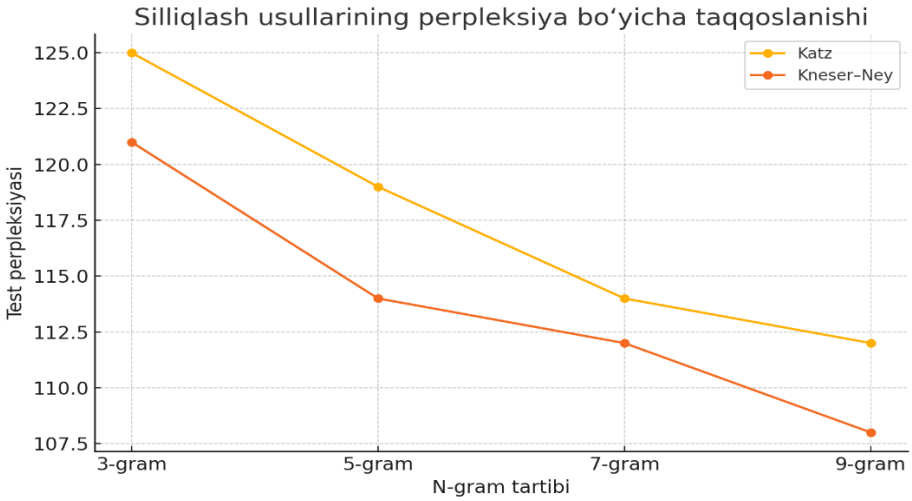
2) Logarifmik grafik — Eksponensial o‘sishni yanada ravshanroq ko‘rsatadi va yuqori tartibli N-gram modellarning resurs talabini vizual baholashga yordam beradi.

N-grammalarning kombinator portlashi

Korpus hajmi va til boyligi oshishi bilan yuqori tartibli N-gramlar soni keskin ortadi. Masalan, o‘zbek tilid milliy korpusida trigrammalar soni unigramlardan bir necha barobar ko‘p bo‘lib, 9.549.450 unikal trigramma kuzatilgan (1-jadvalda 7-grammagacha ko‘rsatilgan). Bu o‘zbek tiliga xos holat; aglutinativ tuzilishi va erkin so‘z tartibi tufayli o‘zbek tilida ham mumkin bo‘lgan so‘z ketma-ketliklari kombinatsiyasi behisob bo‘lib, ulardan amaliy kuzatilganlari nisbatan kam bo‘ladi. Demak, N qancha katta bo‘lsa, model parametrlarining soni (kuzatilgan N-grammalar soni) shuncha keskin oshadi. Shu sababli korpus hajmi cheklanganda odatda 3-gram, 4-gram kabi past tartibli

modellargacha foydalaniladi; undan yuqori tartiblar uchun ishonchli statistikani yig‘ish juda qiyinlashadi.

Kam chastotali so‘zlar va kontekstlar N-gram modellarda alohida muammo tug‘diradi. Korpusda kam kuzatilgan birikmalar uchun maksimum haqqoniylik ehtimoli past bo‘ladi, garchi real tilda bunday kombinatsiya mantiqan mumkin bo‘lsa ham. Silliqlash usullari aynan shunday kam uchraydigan holatlar ehtimolini 0 dan yuqoriroq qilib, modelning generallashuvi (umumlashma qobiliyati)ni oshiradi. Misol tariqasida, O‘zbek tili bo‘yicha tuzilgan 3-, 5-, 7- va 9-gram modellarning test to‘plamdagi perpleksiyasini (quyida tushuntiriladi) solishtiramiz (5-rasm). Kneser–Ney silliqlashi qo‘llanganda 3-gram model test perpleksiyasi 121, 5-gramda 114, 7-gramda 112, va 9-gramda 108 ni tashkil qilgan. Ya’ni, yuqori tartibli model aniqroq probabilistik bashorat bermoqda (perpleksiya pasaymoqda). Katz silliqlashi ham xuddi shunday tendensiyani ko‘rsatgan, ammo uning qiymatlari biroz balandroq (masalan, 9-gram Katz modeli 112 perpleksiya) va umuman Kneser–Ney interpolatsiyali yondashuvdan samaradorlikda ortda ekanligi kuzatilgan. Shunday qilib, statistik modellarda N oshishi bilan nazariy kontekst qamrovi kengaysa-da, amaliy cheklovlar tufayli silliqlash va resurs cheklovlari (xotira, korpus hajmi) inobatga olinadi.



5-rasm. Silliqlash usullarining perpleksiya bo‘yicha taqqoslanishi

Xulosa

Ushbu maqolada, O‘zbek tilida n-gram asosidagi til modelini qurish bosqichlari va texnik jihatlari ishlatilgan vositalar chuqur tahlil qilindi. Model yaratish jarayonida ma’lumotlarni *tozalash*, *n-gram chastotalarini hisoblash*, *ehtimollarni silliqlashtirish* hamda modelni samarali va resurs tejamkor tarzda qurish kabi muhim bosqichlarga e’tibor qaratildi. *IRSTLM* va *KenLM* kabi toolkit’lar bilan ishlash jarayonlari solishtirilib, ularning afzallik va kamchiliklari amaliy tajribalar orqali ko‘rsatib berildi. Xususan, IRSTLM o‘zining xotira tejamkorligi bilan ajralib tursa, KenLM maksimal samaradorlik va aniqlikni ta’minlaydi. Ayniqsa, KenLM tomonidan taqdim etilgan modifikatsiyalangan Kneser-Ney silliqlashtirish algoritmining yuqori sifatli natijalar berishi va katta korpuslarda yaxshi perpleksiya ko‘rsatkichlariga erishgani alohida qayd etildi. Shu tariqa, O‘zbek matnlariga asoslangan n-gram modelini samarali tarzda qurish mumkinligi isbotlandi. Bunday model kelgusida avtomatik tarjima, matn yaratish, matnni tahlil qilish, so‘z tavsiya qilish kabi tabiiy tilni qayta ishlashga oid bir qator muammolarni hal etadi.

Adabiyotlar

1. Claude Shannon “A Mathematical Theory of Communication”. 1948. United States
2. M. Davies. 2011. Google Books (American English) Corpus (155 billion words, 1810-2009). http://googlebooks.byu.edu/.
   Nadas, A. (1984). Estimation of probabilities in the language model of the IBM speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*(4), 859-861.
3. Church, K. W., & Gale, W. A. (1991). A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech & Language*, *5*(1), 19-54.
4. Mani, S., Gothe, S. V., Ghosh, S., Mishra, A. K., Kulshreshtha, P., Bhargavi, M., & Kumaran, M. (2019, January). Real-time optimized n-gram for mobile devices. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)* (pp. 87-92). IEEE.
5. R. Ismail. (2014, May). Comparison of modified kneser-ney and witten-bell smoothing techniques in statistical language model of bahasa Indonesia. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)* (pp. 409-412). IEEE.
6. Heafield, K., Pouzyrevsky, I., Clark, J. H., & Koehn, P. (2013, August). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 690-696).