O‘ZBEK TILIDAGI MATNLARIDAGI PEREFRAZ BIRLIKLARNI ANIQLASH

Axmedova Xusniya Xusanovna[1], Sultanov Djamshid Baxodirovich[2]

[1]Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiylari universiteti,
[2]Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiylari universiteti,
E-mail: sdjamshid@gmail.com

K E Y W O R D S

perefraz gaplar, semantic o‘xshashliklar, Jaccard algoritmi, Glove algoritmi, Mashinali o‘qitish

A B S T R A C T

O‘zbek tilidagi matnlari tarkibidagi perefraz juftliklarni aniqlash masalasining yechish ko‘plab NLP masalalarini yechishda qulayliklar yaratib beradi. Bu plagiat tizimlaridan farqli ravish bitta matn tarkibidagi duvlikatlarni aniqlaydi. Perefraz gaplarni aniqlashda ko‘plab ML algoritmlaridan foydalaniladi. Ushbu maqolada gaplarning o‘xshashligini aniqlashda Jaccard algoritmi, so‘zlarning semantic o‘xshashlik to‘plamini tuzishda Glove algoritmidan foydalanish ketma-ketliklari keltirilgan. Perefrazni aniqlash axborot tizimining foydalanuvchilari hamda unda foydalanishda kechadigan jarayonlar haqida ma’lumotlar keltirilgan.

Kirish

Perefrazlash jarayoni matnni qayta yozishni, so‘zlar va ularning tartibini o‘zgartirishni, shu bilan birga asl ma’noni saqlab qolishni o‘z ichiga oladi. Qayta yozishni aniqlash Tabiiy tilga ishlov berish (NLP) sohasidagi turli vazifalarda, jumladan, plagiatni aniqlash, mualliflikni aniqlash, savol-javob tizimlarida qo‘llash, matnni qisqartirish, mashina tarjimasi, umumiy matn tahlili va boshqalarda muhim rol o‘ynaydi. Matnlarning semantik o‘xshashlik darajasini aniqlash kabi umumiyroq vazifa NLP sohasida katta ahamiyatga ega.

Perefraz qilingan matn plagiatni aniqlash tizimlari duch keladigan eng qiyin muammolardan biridir. Ko‘pgina plagiatni aniqlash tizimlari odatda umumiy so‘zlar va kichik o‘zgarishlarni aniqlash uchun yaratilgan, ammo keng semantik tuzilma va jumladagi o‘zgarishlarni aniqlashga qodir emas. Shu sababli, ko‘plab qayta yozish holatlari sezilmay qoladi.

Adabiyotlar tahlili

Ko‘pgina plagiatni tekshirishga mo‘ljallangan axborot tizimlari gaplarning qismlari yoki umumiy so‘zlarni taqqoslashga asoslanadi, ammo perefrazlarni aniqlash tabiiy tilga ishlov berish sohasida hanuz dolzarb vazifa bo‘lib qolmoqda va u ko‘pchilik onlayn platformalar va ilovalarda amalga oshirilmagan. Bundan tashqari, mavjud plagiatni tekshirish tizimlari perefrazlarni aniqlash va asl manbani yetarli aniqlikda ko‘rsatib bera olmaydi [1].

Shunday qilib, perefrazlarni aniqlash dolzarb muammo bo‘lib, hozirgi kunda ko‘plab ilmiy ishlarda tadqiqotchilar chuqur o‘rganish usullari va millionlab parametrlarga ega bo‘lgan nutq modellarining arxitekturalaridan foydalanmoqda. Ammo ushbu muammoni hal qilish tarixi ikki atama o‘rtasidagi semantik farqni WordNet ma’lumotlar bazasida masofa sifatida hisoblashga asoslangan oddiyroq usullardan boshlanadi [2]. WordNet – so‘zlar (sinsetlar) o‘rtasidagi semantik bog‘lanishlarning ma’lumotlar bazasi bo‘lib, unda sinonimlar, giponimlar (subtiplar), meronimlar (qismlar) va boshqalar mavjud. Ma’lumotlar bazasining iyerarxik tuzilmasi turli formulalar yordamida alohida so‘zlarning semantik bog‘liqligini hisoblash imkonini beradi.

WordNet lug‘atidagi so‘zlarning semantik bog‘liqlik qiymatlariga asoslangan yondashuv taklif qilingan[3]. Ushbu usul so‘zlar juftligini so‘z turkumlariga qarab taqqoslash g‘oyasiga asoslanadi.

© X.X. Axmedova, Dj.B. Sultanov

Tadqiqotchilar WordNet asosida olti xil semantik o‘xshashlik masofalarini qo‘llaganlar [4-9]. Maksimal o‘xshashlik faqat bir xil so‘z turkumi bo‘lgan so‘zlar o‘rtasida qidiriladi. Ikki yo‘nalishlilikni amalga oshirish uchun taklif qilingan funksiyaning ikki qiymatining o‘rta arifmetigidan foydalaniladi, uning qiymati so‘zlar juftligining maksimal o‘xshashlik qiymati va ularning o‘ziga xosligiga bog‘liq. Shuningdek, barcha olti masofa semantik o‘xshashlikni yakuniy baholash ko‘rsatkichlarini o‘rtacha olish orqali birlashtirilishi mumkin.

Ba’zi tadqiqotlarda so‘zlarning semantik o‘xshashligi sinsetlar o‘rtasidagi eng qisqa yo‘l uzunligiga bog‘liq ekanligi muhoka qilingan [4]. Tabiiy tillarda so‘z o‘xshashligi berilgan terminlarga mos keluvchi so‘z ta’riflari to‘plamlarining kesishmasi sifatida aniqlanadi [5]. Boshqa ilmiy manbaga ko‘ra – katta korpusda umumiy ajdod bilan uchrashish ehtimolligiga, umumiy ajdod va taqqoslanayotgan ikki sinset bilan uchrashish ehtimolligiga bog‘liq [6-9].

Turli semantik o‘xshashlikni aniqlash va perefrazalarni aniqlash uchun turli metodologiyalardan foydalanilgan (umumiy n-gramlarning sonini hisoblashdan tortib, chuqur ML usullarini qo‘llashgacha), shuning uchun eng yuqori tasniflash aniqligini olish uchun, gaplarning semantik o‘xshashligini o‘lchashning turli algoritmlarini solishtirish va/yoki birlashtirish zarur [10].

Asosiy qism

O‘zbek tili matnlaridagi perefraz gaplarni aniqlashni amalga oshiruvchi tadqiqotda xususiyat sifatida quyidagi semantik o‘xshashlik ko‘rsatkichlari tanlangan:

- Umumiy n-gramlar uchun Jaccard koeffitsienti;
- Jumlalarning vektorli ko‘rinishlari orasidagi kosinus masofasi;
- Word Mover’s Distance [11];
- semantik o‘xshashlikni aniqlashda Glove algoritmi;
- Ikki ML modelining bashorati: takrorlanuvchi va RoBERTa asosidagi Siamese NN [12].

n-gram — bu n ta so‘zlar ketma-ketligi. Matnda perefrazalarni aniqlash kontekstida umumiy n-gramlarning soni, har ikkala jumladagi umumiy n-gramlar soniga nisbatan normallashtirilgan holda, semantik jihatdan yaqin bo‘lgan, ammo perefraz bo‘lmagan semantik jihatdan o‘xshash jumlalarni aniqlashga yordam beradi. Bu ikkinchi jumla birinchi jumladagi so‘zlarning o‘rnini almashtirish orqali olinganligi sababli butunlay boshqa ma’noga ega bo‘ladi. O‘xshashlikni aniqlashda turli algoritmlardan foydalanish mumkin. Cosinus o‘xshashlik, Jaccard o‘xshashlik, TF-IDF, Glove, Wordnetga asoslangan sinset to‘plamlaridan foydalanish kabi algoritmlar shular jumlasidandir. Quyida ushbu algoritmlardan foydalanish ketma-ketligini ko‘rib chiqamiz.

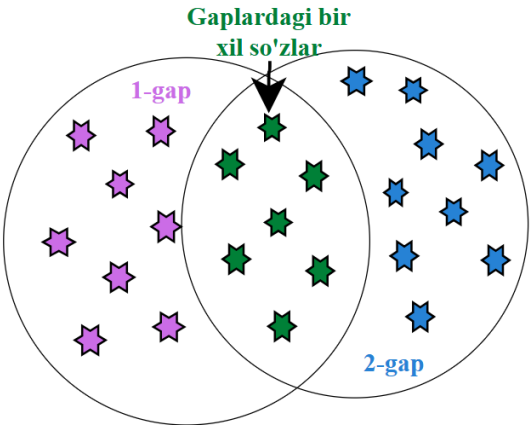
Jaccard algoritmi. Ikki to‘plam uchun Jaccard koeffitsienti quyidagicha hisoblanadi:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (1)$$

Bu yerda,

$|A \cap B|$–A va B to‘plamlarining umumiy elementlari soni.

$|A \cup B|$- A va B to‘plamlaridagi umumiy elementlar soni



1-rasm: Jaccard o‘xshashlik

Har bir jumladagi n-gramlarni hisoblash uchun jumlalar avval kichik harflarga o‘tkaziladi, barcha ajratgichlar va qo‘shimcha belgilar olib tashlanadi. 2-, 3- va 4-gramlar uchun umumiy Jaccard koeffitsientlari hisoblandi. Quyida ikkita gapning semantik o‘xshashlik koeffitsiyenti hisoblashni ko‘rib chiqamiz.

1. *Navbatdagi yig‘ilish keyingi juma kuni bo‘lib o‘tadi*

(cc) BY © X.X. Axmedova, Dj.B. Sultanov

2. *Keyingi majlisni kelasi hafta jumada o‘tkazamiz.*

Ushbu ikkita gapda semantik o‘xshash so‘zlar quyidagilar:

Navbatdagi->keyingi-> kelasi, yig‘ilish-> majlis, juma kuni->jumada, bo‘lib o‘tadi->o‘tkazamiz

Keltirilgan semantik o‘xshash so‘zlarga asoslanib Jaccard koeffitsiyentini hisoblaymiz.

A={n*avbatdagi, yig‘ilish, keyingi, juma, kuni, bo‘lib, o‘tadi* }

B={*keyingi, majlisni, kelasi, hafta, jumada, o‘tkazamiz*}

$$A \cap B = \{keyingi, yig'ilish, kelasi, juma, o'tadi\}$$

$$A \cup B = \{keyingi, yig'ilish, hafta, juma, kuni, bo'lib\ o'tadi\}$$

Hosil bo‘lgan to‘plamlarning modullari asosida Jaccard koeffitsiyenti hisoblanadi.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{5}{7} = 0.714 = 71.4\%$$

Jaccard formulasi asosan so‘zlarga asoslanadi, ammo shuni alohida ta’kidlash kerakki, o‘zbek tilida so‘z birikmasining so‘z bilan sinonimlik holatlari ham uchraydi. Bunday hollarda bu usuldan foydalanish samarali natija bermasligi mumkin. Masalan yuqoridagi (A) va (B) gaplardagi *(juma kuni va jumada)* hamda (*bo‘lib o‘tadi va o‘tkaziladi*) juftliklari ham kontekstda sinonim bo‘la oladi. Ushbu muammoni bartaraf etishda boshqa algoritmlardan foydalanish maqsadga muvofiq. Ya’ni so‘zlarning vektorli ko‘rinishlarini olish usullariga murojaat qilish kerak. Albatta so‘zlarning sinonomik qatorlarni ajratishda so‘z va so‘z brikmalarini ham inobatga olish kerak. So‘z yoki jumlalar o‘rtasidagi semantik o‘xshashlikni BERT yordamida hisoblash ham mumkin. Transformer modellaridan, masalan BERT, so‘z yoki jumlalarning semantik o‘xshashligini hisoblash uchun embedding vektorlar olinadi va ular orasidagi masofa yoki o‘xshashlik o‘lchanadi. Asosiy usullar quyidagilardan iborat:

1. So‘zlar o‘rtasidagi o‘xshashlik (token embeddinglari orqali)
2. Jumlalar o‘rtasidagi o‘xshashlik (sentence embeddinglar orqali)

Semantik o‘xshashlik qiymatlarini baholashda Bert algoritmidan foydalanish zamonaviy qarashlardandir. Oldindan o‘qitilgan modeldan foydalanib, har bir jumla uchun vektorli ko‘rinish matritsasi quyidagi o‘lchamlarga ega: torch.Size ([1, 20, 50]), bu yerda 1 – bata bo‘yicha jumlalar soni, 20 – jumlaning maksimal uzunligi va 50 – vektorli joylashtirish o‘lchami. Jumlaning vektorli ko‘rinishini olish uchun ma’lumotlar jumladagi har bir so‘z vektorining o‘rtacha qiymat hisoblanadi, natijada uzunligi 50 qiymatdan iborat bo‘lgan joylashtirilgan vektor hosil bo‘ladi.

$$C = \frac{(\sum A_i B_i)}{((\sum A_i^2)^{\frac{1}{2}} (\sum B_i^2)^{\frac{1}{2}})} \quad (2)$$

Ushbu funksiya semantik jihatdan o‘xshash jumlalarni aniqlashi mumkin, ammo aynan bir xil so‘zlardan iborat bo‘lgan holatdagina, biroq sinonim bo‘lmagan jumlalar kichik kosinus masofasiga ega bo‘ladi. So‘zlar o‘rtasidagi masofani o‘lchash uchun Word Mover Distance (WMD) vektorlarning joylashuviga asoslangan holda jumlalar orasidagi semantik masofani hisoblaydi. WMD masofasi ikki matn hujjatlari orasidagi farqni o‘lchab, bir hujjatning so‘z vektorlarini boshqa hujjatning so‘z vektorlari joylashuviga yetkazish uchun zarur bo‘lgan minimal "o‘zgartirish masofasi"ni aniqlaydi. Avvalgi funksiyaga o‘xshab, bunday masofa metrikasi semantik jihatdan o‘xshash so‘zlar juftliklarini aniqlashga yordam beradi, lekin so‘zlarning tartibini o‘zgartirish orqali hosil qilingan, sinonim bo‘lmagan jumlalarni aniqlashda yordam bermaydi.

So‘zlarning vektorli ko‘rinishlarini olishning turli usullari mavjud: *GloVe, Word2Vec: CBOW / Skip-Gram, fastText.*

Glove algoritmi. Glove algoritmi yordamida so‘z vektorlarini hosil qilish jarayonini ko‘rib chiqamiz. GloVe (Global Vectors for Word Representation) — bu so‘zlarni global statistikalar asosida vektorlar orqali ifodalashga asoslangan

© X.X. Axmedova, Dj.B. Sultanov

word embedding algoritmi. Word2Vecdan farqli o'laroq, GloVe butun matnning statistik xususiyatlarini hisobga oladi va bu uni ko'proq global semantik bog'liqliklarni o'rganish uchun samarali ekanligini anglatadi. GloVe modeli katta datasetlarda yaxshi natija beradi va so'zlar orasidagi aloqalarni aniqroq ko'rsatishda samaralidir. GloVe algoritmining ketma-ketligi quyidagicha amalga oshiriladi va algoritm matndagi so'zlarni vektorlar bilan ifodalashda global statistikalarni hisobga oladi. Quyidagi jarayonlar GloVe modelining ishlash ketma-ketligini tashkil etadi:

–*Matnni tahlil qilish va so'zlar orasidagi ko'plab juftliklarni yaratish:* Dastlab, matnni o'rganish uchun kontekstli so'z juftliklarini yaratish kerak. Bu jarayon davomida, har bir so'z uchun, uning atrofidagi so'zlarni (konteksti) aniqlash va ularni juftliklar sifatida yig'ish talab etiladi.

–*So'zlar o'rtasidagi ko'pgina juftliklarning ehtimolliklarini hisoblash:* Har bir so'z juftligi uchun birgalikda uchrashish ehtimolligi (probability) hisoblanadi. Bu ehtimolliklar so'zlarning birgalikda necha marta uchraganiga asoslanadi. Agar so'zlar yaqinroq, ya'ni, ular bir-biri bilan ko'proq aloqada bo'lsa, ularning ehtimolligi yuqori bo'ladi.

–*So'z vektorlarini tasodifiy boshlang'ich qiymatlarda o'rnatish:* Har bir so'z uchun tasodifiy vektor yaratish kerak. Bu vektorlar dastlab biror qiymatga ega bo'ladi, lekin keyinchalik algoritm yordamida optimallashtiriladi va ular har bir so'zning semantik ma'nosini aks ettiradi.

–*Optimallashtirish funktsiyasini yaratish.* Ushbu modelda so'zlar orasidagi ehtimolliklarni hisoblash uchun optimallashtirish funksiyasi ishlatiladi.

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(P_{ij}) \cdot (w_i^T \cdot \tilde{w}_j + b_i + \check{b}_j - logP_{ij})^2 \quad (3)$$

Bu yerda,

$w_i^T, \tilde{w}_j$– so'zlarning vektor ifodalari,

P_{ij}–i- va j- so'zlar orasidagi yaqinlik ehtimolligi,

$f(P_{ij})$–ehtimollikning logarifmik funksiyasi

–*Funksiyani minimallashtirish.* Optimallashtirish funksiyasini minimallashtirish uchun gradient descent ishlatiladi. Bu jarayon so'zlarning vectorlarini optimallashtiradi. Gradiyentni hisoblash orqali so'z vektorlari va ularning semantik yaqinligi aniqlanadi.

–*Bias qiymatlarni kiritish.* Har bir so'z uchun bias parametrlarni qo'shish modelning samaradorligini oshirishi mumkin. Bu parametrlar so'zlarning individual xususiyatlarini hisobga oladi.

Python dasturlash tilida Glove algoritmidan foydalanish mumkin. Buning uchun *gensim* kutubxonasini o'rnatish lozim. Glove algoritmidan foydalangan holda target so'zning semantic o'xshashlarini aniqlash uchun albatta yetarlicha dataset jamlanishi kerak. Jamlangan dataset ustida matnlarni dastlabki ishlov berish jarayonlari amalga oshirilishi va undan keyin har bir so'zning semantic vektori hosil qilinadi. Quyida Glove algoritmidan foydalanish python kodini keltiramiz:

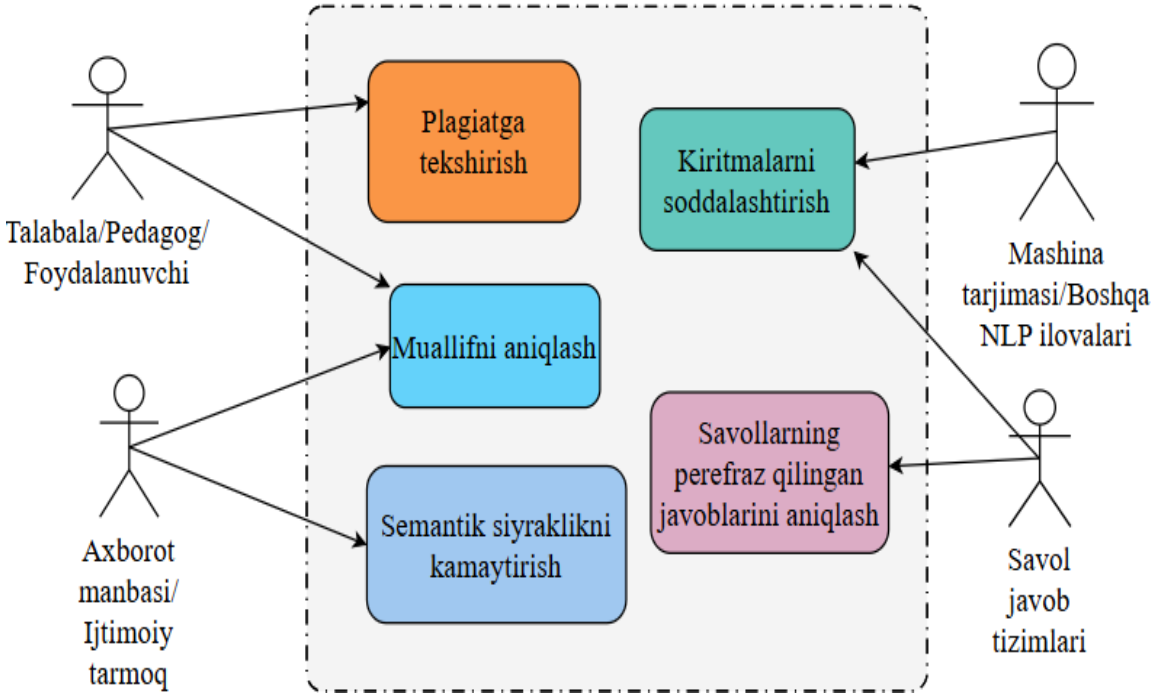
```
def loadGlove():
        import gensim.downloader as api
        import numpy as np
         glove_model = api.load("glove-wiki-gigaword-50")
        file_path = "matn.txt"
            with open(file_path, "r", encoding="utf-8") as f:
              text = f.read()
        words = text.lower().split()
        vectors = [glove_model[word] for word in words if word in glove_model]
        if vectors:
           text_vector = np.mean(vectors, axis=0)
          print("Matnning umumiy GloVe vektori:", text_vector)
        else:
           print("Matn tarkibidagi so'zlar GloVe modelida topilmadi.")
```

Ushbu python kod natijasida *.txt* faylda keltirilgan matndagi har bir so'zning semantic vectori hosil qilinadi. So'z va uning semantic vectoridan iborat fayl ustida target so'zning har bir sinonimining o'xshashlik koeffitsiyenti

hisoblanadi. Aniqlangan koeffitsiyentlarni baholash asosida har bir so‘zning semantic o‘xshashliklaridan iborat to‘plami hosil qilinadi. Gapning har bir n-gramining semantic o‘xshashlik to‘plamining elementlari asosida sinonim gaplarni aniqlash mumkin.

Perefrazlashni aniqlashga oid axborot tizimi axborot resurslari va ijtimoiy tarmoqlar moderatorlari uchun mo‘ljallangan [12-18]. Bundan tashqari, perefrazlashni aniqlash algoritmi mavjud plagiat tekshirish tizimlariga qo‘shilishi mumkin. Axborot tizimi (AT)ning akademik ishlarning o‘ziga xosligini tekshirishdagi talablar tavsifining misolini keltirish muhimdir. Axborot tizimining ishlashi ilmiy rahbar, matn muallifi, matnni tekshiruvchi shaxs va dasturiy ta’minot tomonidan ta’minlanadi. O‘qituvchi muallifning ishini yozishning barcha bosqichlarini tekshiradi va avtomatik ravishda originalni tekshirishdan oldin TPPni o‘tkazadi. Muallif tekshiruvchi bilan faol muloqot qiladi, uning tanqidlari va izohlariga javob beradi, ishni uning fikrlariga asosan tuzatadi va keyin uni ko‘rib chiqish uchun yuboradi.

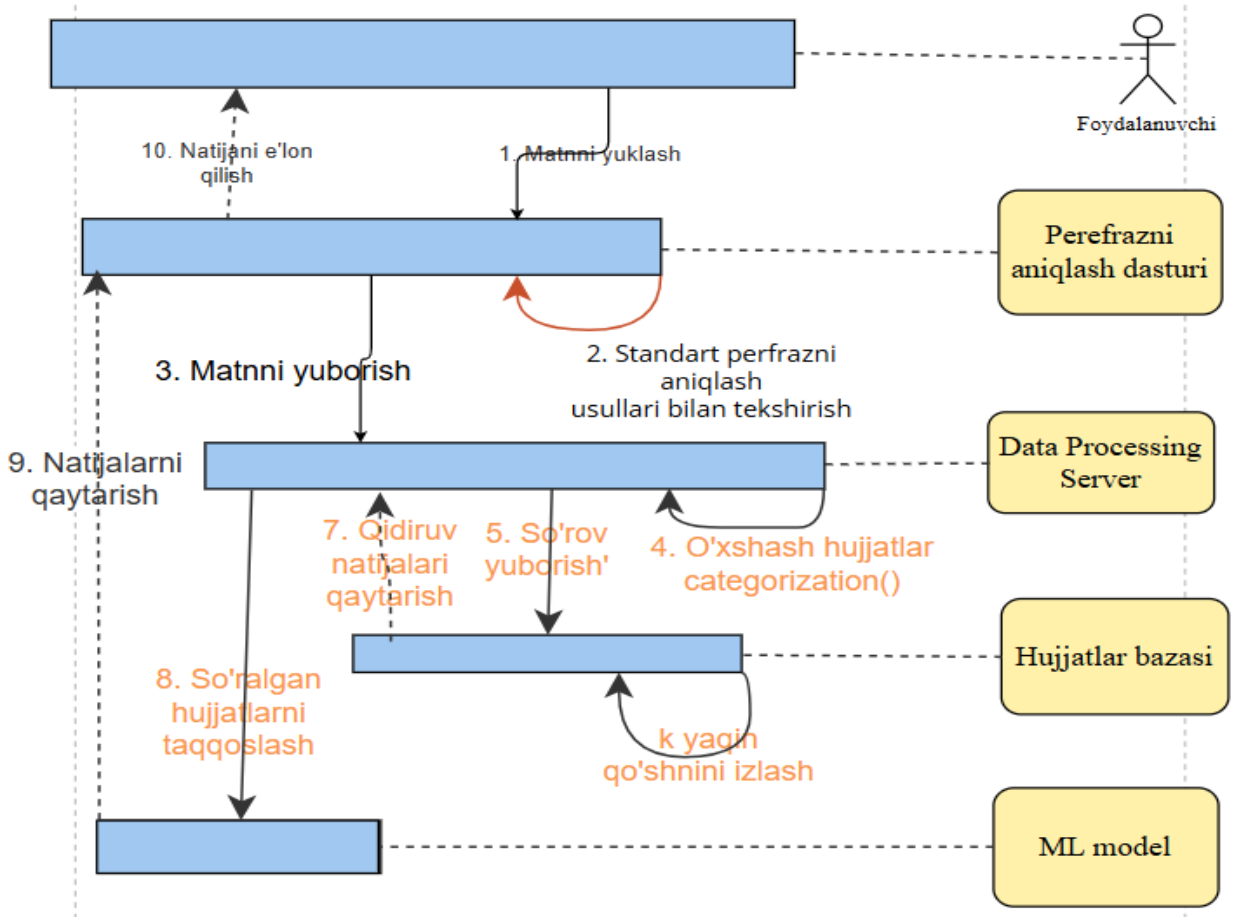
Foydalanish diagrammasida (2-rasm) asosiy ishtirokchilar — tizim funksiyalarini turli usullarda amalga oshiruvchi foydalanuvchilar va ilovalardir. Platformaning individual foydalanuvchisi tizimni plagiat va kontentning o‘ziga xosligini tekshirish ilovasi sifatida ishlatishi mumkin. Bunday tizim talabalar va tadqiqotchilarning akademik darajasini tekshirish uchun qo‘llanilishi mumkin. Bundan tashqari, plagiatni tekshirish tizimi matn muallifini aniqlash uchun ishlatilishi mumkin. Perefraz mavjudligini qo‘shimcha tekshirish orqali bunday tizimning to‘liqligi (eslash ko‘rsatkichi) oshadi. Bu holda, barcha haqiqiy manbalardan to‘g‘ri ko‘rsatilgan manbalar ulushi ortadi.



2-rasm: Perefrazni aniqlash axborot tizimining foydalanish holati diagrammasi

Ma’lumotlarni qayta ishlash serveri ML algoritmlaridan foydalangan holda hujjatlarni taqqoslash usullarini o‘z ichiga oladi va ma’lumotlarni qiyoslash juda resurs talab qiluvchi jarayon bo‘lgani uchun eng yaqin qo‘shnilarni topish uchun vektor joylashtirish yoki hujjatlarni tasniflashni qo‘llaydi. Ma’lumotlarni qayta ishlash serveri shuningdek, klassifikator uchun kiruvchi ma’lumot sifatida ishlatiladigan qo‘shimcha xususiyatlarni ham hisoblash kerak. Bunday qo‘shimcha xususiyatlar umumiy nomlangan obyektlar yoki so‘z turkumlarini o‘z ichiga olishi mumkin. Bunday xususiyatlarni hisoblash uchun maxsus tahlilchilarni qo‘shimcha amalga oshirish talab etiladi. Ketma-ketlik diagrammasi tizim komponentlari o‘zaro ta’sirining vaqt jihatlarini ko‘rsatadi. (3-rasm)

(cc) BY © X.X. Axmedova, Dj.B. Sultanov



1-rasm: Perefrazni aniqlash jarayonlari

Perefrazani aniqlashda ML modelini o‘qitish, sinovdan o‘tkazish va xususiyatlarni ajratish katta hajmdagi ma’lumotlar to‘plami yig‘ish dastlabki qadam hisoblanadi. Perefrazni aniqlash uchun zarur datasetni shakllantirishda o‘zbek tilidagi rasmiy internet saytlari ma’lumotlaridan foydalanildi. Ya’ni *kun.uz, daryo.uz, stadion.uz, qalampir.uz, effect.uz* kabi jami 12 ta rasmiy saytlar tanlab olindi. Internet saytlaridagi matnlarni yuklab olishda Python dasturlash tili va *Scrapy* frameworkidan, murakkab tuzilishga ega va interaktiv buttonlarga ega bo‘lgan internet saytlari uchun *brouzerni avtomatlashtirish* uchun mo‘ljallangan ochiq kodli dasturiy ta’minot *(Selenium Browser Automation Project)* dan foydalanildi.

Xulosa

Ushbu tadqiqot O‘zbek tili matnlari tarkibidagi perefraz jumlalarni aniqlash masalasining yechimiga bag‘ishlangan. Antiplagiat axborot tizimlaridan farqli o‘laroq matn tarkibidagi semantic o‘xshashliklarni aniqlaydi. Bu masalani yechish matnlardagi takroriy jumlalarni aniqlash, dublikatlarni oldini olish, matnlarni annotatsiyasini hosil qilishga xizmat qiladi. Matn tarkibidagi perefraz jumlalarni aniqlashda zarur bo‘lgan dataset rasmiy internet manbalaridan jamlangan bo‘lib, bunda Scrapy va Selenium frameworklari yordamidan foydalanildi.

Foydalanilgan adabiyotlar

1. Pronoza E., Yagunova E., Kochetkova N. Sentence Paraphrase Graphs: Classification Based on Predictive Models or Annotators' Decisions?. In: Sidorov G., Herrera-Alcántara O. (eds) Advances in Computational Intelligence. MICAI 2016. Lecture Notes in Computer Science, vol 10061. Springer, Cham (2017).
2. Pronoza, E., Yagunova, E. Comparison of sentence similarity measures for Russian paraphrase identification. In Artificial

© X.X. Axmedova, Dj.B. Sultanov

Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), pp. 74–82, IEEE (2015a).
3. Pronoza, E., Yagunova, E. Low-Level Features for Paraphrase Identification. In: Sidorov, G., Galicia-Haro, Sofía N. (eds.) MICAI 2015. LNCS, vol. 9413, pp. 59–71. Springer, Cham (2015b).
4. Pronoza, E., Yagunova, E., Pronoza, A. Construction of a Russian Paraphrase Corpus: Unsupervised Paraphrase Extraction. Proceedings of the 9th Russian Summer School in Information Retrieval, August 24–28, 2015, Saint-Petersburg, Russia, (RuSSIR 2015, Young Scientist Conference), Springer CCIS (2015).
5. Androutsopoulos, I., Prodromos Malakasiotis, P. A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research, v. 38: 135–187 (2010).
6. Fernando, S., Stevenson, M. A semantic similarity approach to paraphrase detection. In Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, pp. 45–52, (2008).
7. Pham, N., Bernardi, R., Zhang, Y. Z., Baroni, M. Sentence paraphrase detection: When determiners and word order make the difference. In Proceedings of the Towards a Formal Distributional Semantics Workshop at IWCS 2013, pp. 21–29 (2013).
8. Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiský, T., Blunsom, P. Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664. (2015)
9. He, H., Gimpel, K., Lin, J. Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1576–1586 (2015).
10. M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger, From Word Embeddings to Document Distances, JMLR: W&CP 37 (2015) 957-966.
11. Y. Zhang, J. Baldridge, L. He, Zhang Y. PAWS: Paraphrase Adversaries from Word Scrambling, ArXiv, 2019. DOI: 10.48550/arXiv.1904.01130.
12. V.-A. Oliinyk, V. Vysotska, Y. Burov, K. Mykich, V. B. Fernandes, Propaganda Detection in Text Data Based on NLP and Machine Learning, CEUR workshop proceedings 2631 (2020) 132-144.
13. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, ArXiv, 2019. DOI: 10.48550/arXiv.1907.11692.
14. V. Vysotska, Y. Burov, V. Lytvyn, A. Demchuk, Defining Author's Style for Plagiarism Detection in Academic Environment, in: Proceedings of the International Conference on Data Stream Mining and Processing, DSMP, 2018, pp. 128-133. DOI: 10.1109/DSMP.2018.8478574.
15. N. Shakhovska, I. Shvorob, The method for detecting plagiarism in a collection of documents, in: International Conference on Computer Sciences and Information Technologies, 2015, pp. 42-145.
16. O. Karnalim, G. Kurniawati, Programming Style on Source Code Plagiarism and Collusion Detection, International Journal of Computing 19(1) (2020) 27-38. 17. N. Khairova, A. Shapovalova, O. Mamyrbayev, N. Sharonova, K. Mukhsina, Using BERT model to Identify Sentences Paraphrase in the News Corpus, CEUR Workshop Proceedings Vol-3171, (2022) 38-48.
17. N. Grabar, T. Hamon, Exploitation of the morphology for automatic extraction of general paraphrases of medical terms, Revue Traitement Automatique des Langues 57(1) (2016) 85–109.
18. I. Eshkol-Taravella, N. Grabar, Paraphrastic reformulations in spoken corpora, Lecture Notes in Computer Science 8686 (2014) 425–437.

(cc) BY