IJTIMOIY-GUMANITAR SOHADA ILMIY-INNOVATSION TADQIQOTLAR

ILMIY METODIK JURNALI

ISSN 3060-5059

VOL.3 № 4 2026

FAYLDAGI OBYEKTLARNI SEMANTIK TAHLIL QILISH ALGORITMLARINI ERGONOMIK METODIKASI

Xolmatov Javlon Yusupovich
O‘zbekiston Milliy universiteti Jizzax filiali, katta o‘qituvchi
Mamaraimov Abror Kamalidin o‘g‘li
O‘zbekiston Milliy universiteti Jizzax filiali, katta o‘qituvchi

Annotatsiya
Ushbu maqolada PDF formatidagi elektron adabiyotlar va fayllar ichidan kerakli ma’lumotlarni tezkor topish va ma’lumotlarini semantik jihatdan tahlil qilish masalasi ko‘rib chiqiladi. Sun’iy intellekt asosida ishlab chiqilgan semantik qidiruv algoritmi taklif etiladi. Ushbu tadqiqotda matnlarni semantik tahlil qilish uchun zamonaviy axborot qidiruv va tabiiy tilni qayta ishlash usullari qo‘llanildi. Olingan natijalar shuni ko‘rsatdiki, chuqur o‘rganish modellari klassik usullarga nisbatan yuqori aniqlikni ta’minlaydi.

Kalit so‘zlar: sun’iy intellekt, semantik qidiruv, TF-IDF, Word2Vec, talaba javoblarini tekshirish, PDF qidiruv, matn o‘xshashligi, kosinus o‘xshashligi.

ЭРГОНОМИЧЕСКАЯ МЕТОДОЛОГИЯ АЛГОРИТМОВ ДЛЯ СЕМАНТИЧЕСКОГО АНАЛИЗА ОБЪЕКТОВ В ФАЙЛЕ

Холматов Жавлон Юсупович
Джизакский филиал Национального университета Узбекистана, старший преподаватель
Мамараимов Аброр Камалидин угли
Джизакский филиал Национального университета Узбекистана, старший преподаватель

Аннотация
В данной статье рассматривается вопрос быстрого поиска необходимой информации в электронной литературе и файлах в формате PDF, а также семантического анализа их данных. Предложен алгоритм семантического поиска, разработанный на основе искусственного интеллекта. В данном исследовании для семантического анализа текстов использовались современные методы информационного поиска и обработки естественного языка. Полученные результаты показали, что модели глубокого обучения обеспечивают более высокую точность по сравнению с классическими методами.

Ключевые слова: искусственный интеллект, семантический поиск, TF-IDF, Word2Vec, проверка ответов студентов, поиск в PDF-файлах, сходство текста, косинусное сходство.

ERGONOMIC METHODOLOGY OF ALGORITHMS FOR SEMANTIC ANALYSIS OF OBJECTS IN A FILE

Kholmatov Javlon Yusupovich
Jizzakh Branch of the National University of Uzbekistan, Senior Lecturer
Mamaraimov Abror Kamalidin ugli
Jizzakh Branch of the National University of Uzbekistan, Senior Lecturer

Abstract
This article examines the issue of quickly searching for relevant information in electronic

literature and PDF files, as well as the semantic analysis of their data. A semantic search algorithm developed using artificial intelligence is proposed. This study utilized modern methods of information retrieval and natural language processing for semantic text analysis. The results obtained demonstrated that deep learning models provide higher accuracy compared to classical methods.

Keywords: artificial intelligence, semantic search, TF-IDF, Word2Vec, student response validation, PDF search, text similarity, cosine similarity.

Zamonaviy ta’lim jarayonida raqamli o‘quv qo‘llanmalar keng tarqalmoqda. Talabalarga PDF formatida adabiyotlar berilib, ularning shu nazariy ma’lumotlarni o‘rganilganlik darajasini bilish uchun shular orqali tuzilgan nazariy savollar va topshiriqlar beriladi. Ammo millionlab talaba javoblarini qo‘lda tekshirish imkonsiz[6].

PDF (Portable Document Format) fayli — bu matn, grafik, tasvir, shriftlar va gipermurojaatlarni qurilmadan qat'iy nazar bir xil ko'rinishda saqlaydigan formatdir. U ob’ektlarga asoslangan tuzilmani tashkil etadi.Mavjud avtomatik tekshirish tizimlari faqat kalit so‘zlar bo‘yicha ishlaydi, semantik ma’noni tushunmaydi. Masalan, talaba “kuchli imperiya” yozsa, kitobda “qudratli davlat” deb yozilgan bo‘lsa – tizim xato deb hisoblaydi.

Maqola maqsadi – PDF fayl matni ichidagi javob bilan talaba javobini semantik jihatdan solishtiradigan algoritmlarni tahlil qilish va ergonomik metodikani ishlab chiqish.

Adabiyotlar tahlili

Axborot qidiruv tizimlarida klassik usullar:

– Linear Search – barcha matnni ketma-ket skan qilish O(n)
– Binary Search – saralangan ma’lumotlarda O(log n)
– Inverted Index – kalit so‘zlar indeksi O(1) [1],[5],[7].

1-jadval.

Xususiyat	**Linear Search (Chiziqli qidiruv)**	**Binary Search (Ikkilik qidiruv)**	**Inverted Index (Teskari indeks)**
Ishlash prinsipi	Ma’lumotlar boshidan oxirigacha ketma-ket tekshiriladi	Qidiruv maydoni har qadamda yarmiga bo‘linadi va o‘rtadagi element tekshiriladi	Har bir kalit so‘z uchun alohida indeks yaratiladi: so‘z → hujjatlar ro‘yxati
Vaqt murakkabligi	O(n)	O(log n)	O(1)
Talab qilinadigan shart	Maxsus shart yo‘q, tartibsiz ma’lumotlarda ham ishlaydi	Ma’lumotlar saralangan bo‘lishi kerak	Indeks oldindan yaratilgan bo‘lishi kerak
Afzalligi	Algoritm juda oddiy, implementatsiya qilish oson	Juda tez ishlaydi	Juda tez qidiruv, millionlab hujjatlarda ham samarali
Kamchiligi	Katta hajmdagi ma’lumotlarda sekin	Avval ma’lumotni saralash kerak	Indeks yaratish va yangilash uchun qo‘shimcha xotira va vaqt kerak
Misol	1 mln so‘z ichidan “apple” ni topish uchun 1 mln tekshiruv bo‘lishi mumkin	1 mln saralangan so‘zda taxminan 20 ta solishtirish yetarli	“apple” so‘zi qaysi hujjatlarda borligi indeksdan darhol olinadi

Sun’iy intellekt yondashuvlari:

– TF-IDF – so‘z muhimligini hisoblash

– Word2Vec – so‘zlar vektorlari
– BERT – chuqur semantic tahlil [3],[4],[8],[9].

2-jadval.

Xususiyat	TF-IDF	Word2Vec	BERT
Asosiy g‘oya	So‘zning hujjat ichida qanchalik muhimligini statistik hisoblash	So‘zlarni zich vektorlar (embedding) ga aylantirish	Kontekstga bog‘liq chuqur semantik vektorlar yaratish
Ishlash prinsipi	So‘z chastotasi (TF) va hujjatlar orasidagi noyoblik (IDF) asosida baholanadi	O‘xshash kontekstda kelgan so‘zlar o‘xshash vektor oladi	Transformer modeli yordamida so‘zlar kontekst bilan birga tahlil qilinadi
Asosiy formula / model	TF-IDF = TF × IDF IDF = log(N / (df + 1))	Neyron tarmoq asosidagi embedding modeli (CBOW, Skip-gram)	Transformer arxitekturasi asosidagi chuqur neyron tarmoq
Semantikani tushunishi	Past (faqat statistik)	O‘rtacha (semantik o‘xshashlikni ushlaydi)	Juda yuqori (kontekstni tushunadi)
Kontekstni hisobga olish	Yo‘q	Yo‘q (har so‘z uchun bitta vektor)	Ha (har jumlada boshqa vektor bo‘lishi mumkin)
Hisoblash murakkabligi	Juda tez	O‘rtacha (modelni o‘qitish kerak)	Og‘ir va resurs talab qiladi
Afzalliklari	Oddiy, tez, o‘qitish talab qilmaydi	Semantik o‘xshashlikni yaxshi aniqlaydi	Eng kuchli semantik tushunish
Kamchiliklari	Sinonimlar va kontekstni tushunmaydi	Ko‘p ma’noli so‘zlar uchun bitta vektor beradi	Juda katta model va sekin ishlaydi
Tipik qo‘llanishlar	Qidiruv tizimlari, kalit so‘z ajratish	Semantik o‘xshashlik, tavsiya tizimlari	NLP vazifalari: sentiment, NER, savol-javob
Misol	“apple” so‘zi hujjatda ko‘p uchrasa muhim deb baholanadi	king − man + woman ≈ queen	“bank” so‘zi kontekstga qarab turli ma’no oladi
Qachon ishlatiladi	Tez, oddiy kalit so‘zli qidiruv	So‘zlar o‘rtasidagi semantik o‘xshashlik kerak, lekin resurs cheklangan	Eng aniq semantik tushunish, murakkab matn tahlili

Metodika. Foydalanuvchilar tomonidan ushbu tizimlarni o‘rganishda ayrim jihatlarni hisobga olish talab etiladi. Biz bu yerda ushbu modellarni o‘rganishda “Ekspert baholash metodi” dan foydalanishni taklif qilamiz. [2]

Ekspert baholash metodi quyidagicha amalga oshiriladi: dastlab muammoga ta’sir qiluvchi asosiy omillar yoki muammoning yechimi bo‘lgan bir nechta variantlari tanlab olinadi. Mazkur muammoning yechimlari yoki unga ta’sir qilayotgan asosiy faktorlarni ekspertlarning fikrini jamlash orqali aniqlashda ekspertlarning fikrini olish uchun erkin suhbat yoki savol-javob ko‘rinishidagi intervyu hamda anketalashtirishdan foydalaniladi, ushbu jarayonda har bir ekspert taqqoslanayotgan faktorlarga yoki muqobil variantlarga miqdoriy baho beradi ya‘ni ularni

tabaqalaydi. So‘ngra ekspert guruhlari qatnashchilarning baholari jamlanadi [2].

Ushbu metod yordamida 1- va 2-jadvallar ergonomikasini ko‘rib quyidagi xulosaga kelamiz: talaba javoblarini tekshirish uchun semantik qidiruv eng samarali hisoblanadi.

Taklif etilayotgan g‘oya:

1. PDF dan matn olinadi
2. Talaba javobi olinadi
3. Har ikkisi tozalanadi
4. Vektorga o‘tkaziladi (TF-IDF yoki Word2Vec)
5. O‘xshashlik hisoblanadi
6. Natija chiqariladi

Quyida ushbu algoritmning ishlashini aniq bir misol sifatida ko‘ramiz:

Savol: Mulohaza deb nimaga aytiladi?

1. PDF matnda ushbu javob turibdi: A – Mulohaza deb, faqat chin yoki yolg‘on qiymat qabul qiluvchi darak gaplarga aytiladi.

2. Talaba javobi: Masalan, B – 1 yoki 0 qiymat qabul qiladigan darak gaplar mulohaza deyiladi (yoki, yana boshqacha javob bo‘lishi mumkin)

3. Har ikkisi tozalanadi. Tozalash – bu matndan “keraksiz elementlarni olib tashlab”, uni algoritm uchun tayyorlash.

Tozalash quyidagicha amalga oshiriladi:

1. Kichik harfga o‘tkazish:
   – A1 – mulohaza deb, faqat chin yoki yolg‘on qiymat qabul qiluvchi darak gaplarga aytiladi;
   – B1 – 1 yoki 0 qiymat qabul qiladigan darak gaplar mulohaza deyiladi;
2. Belgilarni olib tashlash:
   – A2 – mulohaza deb faqat chin yoki yolgon qiymat qabul qiluvchi darak gaplarga aytiladi;
   – B2 – 1 yoki 0 qiymat qabul qiladigan darak gaplar mulohaza deyiladi;
3. Tokenizatsiya - so‘zlarga ajratish:
   – A3 – “mulohaza”, “deb”, “aytiladi”, “faqat”, “chin”, “yoki”, “yolgon”, “qiymat”, “qabul”, “qiluvchi”, “darak”, “gaplarga”;
   – B3 – “1”, “yoki”, “0”, “qiymat”, “qabul”, “qiladigan”, “darak”, “gaplar”, “mulohaza”, “deyiladi”;
4. Stop-so‘zlarni olib tashlash – kelishik qo‘shimchalari, “-lar”, “deb”, “edi” va hokazo kabi so‘zlar, qo‘shimchalarni olib tashlash:
   – A4 – “mulohaza”, “aytiladi”, “faqat”, “chin”, “yolgon”, “qiymat”, “qabul”, “qiluvchi”, “darak”, “gap”;
   – B4 – “1”, “0”, “qiymat”, “qabul”, “qiladigan”, “darak”, “gap”, “mulohaza”, “deyiladi”;
5. Stemming/normalizatsiya-so‘zlarni ildizga keltirish:
   – A5 – “mulohaza”, “aytil”, “faqat”, “chin”, “yolgon”, “qiymat”, “qabul”, “qil”, “darak”, “gap”;
   – B5 – “1”, “0”, “qiymat”, “qabul”, “qil”, “darak”, “gap”, “mulohaza”, “deyil”;
6. Bo‘sh joylarni olib tashlash:
   – A6 – “mulohaza”, “aytil”, “faqat”, “chin”, “yolgon”, “qiymat”, “qabul”, “qil”, “darak”, “gap”;
   – B6 – “1”, “0”, “qiymat”, “qabul”, “qil”, “darak”, “gap”, “mulohaza”, “deyil”.

4. Vektorga o‘tkaziladi (TF-IDF yoki Word2Vec) – A va B matnlar lug‘atda umumlashtiriladi. A va B matnlar lug‘atdagi har bir so‘z qiymati 0/1 ga teng bo‘lgan lug‘at-vektorga ajratiladi. Vektor har bir qiymati matndagi mavjud so‘zlarga qarab, mavjud bo‘lsa – 1, mavjud bo‘lmasa – 0, qiymat qabul qiladi.

Lug‘at quyidagi ko‘rinishda bo‘ladi:

D={“mulohaza”, “aytil”, “faqat”, “chin”, “yolgon”, “qiymat”, “qabul”, “qil”, “darak”, “gap”, “1”, “0”, “deyil”}

Har bir matn vektorga aylantiriladi:

V_A=[1,1,1,1,1,1,1,1,1,1,0,0,0]

V_B=[1,0,0,0,0,1,1,1,1,1,1,1,1]

5. O‘xshashlik hisoblanadi – To‘g‘ridan-to‘g‘ri kalit so‘zlar(indeks) o‘xshashligini tekshirilganda:

$$S(A,B): \cos\theta = \frac{A \cdot B}{|A| \cdot |B|}$$

$$A \cdot B = 6$$

$$|A| = 3{,}3$$

$$|B| = 3$$

$$\cos\theta = \frac{A \cdot B}{|A| \cdot |B|} = \frac{6}{3{,}3 \cdot 3} \approx 0{,}6$$

O‘xshashlik 0,6 (60%) ga teng bo‘lmoqda, va bu natijalarni tekshirishda ayrim xatoliklarga olib keladi. Bu yerda asosiy muammo vektorning qiymatlari sifatida qabul qilinayotgan so‘zlar – sonlar ma’no jihatidan bir xil bo‘lsada, lekin qiymat jihatidan har xil hisoblanayapti. Ya’ni: “chin”=”1”, “yolgon”=”0”, “deyil”=”aytil” so‘zlar hisobga olinmayapti.

Endi vektorlarni boshqatdan yaratamiz, ya’ni yuqoridagi ma’no jihatdan bir xil so‘zlarni inobatga olgan holda ushbu V_A, V_B vektorlarni yaratamiz:

Yangi lug‘at quyidagi ko‘rinishda bo‘ladi:

D={“mulohaza”, “aytil”, “faqat”, “chin”, “yolgon”, “qiymat”, “qabul”, “qil”, “darak”, “gap”}

V_A=[1,1,1,1,1,1,1,1,1,1]

V_B=[1,1,0,1,1,1,1,1,1,1]

So‘zlarning semantik tahlil asosida sinonimlarining bir xilligi hisobga olingan holda o‘xshashligini tekshirilganda:

$$A \cdot B = 9$$

$$|A| \approx 3{,}2$$

$$|B| = 3$$

$$\cos\theta = \frac{A \cdot B}{|A| \cdot |B|} = \frac{9}{3{,}2 \cdot 3} \approx 0{,}95$$

E’tibor qaratsak, bu yerda, o‘xshashlik darajasi 0,89 (89%) ni tashkil qilmoqda. Bu bilan B javobning A javobga juda yaqinligini aniqlaymiz.

6. Natija chiqariladi: Natija quyidagi baholash mezonlari asosida qo‘yiladi.

Baholash mezonlari:

O‘xshashlik	**Baho**	**Tavsif**
≥ 0.80	5	To‘liq to‘g‘ri
0.65-0.79	4	Asosiy ma’no to‘g‘ri
0.50-0.64	3	Qisman to‘g‘ri
0.30-0.49	2	Noto‘g‘ri
< 0.30	1	Mavzudan chetga chiqgan

Bu yerda dasturda ushbu Word2Vec modeli yaratilib, matnlar asosida o‘qitiladi:

model = Word2Vec(sentences, vector_size=50, window=3, min_count=1, seed=42)

bu yerda, sentences – o‘quv ma’lumotlari:

sentences = [

tokens_A,

```
        tokens_B,
        ["mulohaza", "fikr"],
        ["chin", "togri"],
        ["yolgon", "notogri"],
        ["aytil", "deyil"],
        ["qiluvchi", "qiluvchi"],
        ["darak", "gap"]
    ]
```

bu:

– model nimani o‘rganishini belgilaydi;

– qancha katta bo‘lsa, model shuncha aniq bo‘ladi.

vector_size=50 – vektor uzunligi

har bir so‘z nechta sondan iborat bo‘ladi, ya’ni, so‘zlarni sonlarga aylantirib beradi. Masalan: “mulohaza” = [0.12, -0.34, 0.22, ..., 0.10] (50 ta son).

Vektor uzunligini tanlashda ushbu jadval asosida tanlansa maqsadga muvofiq hisoblanadi:

Vektor uzunligi qiymatlari	**Natija**
10-50	tez, oddiy
100-300	yaxshi aniqlik
300+	kuchli, lekin sekin

Bizni eksperimentimizda vector_size=50 o‘rtacha hisoblanadi.

window=3 – so‘zlarning atrofini o‘rganish parametri. Model bir so‘z atrofidagi nechta so‘zni ko‘rishini anglatadi. Masalan, ““mulohaza” “chin” “yolgon” qiymat” “qabul”“ berilgan bo‘lsa, va agar window=3 bo‘lsa: “yolgon” uchun: [“mulohaza”, “chin”, qiymat”, “qabul”] bo‘ladi. Quyida window uchun kattaliklar berilgan:

window	**Natija**
kichik (2–3)	lokal ma’no
katta (5–10)	umumiy ma’no

Bizni eksperiment uchun window=3, bu holat tezkor ishlash uchun va local ma’noni anglash uchun yetarli hisoblanadi.

min_count=1 – minimal chastota

so‘z necha marta chiqsa model uni o‘rganadi

Qiymat	**Natija**
1	hamma so‘zlar olinadi
2+	kam uchraydiganlar tashlanadi

Ushbu eksperimentda min_count =1 – barcha so‘zlar ishlatiladi

seed=42 – tasodifiylikni boshqarish. model har safar bir xil natija berishi uchun ishlatiladi. Aks holda, dastur har safr ishga tushirilganda vector qiymatlar o‘zgaradi va o‘qitilish murakkablashib, vaqt ko‘p sarflaydi. Bu yerda – bizning misolimizda 42-shunchaki standart son.

Biz yaratilgan Word2Vec modeli: chin ≈ 1, yolgon ≈ 0, mulohaza ≈ fikr va hokazo kabi o‘rganadi. Har bir so‘z uchun alohida vektor beriladi. Bu yerda, Word2Vec so‘zlarni kontekst orqali o‘rganadi. Ya’ni, “chin” va “1” birga chaqirilsa o‘zaro yaqinlashadi.

Xulosa. Xulosa qilib aytadigan bo‘lsak, taklif etilgan algoritm PDF kitoblardan talaba javoblarini semantik tahlil orqali yuqori aniqlikda tekshiradi. Tizim o‘zbek tillarida muvaffaqiyatli ishlaydi.

Ushbu eksperimentdagi Word2Vec(...) modeli quyidagini ta’minlaydi:

– sinonimlarni aniqlash;
– semantik yaqinlik;
– matnni tushunish.

Eng muhimi model sifati bu, sentences sifati. Agar, dataset kichik bo‘lsa natija yomon yoki o‘rtacha, va agar katta dataset bilan ishlanganda natija yuqori bo‘ladi.

Word2Vec modelida vektor o‘lchami, kontekst oynasi va minimal chastota kabi parametrlar modelning semantik aniqligiga sezilarli ta’sir ko‘rsatadi. Ushbu parametrlarning optimal tanlanishi matnlar orasidagi o‘xshashlikni aniqlash sifatini oshiradi.

FOYDALANILGAN ADABIYOTLAR RO‘YXATI

1. Abdurahmonov S. Matnlarni qayta ishlash va tabiiy til texnologiyalari. – Toshkent, 2020.
2. Axmedov J. R. Tizimli tahlil: darslik. – Jizzax: Ilm nuri print MChJ, 2023. – 266 bet.
3. Devlin J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // NAACL-HLT. – 2019.
4. Goldberg Y., Levy O. word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method [Electronic resource]. – 2014. – Mode of access: arXiv.
5. Jones K. S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. – 1972. – Vol. 28, № 1. – P. 11–21.
6. Karaboga D. et al. Automatic Short Answer Grading Using Deep Learning // Computers & Education. – 2020.
7. Karimov A., Tursunov B. Sun’iy intellekt va uning qo‘llanilishi. – Toshkent, 2021.
8. Le Q., Mikolov T. Distributed Representations of Sentences and Documents // International Conference on Machine Learning (ICML). – 2014.
9. Mikolov T. et al. Efficient Estimation of Word Representations in Vector Space [Electronic resource]. – 2013. – Mode of access: arXiv:1301.3781.