

UPOTREBLJIVOST GRAPHQL BIBLIOTEKE ZA RAZVOJ NLP INTERFEJSA**USABILITY OF GRAPHQL LIBRARY FOR DEVELOPMENT OF NLP INTERFACES**Dragan Borkovac, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – *Konceptualno rešenje sistema koje korisnicima omogućava pristup podacima iz skladišta koristeći naredbe na prirodnom jeziku. Korisnici putem NLP interfejsa postavljaju upite na prirodnom jeziku kako bi dobili željene podatke. Sistem automatski prevodi upite na prirodnom jeziku u GraphQL upite kojim je pokriven celokupan API sistem, uključujući i skladište podataka.*

Ključne reči: *NLP, GraphQL, API*

Abstract – *Conceptual solution of a system that allows users to access data from the warehouse using natural language queries. Through the NLP interface, users make queries in natural language in order to get the desired data. The system automatically translates natural language queries into GraphQL queries that cover the entire API system, including the data warehouse.*

Keywords: *NLP, GraphQL, API***1. UVOD**

U današnjem digitalnom dobu, velika količina podataka je dostupna korisnicima širom sveta. Iz tog razloga sve češće dolazi do pojave problema efikasnog pristupa i pretraživanja podataka. Ovaj problem je najizraženiji kada je u pitanju raznolikost formata i struktura u kojima se podaci čuvaju. Tradicionalni načini pretrage, često zasnovani na predefinisanim ključnim rečima ili kompleksnim upitima, se ovde pokazuju kao neefikasni i zahtevaju od korisnika duboko razumevanje strukture podataka.

Upravo u cilju olakšavanja pretrage i pristupa podacima, nastaje ideja o sistemu koji korisnicima omogućava pristup podacima koristeći naredbe na prirodnom jeziku. Ovaj rad istražuje kako se putem interfejsa za obradu prirodnog jezika (NLP) mogu postavljati upiti na prirodnom jeziku kako bi se dobili željeni podaci iz skladišta podataka. Sama srž sistema leži u automatskom prevodenju ovih upita na prirodnom jeziku u GraphQL upite koji pokrivaju celokupni API sistem, uključujući i skladište podataka. Korišćenje GraphQL biblioteke omogućava precizno definisanje podataka koje korisnik želi da dobije, što čini ovaj pristup izuzetno fleksibilnim i efikasnim.

U okviru rada, razmotreni su osnovni koncepti NLP i GraphQL tehnologija i analizirano je trenutno stanje u

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red.prof.

praksi kroz pregled postojećih istraživanja koja povezuje ove dve tehnologije. Pored toga, istražene su osnove prepoznavanja jezika i tehnike semantičke analize koje su neophodne za uspešno mapiranje prirodnog jezika na GraphQL upite. Nakon toga, razmotren je dizajn i sama implementacija NLP sistema, kao i razvoj GraphQL interfejsa. Na kraju, demonstrirana je upotreba ovakvog sistema u realnom scenariju.

Cilj ovog rada jeste predstavljanje konceptualnog rešenja koje kombinuje prednosti NLP i GraphQL tehnologija kako bi omogućilo korisnicima intuitivno i efikasno pretraživanje podataka putem naredbi na prirodnom jeziku. Ovakvo rešenje može imati širok spektar primena, od olakšavanja pristupa kompleksnim bazama podataka do unapređenja korisničkog iskustva u različitim digitalnim okruženjima. Ono može smanjiti krivu učenja za programere kada je u pitanju GraphQL tehnologija, a može i predstavljati potencijalni alat za pristup podacima za netehnička lica.

2. OSNOVNI KONCEPTI I STANJE U PRAKSI

U ovom poglavlju biće predstavljani osnovni koncepti obrade prirodnog jezika i GraphQL biblioteke.

2.1. Uvod u NLP

Obrada prirodnog jezika (NLP) odnosi se na granu računarske nauke - konkretnije, granu veštačke inteligencije (AI) koja se bavi davanjem računarima sposobnosti da razumeju tekst i reči na sličan način kao i ljudi [1].

NLP kombinuje računarsku lingvistiku, zasnovanu na pravilima modelovanja ljudskog jezika, sa statističkim, mašinskim i dubokim učenjem [1]. Ove tehnologije omogućavaju računarima da obrade ljudski jezik u obliku tekstualnih ili glasovnih podataka i da "razumeju" njegovo potpuno značenje, uz nameru i emociju govornika ili pisca [1].

NLP predstavlja osnovu programskih sistema koji prevode tekst sa jednog jezika na drugi, odgovaraju na govorne komande i brzo sumiraju velike količine teksta [1]. Interakcija sa NLP sistemima sreće se i u obliku glasom upravljanih GPS sistema, digitalnih asistenata, softvera za pretvaranje govora u tekst, chatbotova za korisničku podršku i slično [1]. NLP takođe igra sve veću ulogu i u poslovnim rešenjima koja pomažu u optimizaciji poslovnih operacija, povećavaju produktivnost zaposlenih i pojednostavljaju razne poslovne procese [1].

2.2. Uvod u GraphQL

GraphQL je tip programskog interfejsa aplikacije (ili upitni jezik), ali i osnovni mehanizam za odgovaranje na postavljene upite [2].

GraphQL korisnicima pruža pojednostavljeni API idealan za mobilne aplikacije i implementacije sistema sa kompleksnim šemama gde su potrebni samo određeni podskupovi podataka [2]. Koristeći GraphQL programeri mogu precizno da definišu podatke koje žele da preuzmu [2]. Pored toga, GraphQL korisnicima omogućava preuzimanje podataka po zahtevu, a ne sve odjednom, trenutno primenjivanje promena i integraciju više izvora podataka [2].

3. MAPIRANJE PRIRODNOG JEZIKA NA GRAPHQL

U ovom poglavlju biće dat pregled različitih tehnika koje predstavljaju osnovu za precizno mapiranje prirodnog jezika na GraphQL upite.

3.1. Tokenizacija

Tokenizacija, u domenu obrade prirodnog jezika i mašinskog učenja, predstavlja proces pretvaranja tekstualnog niza u manje delove koji se zovu tokeni [3]. Ovi tokeni mogu biti mali poput karaktera ili dugi kao reči [3]. Bitnost ovog procesa ogleda se u tome što pomaže mašinama da razumeju ljudski jezik razbijajući ga na male komade, koji su lakši za analizu [3].

Primarni cilj tokenizacije je predstavljanje teksta na način koji je odgovarajući za mašine, a bez gubitka konteksta [3]. Pretvaranjem teksta u tokene, algoritmi mogu lakše da identifikuju obrasce što je ključno za razumevanje i odgovarajuće reagovanje na ljudski unos [3].

3.2. Parsiranje

Parsiranje, u oblasti obrade prirodnog jezika, je proces ispitivanja gramatičke strukture i odnosa unutar date rečenice ili teksta [4]. Ono podrazumeva analizu teksta da bi se utvrdile uloge određenih reči, kao što su imenice, glagoli i pridevi, kao i njihovi međusobni odnosi [4].

Ova analiza proizvodi strukturiranu reprezentaciju teksta, omogućavajući NLP procesima da razumeju kako se reči u frazi povezuju jedna sa drugom [4]. Parseri otkrivaju strukturu rečenice konstruisanjem stabala zavisnosti koja ilustruju hijerarhijske i sintaksičke odnose između reči [4].

3.3. Prepoznavanje imenovanih entiteta (NER)

Prepoznavanje imenovanih entiteta je komponenta ekstrakcije informacija koja ima za cilj da identifikuje i kategorizuje imenovane entitete unutar nestrukturiranog teksta [5]. NER podrazumeva identifikaciju ključnih informacija u tekstu i klasifikaciju u skup unapred definisanih kategorija [5]. Entitet je stvar o kojoj se konstantno govori ili na koju se poziva u tekstu, kao što su imena osoba, organizacija, lokacija, vremenskih izraza, količine, procenata i više unapred definisanih kategorija [5].

3.4. Analiza sentimenata

Analiza sentimenata predstavlja proces klasifikacije kojim se utvrđuje da li je blok teksta pozitivan, negativan ili neutralan [6]. Cilj koji rudarstvo sentimenata (eng. sentiment mining) pokušava da postigne je da se analiziraju mišljenja ljudi na način koji može pomoći preduzećima da se dalje razvijaju i šire [6]. Fokusira se ne samo na polaritet (pozitivan, negativan i neutralan), već i na emocije (srećne, tužne, ljute, itd.) [6].

4. DIZAJN I IMPLEMENTACIJA NLP SISTEMA

Koraci u razvoju NLP sistema:

1. Prikupljanje podataka za treniranje

Prvi korak je prikupljanje velikog skupa podataka rečenica na prirodnom jeziku i njima odgovarajućih GraphQL upita [7]. Ovi podaci mogu se prikupiti ručno ili koristeći već postojeće skupove podataka koji su dostupni na mreži [7].

2. Pretprocesiranje podataka

Sledeći korak je pretprocesiranje podataka čišćenjem, normalizacijom i tokenizacijom teksta [7]. Ovo može uključivati uklanjanje stop reči, interpunkcije i specijalnih znakova, kao i pretvaranje teksta u mala slova [7].

3. Treniranje jezičkog modela

Obučavanje jezičkog modela koristeći prethodno pretprocesirane podatke da generiše GraphQL upite na osnovu ulaznih rečenica [7]. Opcije su fino podešavanje pretreniranog modela ili obučavanje novog modela od nule [7].

4. Evaluacija modela

Kada je model obučan, trebalo bi proceniti njegove performanse koristeći skup testnih rečenica i njima odgovarajućih upita [7]. Mogu se koristiti metrike kao što su tačnost, preciznost i pamćenje da bi se izmerile performanse modela [7].

5. Primena modela

Kada je model obučan i evaluiran, može biti primenjen kao veb servis ili API kome korisnici mogu da pristupe kako bi generisali odgovarajuće GraphQL upite na osnovu zadatih upita na prirodnom jeziku [7].

6. Poboljšavanje modela

Tokom vremena model treba nastaviti poboljšavati prikupljanjem povratnih informacija od korisnika i obučavanjem na novim podacima [7]. Ova poboljšanja će učiniti da model vremenom postane tačniji i robusniji [7].

4.1. Treniranje jezičkog modela

BART

BART model, predstavljen 2019. godine, je model sekvenca-do-sekvence obučan kao autoenkoder za smanjenje šuma [8]. Ovo znači da fino podešen BART model može uzeti tekstualnu sekvencu (na primer, na engleskom jeziku) kao ulaz i proizvesti drugačiju tekstualnu sekvencu na izlazu (na primer, na francuskom jeziku) [9]. Ovaj tip modela je relevantan za mašinsko

prevođenje. odgovaranje na pitanja, sumiranje teksta ili klasifikaciju sekvenci (kategorizacija rečenica ili leksema ulaznog teksta) [9].

Pošto nenadgledano pretreniranje BART transformatora rezultuje jezičkim modelom, moguće je fino podesiti ovaj jezički model za određeni NLP zadatak [9]. Kako je model već pretreniran, za fino podešavanje nisu potrebni masivni označeni skupovi podataka (u odnosu na ono što bi bilo potrebno za obuku od nule) [9]. BART model se može fino podesiti na skupove podataka specifičnih za domen za razvoj aplikacija kao što su medicinski čebotovi, pretvaranje teksta na prirodnom jeziku u programski kod ili SQL upite, aplikacije za prevođenje jezika specifičnih za kontekst ili alat za parafraziranje istraživačkih radova [9].

BART transformator je obučen kao autoenkoder za uklanjanje šuma, tako da podaci za obuku uključuju „pokvaren“ tekst ili tekst sa šumom i njegovo mapiranje u čisti ili originalni tekst [9]. Format obuke je sličan obuci bilo kog autoenkodera za smanjenje šuma [9]. Autoenkoderi se obučavaju da uklone šum ili poboljšaju kvalitet slike tako što imaju slike sa šumom u podacima obuke mapirane sa čistim, originalnim slikama kao ciljem [9].

T5 model

Model T5, skraćeno od Text-to-Text Transfer Transformer, je model obrade prirodnog jezika koji je razvio Google [10]. Zasnovan je na transformer arhitekturi, koja je vrsta neuronske mreže za koju se pokazalo da je veoma efikasna u NLP zadacima [11]. Jedinstven je jer je dizajniran da bude veoma prilagodljiv i može se koristiti za širok spektar zadataka, uključujući klasifikaciju teksta, odgovaranje na pitanja i prevod jezika [11].

Sastoji se od enkoder-dekoder strukture, slično drugim sekvencija-do-sequencija modelima [11]. T5 model koristi skup slojeva transformatora u komponentama enkodera i dekodera, omogućavajući mu da uoči i obradi hijerarhijske reprezentacije ulaznih i izlaznih sekvenci [11].

Enkoder obrađuje ulaznu sekvenciju, koja može biti kombinacija opisa zadatka i ulaznih podataka [11]. Primenjuje mehanizme samo-pažnje kako bi uočio odnose između reči i kontekstualnih informacija [11]. Mehanizam samo-pažnje omogućava modelu da se bavi različitim delovima ulazne sekvence, uzimajući u obzir zavisnosti i odnose između reči [11].

Dekoder, s druge strane, generiše izlaznu sekvenciju [11]. Uzima konačno skriveno stanje enkodera kao ulaz i vodi računa o relevantnim delovima ulazne sekvence tokom procesa dekodiranja [11]. Dekoder generiše izlaz korak po korak, koristeći mehanizam pažnje kako bi se fokusirao na različite delove ulaza dok predviđa sledeći token [11].

I slojevi ekodera i dekodera u T5 modelu koriste mehanizam samo-pažnje sa više glava, što omogućava modelu da razmotri više perspektiva i uhvati različite zavisnosti unutar ulaznih i izlaznih sekvenci [11]. Mehanizam pažnje poboljšava sposobnost modela da se nosi sa zavisnošću dugog dometa i efikasno uoči kontekstualne informacije [11].

T5 model takođe uključuje poziciono kodiranje za kodiranje informacija o položaju ulazne sekvence [11]. Ovo poziciono kodiranje pomaže modelu da razume redosled i poziciju leksema u nizu, što je ključno za uočavanje sekvencijalne prirode jezika [11].

5. DIZAJN I IMPLEMENTACIJA GRAPHQL API SISTEMA

Kreiranje GraphQL šeme

Šema je struktura podataka koje GraphQL klijenti mogu da zahtevaju.

Na listingu broj 1 prikazan je programski kod za kreiranje GraphQL šeme.

```
const { ApolloServer, gql } = require('apollo-server');

const typeDefs = gql`

  type Apartment {
    title: String
    description: String
    price: Float!
    floor: Int!
    area: Float!
    structure: Float!
    city: String!
    constructionYear: Int
  }

  type Query {
    apartments: [Apartment]
  }
`;
```

Listing 1. Kreiranje GraphQL šeme

Resolver funkcije

Dok šema definiše strukturu naše API biblioteke, *resolver* funkcije definišu odakle dolaze podaci. Koristeći GraphQL podaci se mogu preuzimati iz brojnih izvora podataka, poput SQL i NoSQL baza podataka, REST API biblioteka, drugih GraphQL API biblioteka, pa čak i iz samog statičkog JSON objekta).

6. DEMONSTRACIJA UPOTREBE SISTEMA

Prvi korak u upotrebi sistema jeste postavljenje upita na prirodnom jeziku. Ovaj upit postavlja korisnik sistema putem NLP interfejsa.

Primer upita:

“Pronađi mi stanove u Novom Sadu sa cenom ispod 150.000€ izgrađene najranije 2020. godine.”

Na osnovu upita na prirodnom jeziku, NLP sistem generiše odgovarajući GraphQL upit.

Primer odgovarajućeg GraphQL upita prikazan je na listingu broj 2:

```
query {
  apartments(input: {
    city: "Novi Sad"
    maxPrice: 150000
    minConstructionYear: 2020 })
  {
    title
    description
    price
    city
    constructionYear
  }
}
```

Listing 2. Primer izgenerisanog GraphQL upita

Nakon toga, generisani GraphQL upit se izvršava i dobijeni rezultati se prikazuju korisniku.

Primer dobijenih rezultata prikazan je na listingu broj 3:

```
{
  "data": {
    "apartments": [
      {
        "title": "Jednosoban stan na
Grbavici",
        "description": "Nov stan na odličnoj
lokaciji.",
        "price": 99.000,
        "city": "Novi Sad",
        "constructionYear": 2023
      }
    ]
  }
}
```

Listing 3. Primer rezultata izvršavanja GraphQL upita

7. PRIMENA I UPOTREBLJIVOST GRAPHQL BIBLIOTEKE U NLP SISTEMIMA

Prezentovano rešenje je od velike pomoći pri:

- konverziji složenih upita na prirodnom jeziku,
- rukovanju složenim filterima i agregacijama,
- povećanju efikasnosti upita uz smanjenje broja API poziva,
- jednostavnom rukovanju relacionim podacima i
- integraciji sa drugim tehnologijama i alatima.

8. ZAKLJUČAK

U ovom radu istražena je ideja povezivanja NLP interfejsa i GraphQL API sistema radi olakšavanja pretrage podataka koristeći upite na prirodnom jeziku. Predstavljani su osnovni koncepti ove dve tehnologije i napravljen je pregled trenutnog stanja u praksi. Diskutovani su brojni NLP pristupi i tehnike i neophodni procesi za mapiranje upita na prirodnom jeziku na GraphQL upite. U poglavlju o implementaciji sistema, pružen je pregled potrebnih koraka za uspešnu realizaciju NLP sistema i razmotreni su potencijalni jezički modeli koji bi mogli dovesti do najboljih rezultata, poput BART i T5 modela. Dostupan je i pregled kreiranja jednostavne GraphQL šeme neophodne za implementaciju GraphQL

API sistema. Na kraju, prezentovana je upotreba sistema i predstavljeni su dodatni primeri koji svedoče o upotrebljivosti GraphQL biblioteke za razvoj NLP interfejsa.

Kako je predstavljeno samo konceptualno rešenje dalji koraci bi podrazumevali realnu implementaciju predloženog sistema. Ovaj korak bi omogućio testiranje predloženih modela jezika i procenu tačnosti i efikasnosti rezultata koje oni pružaju.

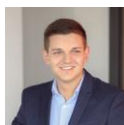
Kroz postojanje brojnih NLP sistema koji se fokusiraju na generisanje SQL upita na osnovu prirodnog jezika jasno je da potreba za ovakvim sistemima postoji. Kada se na to doda jedna od najvećih prednosti GraphQL biblioteke, odnosno mogućnost istovremenog korišćenja različitih izvora podataka, uključujući SQL i NoSQL baze podataka, REST API, mikroservise i slično, očekivan je još veći doprinos sistema koji integrišu NLP i GraphQL.

Sledeći korak u oblasti povezivanja NLP i GraphQL tehnologija jeste kreiranje javno dostupnog skupa podataka za treniranje jezičkih modela koji mapira upite na prirodnom jeziku na njima odgovarajuće GraphQL upite. Ovo bi, ne samo ubrzalo razvoj sistema koji kombinuju ove dve tehnologije već i velikoj meri povećalo tačnost korišćenih jezičkih modela.

9. LITERATURA

- [1] <https://www.ibm.com/topics/natural-language-processing> (pristupljeno u junu 2024.)
- [2] <https://graphql.org> (pristupljeno u junu 2024.)
- [3] <https://www.datacamp.com/blog/what-is-tokenization> (pristupljeno u junu 2024.)
- [4] <https://intellipaat.com/blog/what-is-parsing-in-nlp/> (pristupljeno u junu 2024.)
- [5] <https://www.geeksforgeeks.org/named-entity-recognition/> (pristupljeno u junu 2024.)
- [6] <https://www.geeksforgeeks.org/what-is-sentiment-analysis/> (pristupljeno u junu 2024.)
- [7] <https://ethglobal.com/showcase/the-graph-ai-z9cgp> (pristupljeno u junu 2024.)
- [8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", 2019
- [9] <https://www.projectpro.io/article/transformers-bart-model-explained/553> (pristupljeno u junu 2024.)
- [10] N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", 2020
- [11] <https://databasecamp.de/en/ml-blog/t5-model> (pristupljeno u junu 2024.)

Kratka biografija:



Dragan Borkovac rođen je u Rumi 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2024.god. kontakt: borkovac.dragan96@gmail.com