Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Inteligentné informačné systémy

Vývoj súčastí webovej aplikácie na sémantickú anotáciu datasetov

Výskumný Zámer Diplomovej Práce

Peter Plevko
2022/2023

Ing. Miroslav Rác

1 Úvod

Dátový vedci majú veľmi náročnú prácu. Pre zjednodušenie si tejto náročnej práce potrebujú infraštruktúru ktorá im umožní efektívne pracovať s datasetmi. Taktiež potrebujú infraštruktúru ktorá im pomôže s vyhodnocovaním modelov umelej inteligencie. Potreby dátových vedcov zahŕňajú uchovávanie, organizovanie a anotáciu datasetov v neposlednej rade distribúciu týchto datasetov.

Práca dátových vedcov je rôzna. Cieľom je vytvoriť webovú aplikáciu ktorá umožňuje pracovať s datasetom súčasne človeku (user interface) aj stroju (API). Avšak tu nám nastáva problém. Problémom je, že je potrebné zachovať kompatibilitu s nástrojmi ktoré používajú dátový vedci. Tento problém otvára bohaté možnosti implementácie. Avšak treba brať ohľad na viacero kritérií napríklad cena, dostupnosť, flexibilita a iné. Cieľom projektu je analyzovať požiadavky na architektúru a dátový model pre anotáciu datasetov metadátami. Súčasťou práce je vyhodnotiť zvolenú metódu a technológie vzhľadom na niektoré z vyššie spomenutých kritérií.

2 Analýza problematiky

Naša web stránka bude slúžiť dátovým vedcom ako knowledge base. Knowledge base je znalostná databáza, je to online knižnica informácií o produkte, službe, oddelení alebo téme [1]. V tejto znalostnej databáze sa budú ukladať dáta po každom anotovaní datasetu používateľom. To znamená ak používateľ anotuje stĺpec s názvom plocha viem, že sa jedná o fyzikálnu veličinu plochy a taktiež viem aj v akej jednotke je udávaná (metre štvorcové). Naša web stránka si to uloží a pri ďalšom anotovaní stĺpca s rovnakým menom vyhodí používateľovi možnosť použiť túto uloženú hodnotu. Ak hodnota ktorú užívateľ potrebuje pridať ešte nie je pridaná, tak ju pridá. Užívateľ si bude môcť všetky tieto hodnoty pozrieť a taktiež si bude môcť pozrieť ich vzťahy. Napríklad ak budem mať hodnotu osoba tak táto osoba bude mať v sebe hodnotu email. To znamená, že medzi osobou a emailom platí nejaký vzťah. Tento pristúp je poloautomaticky pretože potrebujeme vstup od používateľa ale o ukladanie sa staráme už mi. Plne automatické vytváranie anotácii je nevyriešený problém [5].

Knowledge engineering je oblasť umelej inteligencie (AI), ktorá sa snaží napodobniť úsudok a správanie ľudského experta v danej oblasti [8]. Dátoví vedci môžu použiť naše API na získanie dát pre trénovanie umelej inteligencie alebo ak už ju majú natrénovanú tak na vyhodnotenie či bola natrénovaná správne. Webová stránka bude pasívna to znamená, že je potrebný užívateľov pokyn (požiadavka na API) aby mu boli dodané informácie. Naše API mu vráti dataset spolu s anotovanými metadátami. Pri tvorbe web stránky nám vznikajú nasledovné otázky. V akom jazyku programovať klienta ? V akom jazyku programovať server ? Použiť rámec ak áno aký ? Použiť server alebo serverless. Použiť relačnú databázu, nerelačnú alebo grafovú ? V akom formáte posielať anotované metadáta o datasete ? V nasledujúcich sekciách sa pozrieme na možnosti akými môžme vyššie spomenuté otázky zodpovedať.

2.1 Klient

V tejto časti sa budem venovať analýze klientskej časti. Podľa stack overflow survey z roku 2022 sú momentálne najpopulárnejšie jazyky na strane klienta JavaScript, HTML/CSS, Python, Java, C#, PHP [7]. Užitočné je aj použitie nejakého rámca, tie nám dovoľujú písať menej kódu, poskytujú knižnice. Mnohé úlohy ktoré budú musieť vývojári v rámci webových aplikácií vykonávať, sú bežné. Príkladom je validácia formulára, sanitácia údajov a operácie CRUD (Create, Read, Update a Delete). Namiesto toho, aby sme pre tieto úlohy museli písať vlastné funkcie, môžeme jednoducho použiť tie, ktoré sú súčasťou rámca. Medzi najpopulárnejšie rámce patria PHP Laravel, Symfony, Javascript React, Vue.js, Python django a mnoho ďalších [2].

2.2 Server

Taktiež máme na výber medzi server a serverless prístupom. Serverless môžeme definovať ako dve oblasti. Prvou je BaaS (backend as a service) ako napríklad firebase. BaaS nahrádza backend externými servicami ktoré poskytujú firmy. Druhou je FaaS (functions as a service) tento pristúp poskytuje platformu pre používateľov na ktorej môžu programovať aplikácie bez toho aby sa starali o infraštruktúru týmto typom je napríklad AWS Lambda. Rozdiel medzi BaaS a FaaS je v tom, že BaaS sa zaoberá celou funkcionalitou servera pričom FaaS reaguje iba na udalosti. Veľkou výhodou serverlessu je to, že platíme iba za použitý čas ale pri klasickom serverovom prístupe platíme aj za nečinný čas [6].

2.3 Databáza

Teraz sa poďme pozrieť nato aký typ databáz použiť. Ako som už vyššie spomínal pre moju webovú aplikáciu je potrebné si ukladať dáta a ich vzťahy. Na výber sa nám ponúka viacero typov databáz. Relačné databázy akými sú napríklad PostgreSQL, grafové databázy ako Neo4j, nerelačné databázy ako MongoDB a v neposlednom rade NewSQL ako SurrealDB. NewSQL je trieda relačných databáz ktoré sa snažia poskytnúť škálovateľnosť nerelačných databáz s tým, že zachovajú ACID [4]. Teraz si vysvetlíme čo je to ACID. Je to skratka zo začiatočných písmen anglických slov Atomicity, Consistency, Isolation a Durability. Všetky tieto princípy sa snaží dodržovať. Bude potrebné dobre sa zamyslieť nad štruktúrou našich dát, nad tým aké operácie budeme nad databázou vykonávať a ktorá je pre nás tou najlepšou voľbou.

2.4 Formát dát

V neposlednom rade nám nastáva otázka v akom formáte posielať dáta. Dáta ktoré pošlem môžu nadobúdať napríklad formát JSON a dataset CSV. Otázkou však je ako takéto niečo uložiť do databázy. Ak by som mal JSON dokument databázu ako MongoDB mohol by som pekne ukladať JSON objekty. Ak by som použil PostgreSQL databázu musel by som tieto metadáta zmeniť na binárne

dáta. Dataset by mohol byť serializovaný na objekt pickle ktorý sa používa v programovacom jazyku python a ten následne uložený v binárnom poli v PostgreSQL. Formát CSV by bolo potrebné taktiež zmeniť na binárny formát a následne ho uložiť. Veľmi veľkým problémom ktorý je potrebné riešiť je, že CSV ktoré mi bude dodané dátovým vedcom bude mať vždy iný formát. To znamená iné stĺpce a iné veľkosti. Nakoniec nám nastáva aj problém autocompletu pretože pri anotovaní sa môžu volať veci aj rovnakým názvom a pritom znamenať rozličnú vec, otázka je ako rozlíšim dva rovnaké názvy pre dve veci ktoré hovoria o niečom inom. Môžem použiť názov a po hoveri aj tooltip s bližšou špecifikáciou.

3 Cieľ práce

Cieľom práce je vytvoriť produkt, webovú aplikáciu ktorá dovolí užívateľovi nahrať dataset vo formáte CSV. Tento dataset si následne anotuje buď pomocou pridania sémantického typu alebo pomocou použitia už existujúcich. Sémanticky typ nám bližšie určuje aký typ hodnoty môžeme v danom stĺpci očakávať. Tieto existujúce dáta si môže aj prezerať bude nato vytvorená samostatná stránka. Tento dataset je následne uložený a môže si ho aj stiahnuť. Taktiež môže poslať request na endpoint a dáta sú mu poslané. Prínosom našej aplikácie bude to, že bude serverless. Z toho vyplýva jej dobrá cenová optimalizácia. Serverless totižto neplatíme za nečinný čas platíme len keď sa používa.

4 Postup implementácie

V tejto časti popíšem ako budem postupovať od začiatku až po vytvorenie finálneho produktu. Najdôležitejšou častou bude analýza v ktorej sa pozriem na všetky možné implementačné možnosti. Každú túto možnosť zanalyzujem pozriem sa na jej výhody a nevýhody a nakoniec vyberiem tu najlepšiu. Finálnym produktom analýzy bude koncový výber technológii a postupov pre webovú aplikáciu.

Samotný vývoj webovej aplikácie bude prebiehať nasledovne. Na začiatok bude potrebné vytvoriť grafické predlohy ako bude stránka vyzerať. Tieto predlohy budú konzultované s vedúcim práce a prípadné nedostatky budú do predlohy zapracované. V momente keď už budú predlohy hotové vytvorím klikateľný prototyp v ktorom budú testovacie dáta. V tomto bode máme fungujúceho klienta teraz je potrebné napojiť ho na server to znamená vytvoriť databázu a naplniť ju testovacími dátami. Klienta je potrebné prepísať aby bral testovacie dáta už z databázy. Máme aj klienta, server aj databázu teraz je potrebné toto všetko sprístupniť používateľom to znamená dostať webovú aplikáciu do cloudu. Sprístupnená aplikácia bude teraz testovaná používateľmi a každá chyba nájdená na produkcii bude opravená. Používatelia budú pridávať do databázy už reálne dáta. Bude potrebné vytvoriť aj užívateľskú príručku.

Nakoniec bude potrebné celý vývoj a testovanie spísať do diplomovej práce.

5 Súvisiace práce

Stránka schema.org [9] slúži na vytváranie schém pre štrukturované dáta. Stránka bola založená spoločnosťami Google, Microsoft a Yahoo. Z tejto stránky sa môžeme inšpirovať tým ako zobrazíme používateľovi už existujúci typ anotovaného stĺpca s vysvetlením a ukázanými vzťahmi tohto typu. Taktiež sa môžme inspirovať štruktúrou prípadne použiť už existujúce typy aby sme nemali na začiatku našu bázu znalostí prázdnu.

Stránka kaggle [10] slúži na sťahovanie najrôznejších datasetov pre trénovanie machine learning modelov. Z tejto stránky sa môžme taktiež inspirovať nájdeme tu dataset na stiahnutie, popis tohto datasetu o tom čo sa v ňom nachádza. Konkrétnejšie vidíme aj štatistiky to znamená koľko je v stĺpci unikátnych hodnôt, koľko je tam hodnôt celkovo, aké sú tam stĺpce a o čom nám každý stĺpec hovorí a aký typ majú. Pod datasetom môžeme vidieť metadáta ktoré nám hovoria kto dataset pridal odkiaľ je a podobne. Vidíme aj štatistiky koľko ľudí si tento dataset pozrelo, stiahlo.

Nasledujúca práca sa zaoberá tým ako vytvoriť plnohodnotnú schému ak údaje chýbajú alebo sú definované len čiastočne [3].

6 Výskumné otázky

- Ktorý typ databázy je pre náš projekt najvhodnejší, je to relačná, nerelačná alebo grafová ?
- Aký je najlepší programovací jazyk pre klienta ?
- Aké výhody prináša použitie serverless prístupu oproti serveru ?
- Čo je najlepší formát na prenos datasetov z endpointu k používateľovi ?

7 Záver

V sekcii Analýza a pohľad do problematiky som priblížil problematiku ktorej sa plánujem venovať v rámci mojej diplomovej práce. Analyzoval som možné spôsoby riešenia môjho problému pričom som každú z nich aj v krátkosti opísal. Vybraním toho správneho spôsobu na riešenie každého daného problému sa budem konkrétnejšie zaoberať v DP1.

V sekcii cieľ práce som si vyšpecifikoval cieľ práce. Pri písaní tohto výskumného zámeru sa vyskytli aj určité výskumné otázky ktorým som venoval vlastnú kapitolu. Úlohou mojej práce bude všetky tieto otázky zodpovedať a moje finálne tvrdenia a výber technológií podložiť vierohodnými dátami a zdrojmi.

Zdroje

[1] Atlassian, *What is a knowledge base?: Atlassian*, Apr. 2020. [Online]. Available: https://www.atlassian.com/itsm/knowledge-management/what-is-a-knowledge-base.

[2] D. H. Curie, J. Jaison, J. Yadav, and J. R. Fiona, "Analysis on web frameworks," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1362, 2019, p. 012 114.

[3] K. Kellou-Menouer, N. Kardoulakis, G. Troullinou, Z. Kedad, D. Plexousakis, and H. Kondylakis, "A survey on semantic schema discovery," *The VLDB Journal*, vol. 31, no. 4, pp. 675–710, 2022.

[4] A. Pavlo and M. Aslett, "What's really new with newsql?" *ACM Sigmod Record*, vol. 45, no. 2, pp. 45–55, 2016.

[5] L. Reeve and H. Han, "Survey of semantic annotation platforms," in *Proceedings of the 2005 ACM Symposium on Applied Computing*, ser. SAC '05, Santa Fe, New Mexico: Association for Computing Machinery, 2005, pp. 1634–1638, ISBN: 1581139640. DOI: 10.1145/1066677.1067049. [Online]. Available: https://doi.org/10.1145/1066677.1067049.

[6] M. Roberts and J. Chapin, *What is Serverless?* O'Reilly Media, Incorporated, 2017.

[7] *Stack overflow developer survey 2022*, Jun. 2022.

[8] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data & knowledge engineering*, vol. 25, no. 1-2, pp. 161–197, 1998.

[9] *Welcome to schema.org*. [Online]. Available: https://schema.org/.

[10] *Your machine learning and data science community*, Feb. 2010. [Online]. Available: https://www.kaggle.com/.