Zaverecna sprava projektu

Projekt : Znackovani slov

Katarina Kejstova < 433 820 >
Pavel Kucera < 422 626 >
Simon Struk < 422 522 >
Jiri Vahala < 422 802 >

Zaverecna sprava projektu : Projekt : Znackovani slov

Katarina Kejstova , Pavel Kucera , Simon Struk a Jiri Vahala

Datum vydání 2016-06-07

Obsah

Kapitola 1. Popis projektu

Zadanie

Pre mnohe ulohy strojoveho ucenia je potrebne ziskat oznackovane data. Vasou ulohou bude vytvorit system, ktory umozni ich znackovanie. Anotator odpoveda len na: otazka - odpoved - hodnotenie (vlastna akcia). Aby sa zabranilo typickym problemom pri znackovani je potrebne implementovat niekolko akcii, ktore napr. pridavaju sum, opakuju otazky, ... Na zaklade vysledkov, by mali byt prezentovane zakladne statistiky.

Kapitola 2. Rozdelenie uloh

Pavel

vytvorenie databaze

spracovanie dat v csv-formate

Jirka

vytvorenie statistik na zaklade ziskanych dat

vybrat vhodne statistiky

Simon

Java backend

JSP server

spracovavanie poziadavkov

Katarina

html frontend, pomocou bootstrap

spracovanie wiki projektu

Kapitola 3. Zaverecne spravy

Simon

V ramci mojho podielu na projekte som si nastudoval, ako sa vytvara webova aplikacia pomocou Java Server Pages a Java servletov. Nasledne som implementoval funkcionalitu zabezpecujucu prihlasovanie a odhlasovanie uzivatelov a ochranu pred neautorizovanym pristupom k privatnym castiam aplikacie. Mojou dalsou ulohou bolo vyriesenie nahravania vstupnych dat vo forme CSV suborov.

Pri prihlaseni uzivatela sa skontroluju predane prihlasovacie udaje voci databaze a pri platnych udajoch sa nastavi session atribut, ktory potom umoznuje pristup k privatnym castiam aplikacie. Tento atribut je kontrolovany filtrom a pri jeho neexistencii je uzivatel presmerovany na prihlasovaciu stranku. Pri odhlaseni uzivatela sa tento atribut zneplatni.

Tato cast aplikacie bola implementovana ako prva a posluzila mojim kolegom ako referencia pri implementovani dalsich servletov a JSP stranok.

Nahravanie vstupnych dat je riesene pomocou servletu, v ramci ktoreho sa skontroluje format CSV suboru a vytvoria sa prislusne baliky po 1000 slovach. Tieto baliky sa potom ulozia do databazy a su pripravene na dalsie spracovanie. Moj navrh bol potom mojimi kolegami prepracovany do jeho finalnej podoby.

Pavel

Navrh databaze

Aplikacni data byla potreba nekma ukladat a protoze jsem v tymu jediny s realnou zkusenosti pouzivani databazi, dobrovolne jsem se chopil vyberu databazove technologie a zpusobu, jakym data ukladat. Jelikoz jsem nikdy predtim nepracoval s zadnou NoSQL databazi, zacal jsem se poohlizet prave timto smerem, abych se naucil neco noveho.

Po pocatecnim hledani padla volba na MongoDB [https://www.mongodb.com/], dokumentovou databazi s nekterymi charektistikami relacnich technologii. MongoDB mi prisla jako dobry mezikrok, kde bych mohl uplatnit sve znalosti a zaroven se naucit praci v jinem paradigmatu. Pri navrhu entit samotnych jsem se snazil zohledit budouci pouziti, aby dotazy na data byly efektivni, pricemz jsem vysel ze schematu, jake bych navrhnul v relacni databazi. Pro zachovani efektivity jsem musel misty duplikovat data, coz by sice mela byt bezna praktika, ale stejne mi to nebylo uplne po chuti a velmi rad bych videl nejakou vetsi fungujici aplikaci nad NoSQL databazi, abych se z ni mohl poucit. Nerad znovu vymyslim kolo.

Navrzene schema slo, s mensimi upravami, pouzit pro veskerou zamyslenou funkcnost aplikace, coz beru jako pozitivum. Pri dalsim vyvoji bych se poohlizel po konzultaci s nekym zkusenejsim, abych dokazal urcit vykonostni dopady.

MVC

Aby aplikace zvladla i „trvale udrzitelny rust“ jsem se snazil veskery napsany kod uklizet do typickych kategorii model, view a controller. To trochu komplikoval fakt, ze jsme nepouzili zadny framework, ktery by podobne rozdeleni podporoval implicitne, ale po pridani malych komponent a nekterych knihoven, se nam podarilo aplikaci rozdelit.

Business logika je nyni kompletne oddelena a k predavani zavislosti uouziva (skoro) vyhradne konstruktory, takze je i lehce prenositelna. Servlety predstavuji jednotlive controllery a jsp soubory pouzivame jako

sablony. V pripadech stranek bez jakekoli logiky jsme jsp soubory pouzili i bez vytvoreni odpovidajiciho servletu.

Katarina

Mojou ulohou bolo vytvorit frontend pre nasu aplikaciu. Kedze bola poziadavka na responzivnu aplikaciu, zvolili sme framework Bootstrap (http://getbootstrap.com/), popularny HTML, CSS a JS framework na vyvoj rezponzivnych mobilnych projektov na webe.

Cely navrh je vo jsp formate, a teda HTML + CSS, vyuzivajuce prvky javy. Vsetka funkcionalita je teda zabezpecena funkciami a servletom, pisanym v jave. Kazdy request je teda najprv spracovany servletom, a nasledne presmerovany na jsp stranku. Sevrlety zabezpecuju ako prihlasovanie a odhlasovanie, tak aj komunikaciu a zapisovanie do databazy, a taktiez vypis dostupnych balickov a nasledne znackovanie slov, ktore som riesila ja.

Navrh som sa pokusala urobit co najviac uzivatelsky privetivy, a jednoducho pochopitelny. Po spusteni aplikacie bolo kazdemu uzivatelovi umoznene prihlasit sa, alebo registrovat. Po naslednom prihlaseni uzivatel obdrzal kratku spravu, vysvetlujucu, co moze robit, a kde co najde.

Menu je situovane v hornej casti, a pocas celej doby, s vynimkou samotneho znackovania, sa nemeni. Ponuka moznosti znackovania, a teda vypis aktualne dostupnych balickov, export roznych statistik, nahranie noveho balicka slov, a odhlasenie.

Samotna znackovacia cast je urobena tak, aby bola pohodlne obsluzitelna aj na tablete, a teda obsahuje velku plochu na odpovedanie "ano" vpravo zelenej farby, a "nie" vlavo cervenej farby. Odpoved je zaznamenana kliknutim na danu plochu.

Wiki stranku projektu sme spracovali priebezne, obsahuje ako rozdelenie uloh, tak dokumentaciu vsetkych casti : databaze, struktury, prihlasovanie a autentizacia, forma odpovedania, a nasledne spracovanie, a teda popis sumu a statistik.

Jiri

Mym ukolem bylo zpracovani nashromazdenych dat v databazi a jejich transformace do citelne podoby.

Nashromasdena data slouzi hlavne pro vypocitani statistik nad ruznymi uzivateli, ale i baliky slov nebo jednotlivymi slovy. Takova data mohou byt dale ruzne interpretovana a vyhodnocovana spravcem aplikace.

Jelikoz se jedna o aplikaci anotujici data, neni mozne pocitat statistiky nad vsim stejne. Proto se rozlisuji dve hlasovani uzivatelu. Prvni typ hlasovani probiha nad slovy, u kterych neni znamy jejich typ. V takovem pripade muzeme generovat statistiky trueRatio a duration nad jednotlivymi slovy a baliky. Druhe hlasovani je nad slovy, u kterych dopredu typ zname. Tato slova jsou pridana do baliku slov schvalne kvuli overovani kvality hlasovani jednotlivych uzivatelu.

Pokud nejaky uzivatel nema dostatecne kvalitni odpovedi na nami znamych slovech, je mozne jej z hlasovani vyradit nebo nebrat jeho hlasy v potaz. Pro pocitani uspesnosti jednoho uzivatele jsem zvolil vypocet confusion matrix a nad ni jeji zname metriky, zahrnujici: positive a negative F1-score, Recall, Precision, Random Agreement Probability, Cohen's Kappa coefficient, overall, average a mean classification accuracy.

Zalezi jen na spravci, jak se k danym statistikam postavi a ktere si vybere ke svym kriteriim.

Statistiky jsou ulozeny do formatu XML pres pouziti XStream knihovny. Pouziti je velice jednoduche a primocare.

Jednotlive dotazy do databaze jsou realizovany pouzitim MongoDB knihovny pro jazyk Java.

Do budoucna by se pro vyssi efektivitu daly statistiky implementovat do databaze, ktera by je on-the-fly prepocitavala. Tento krok by zajistil vyssi efektivitu. Ovsem ani v soucasne verzi neni s efektivitou zadny problem.