https://doi.org/10.53656/math2023-5-3-sys

Science in Education
Научно-методически статии

СИСТЕМА ЗА ИЗВЛИЧАНЕ И ВИЗУАЛИЗАЦИЯ НА ДАННИ ОТ ИНТЕРНЕТ

Гл. ас. д-р Георги Чолаков[1)], доц. д-р Емил Дойчев[1)], проф. д-р Светла Коева[2)]
[1)]*Пловдивски университет „Паисий Хилендарски“*
[2)]*Институт за български език – БАН*

Резюме. Статията представя система, която илюстрира динамично наличието на набори от данни (datasets) и езикови модели в областта на изкуствения интелект в големи хранилища като Hugging Face. Целта на създаването на подобна система е да се покаже, че освен за английски език за всички останали официални европейски езици наборите от данни и езиковите модели, необходими за разработки, които се базират на или използват езикови технологии и изкуствен интелект, имат или средно добра, или фрагментарна поддръжка. Едновременно с това описанието на архитектурата на системата запознава читателите с удобни за използване инструменти като Node-RED, MariaDB и Grafana, които предоставят широки възможности за приложение при решаването на различни задачи, включващи обхождане и събиране на данни от интернет, съхранение на информация в база от данни и визуализация на данните по ясен и функционален начин. Всяка една от тези задачи, както и комбинацията им могат да се прилагат при изпълнението на ученически проекти в гимназиален етап на обучение.

Ключови думи: автоматично извличане на данни; визуализация на данни; набори от езикови данни; езикови модели

1. Увод

Статията представя система, която илюстрира динамично наличието на набори от данни (datasets) и езикови модели за различни езици в областта на изкуствения интелект в големи хранилища като Hugging Face[1].

Системата сравнява на базата на периодична актуализация на данните броя на ресурсите (набори от данни и езикови модели), които са показателни за наличието или за потенциала за разработване на големи езикови модели (Large Language Models, LLMs), следователно за напредъка в областта на изкуствения интелект (Artificial Intelligence, AI). Езиците, за които се прави сравнението, са 24-те официални езика на Европейския съюз.

Системата има следната архитектура: с помощта на Ноуд-РЕД (Node-RED[2]) се обхожда уебинтерфейсът на избраната платформа; извлечените

данни се съхраняват в релационната база от данни „МариаДиБи“ (MariaDB[3]); за визуализация на данните се използва приложението „Графана“ (Grafana[4]), което позволява да се изграждат динамични във времето графики въз основа на събраните данни и дефинирани предпочитания за търсене.

Целта на създаването на подобна система е да се покаже, че освен за английски език за всички останали официални европейски езици наборите от данни и езиковите модели, необходими за разработки, които се базират на или използват езикови технологии и изкуствен интелект, имат или средно добра поддръжка (какъвто е случаят с езици като френски, немски и испански), или фрагментарна поддръжка (за езици като български, словашки, хърватски и др.) (Giagkou et al. 2023, pp. 79 – 83). Ако се проследят съществуващите проучвания за технологичния напредък на езиковите технологии за български (Blagoeva et al. 2012; (Koeva & Stefanova 2022), няма как да не се забележи съществения прогрес, но такъв прогрес е налице и за останалите европейски езици. Ако в сравнителния анализ през 2012 година езиковите технологии за български са били оценени на 15-о място по отношение на останалите европейски езици, то през 2022 година мястото е 22-ро (Koeva 2023, pp. 105). Предложена е метрика на цифрово езиково равенство (Digital Language Equality, DLE): състоянието, при което всички езици имат технологичната поддръжка, която е необходима, за да продължат да съществуват и да се развиват като живи езици в дигиталната епоха (Gaspari et al. 2021, 2022). Метриката се изчислява за всеки отделен език на базата на различни фактори: технологични и контекстуални (Gaspari et al. 2022, pp. 6 – 14). За сравнение, новото при системата, която се представя, е, че данните за технологичния прогрес се събират динамично посредством обхождане на уебинтерфейса на хранилището „Хъгинг Фейс“, а самото хранилище (по наше мнение) е най-бързо развиващата се система за съхранение и разпространение на набори от данни и езикови модели с приложение в изкуствения интелект.

Едновременно с това описанието на архитектурата на системата ще запознае читателите с удобни за използване инструменти, които предоставят широки възможности за приложение при решаването на различни задачи, които включват обхождане и събиране на данни от интернет, съхранение на информация в база от данни и визуализация на данните по ясен и функционален начин.

2. Хранилище на езикови и софтуерни ресурси

„Хъгинг Фейс“ съдържа базирани на Git хранилища с функции, подобни на „ГитХъб“ (GitHub[5]); модели с базиран на Git контрол на версиите; масиви от данни и уебприложения, предназначени за демонстрации на приложения

за машинно обучение. Компанията „Хъгинг Фейс“ (заедно с различни изследователски групи) организира над 1000 изследователи от цял свят, които създават през 2022 година BLOOM[6] – многоезиков голям модел със 176 милиарда параметри.

Цел на нашето изследване е обхождането на хранилището и събиране на информация за публикуваните езикови модели и езикови набори от данни, които се увеличават динамично. Самите модели и набори от данни не се изтеглят, а само метаданните, които ги характеризират: име на ресурса; линк за изтегляне; предназначение (например категоризация на документи, автоматично извличане на резюмета на текст и пр.); размер в единиците, в които авторите са решили да измерват ресурса си (брой думи, брой изречения, брой изображения, брой анотации, обем в гигабайти и др.); езиците, за които се отнасят наборите от данни и моделите; лицензите, с които се разпространяват.

3. Общо описание на използваните технологии

Общата архитектура на софтуерното решение е визуализирана на фигура 1, като по-подробно описание на компонентите е представено след фигурата.

Фигура 1. Обща архитектура на системата

3.1. Извличане на данните

При извличането на метаданни се използва инструментът „Ноуд-РЕД“ (Node-RED), технологично базиран на Node.js[7] и предоставящ начин на работа с потоци, чиято идеология е да улесни използването на API (Application Programming Interface, програмно-приложен интерфейс) – често, без да се налага програмиране на по-ниско ниво, а посредством визуално дефиниране на елементите, съставящи работния поток. Така лесно могат да бъдат обхождани уебстраници с REST (Representational State Transfer[8]) заявки и обработка на резултатите. В нашия случай използването на избрания инструмент позволява обхождане на метаданни по автоматизиран начин,

подобно на обхождането на интернет съдържание, но без да се създава програмен код, с минимално програмиране на JavaScript. Използва се и възможността на инструмента да съхранява резултати в база от данни, от която на по-късен етап могат да бъдат извличани анализи и статистика.

„Ноуд-РЕД“ може да се инсталира на домашен компютър, сървър или да се намира в облак. Инструментът използва поточно базирано програмиране (event-driven, flow-based), при което кодът се активира в зависимост от това какво се съдържа в потока (обработваната информация). Такъв вид програмиране не изисква виртуален сървър, на който да се инсталира системата.

3.2. Съхраняване на данните

Извлечените данни се съхраняват в релационна база от данни „МариаДиБи“ (MariaDB), която е сред най-популярните сред тези с отворен код. Използваната версия е 10.7.8, работеща на операционна система „Линукс Убунту“ (Ubuntu) 20.04, но има възможност за инсталация и под „Уиндоус“ и „МакОС“. Написана е на езиците C, C++, Perl, Bash и предлага интерфейс към най-популярните програмни езици към момента[9]. За заявки се използва езикът SQL, предназначен за създаване, извличане и обработка на данни в релационни системи за управление на бази от данни.

„МариаДиБи“ е подходяща за работа с големи масиви и е с подобрена ефективност при извличането на данни, което е важно за представената система, защото извличането на данните се извършва посредством обединения и сечения на относително големи обеми от данни. „МариаДиБи“ предоставя и възможност за едновременна и конкурентна работа на висок брой от връзки към нея, което би било важно за бъдещото развитие на системата и разгръщането ѝ в разпределена облачна среда. В момента данните, съхранявани в базата, са в структуриран вид, който е подходящ за релационните бази от данни, но в процеса на усъвършенстване на системата е вероятно да се наложи и съхранение на неструктурирани данни, което също се поддържа от „МариаДиБи“.

3.3. Визуализация на данните

За визуализация на данните в удобен за крайния потребител вид се използва системата „Графана“ (Grafana), чрез която могат да бъдат създавани графики, базирани на извлечените от базата от данни. „Графана“ е многоплатформена система с отворен код за интерактивно визуализиране на аналитични данни. Тя предоставя възможност за създаване на комплексна визуализация със сложни заявки към съответната база от данни, което я прави подходяща за потребители с познания относно структурата на съхраняваните

данни и логиката на техните взаимоотношения – това позволява създаването на нови графики от самите потребители, ако имат съответните права. От друга страна, графиките дават ясна представа за събраната информация и правят сравнението на резултатите обозримо за потребители, които може да не са технически компетентни.

„Графана“ позволява да се правят заявки, визуализация и да се изследват различни показатели, регистрационни файлове и проследявания, независимо къде се съхраняват. „Графана“ предоставя инструменти за превръщане на данните от база от данни с времеви редове (time series) в графики с подходяща визуализация. Данните могат да се изследват и представят по различен начин, като се сравняват едновременно различни времеви диапазони, заявки и източници на данни. Могат да се задават шаблони за визуализация с променливи стойности, които могат да се използват повторно за много различни случаи на употреба.

„Графана“ поддържа различни методи за удостоверяване като LDAP (Lightweight Directory Access Protocol) – протокол за достъп до онлайн справочни услуги, и позволява да се свързват потребители с организации, като се задават различни разрешения за достъп до папки и табла за управление, както и до източника на данни, ако е част от „Графана“[10].

4. Извличане на данни от интернет

За извличане на метаданни от хранилището „Хъгинг Фейс“ в системата „Ноуд-РЕД“ са създадени работни потоци:

– за набори от данни – за извличане на метаданни за наборите от езикови данни (фигура 2);

– за модели – за извличане на метаданни за езикови модели (фигура 3).

Отделните възли на работните потоци комуникират чрез съобщение, предавано към всеки следващ възел, съдържащо множество атрибути, в които се намират както нужните за съхранение данни, така и системни данни.

4.1. Извличане на езикови ресурси: модели и набори от данни

Процесите на обработка в двата работни потока са сходни, различават се основно във възлите за обработка на данните с JavaScript, където се прави подготовка за добавянето им в базата данни – тази разлика е заради специфичната структура на извлечените от хранилището данни в двата случая. Поради това описанието е представено само един път, като само различните възли при извличането на наборите от езикови данни са описани допълнително. Процесът започва с итерации върху всеки от заложените езици, към момента – официалните езици на Европейския съюз, представляващи интерес за проучването. Фиг. 1 представя работния процес

за извличане на метаданни за езикови модели. По-долу следва кратко описание на възлите:

● Prepare languages – функционален възел, дефиниращ масива с езиците, чиито ресурси ще бъдат обхождани.

● Loop languages – цикъл, изпълняващ итерациите върху масива с езиците. При приключването му работният поток завършва своята работа.

Фигура 2. Работен поток за извличане на модели

• Set language – този възел поставя кода на текущия за итерацията език в атрибута lang на съобщението, което ще бъде пратено към следващия възел. За код на езиците се използват кодовете от стандарта ISO 639[11].

• Get language – изпълнява HTTP GET[12] заявка към „Хъгинг Фейс“ с параметър езика от текущата итерация.

• Parse HTML – трансформира получения резултат от заявката HTML/JSON до обект.

• Loop models – реализира цикъл за обхождане на всички ресурси от типа модел от страницата. След приключването му управлението се връща към съдържащия го цикъл (Loop pages) за преминаване към следващата уебстраница с модели.

• Prepare page URL – функционален възел, формиращ URL адреса на всяка страница от итерациите.

• Get next page – изпълнява HTTP GET заявка, извличаща съдържанието на текущата страница за итерацията.

• Prepare model URL – формира URL адреса на конкретния ресурс от обхожданите ресурси от текущата страница.

• Loop models – реализира цикъл за обхождане на всички ресурси от типа *модел* от страницата. След приключването му управлението се връща към съдържащия го цикъл (Loop pages) за преминаване към следващата страница с модели.

• Prepare model URL – формира URL адреса на конкретния ресурс от обхожданите ресурси от текущата страница.

• Get model – изпълнява HTTP GET заявка, извличаща HTML страницата на ресурса. В следващия възел този резултат ще бъде анализиран и подготвен за обработка.

• Set msg.datasetResponse – поставя обекта datasetResponse, върнат от предходния възел, в съобщението към следващия възел в потока. Така за обработка остава само този обект, останалите се игнорират, защото не представляват интерес.

• div.SVELTE_HYDRATER.content – извлича от страницата съдържанието на div елемент[13], в който са търсените метаданни за ресурса.

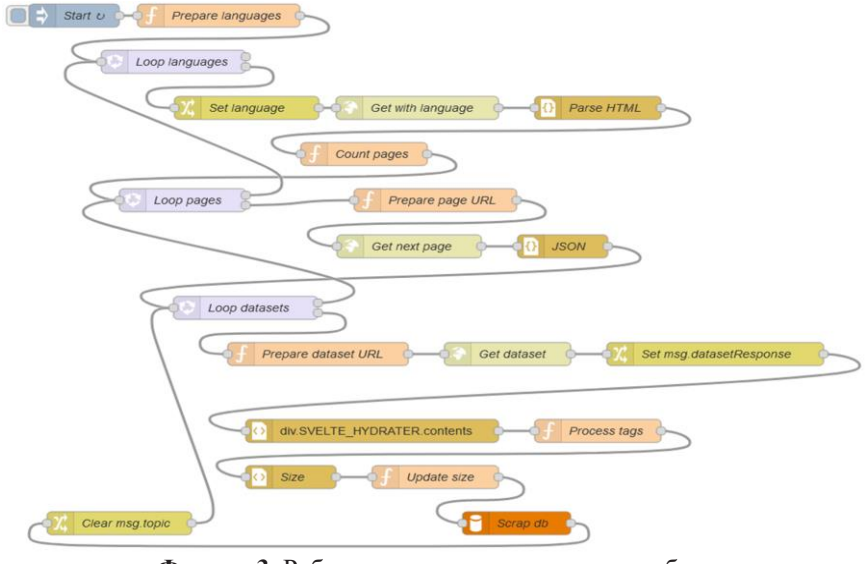
• Process tags – в този функционален елемент се обработват с JavaScript извлечените като обект метаданни и се формират SQL командите за въвеждане на данните като низове, които ще бъдат предадени на съответния възел за изпълнението им към базата от данни.

• Scrap DB – този възел служи за изпълнение на формираните SQL команди към базата от данни, след което се преминава към следваща итерация от обхождането на ресурсите на текущата страница.

• Clear msg.topic – премахва съдържанието на атрибута *topic* от съобщението. Команди, които вече са се изпълнили в базата от данни, няма нужда да се повтарят.

След това се преминава към итерацията за следващия модел.

При извличането от типа *набор от данни* се използва сходен работен поток, който е представен на фигура 3. Разлика има при извличането на размера на ресурса (ако такъв е въведен от авторите му). Допълнителните възли са



Фигура 3. Работен поток за извличане на набори от данни

• Size – от атрибута datasetResponse на съобщението (съдържащ HTML) се извлича HTML елементът, който може да съдържа данни за размера на ресурса, генерирайки обект от него, чийто атрибути са лесни за обработка.

• Update size – проверява дали са налични данни за размер на ресурса в обекта, трансформиран от HTML от горния възел, и ако такива има, генерира SQL команда за актуализиране размера на ресурса в съответната таблица в базата от данни.

4.2. Схема на базата от данни

Данните се съхраняват в релационна база от данни „МариаДиБи“ (MariaDB). Схемата представлява проста версия на аналитична база от данни от типа времеви серии. Идеята е да се съхраняват данни от различни моменти (дати) във времето, които да могат да бъдат сравнявани след това. На фигура 4 е представена схемата на базата от данни за съхраняване на данните от типа *Набор от данни*/*Модел*.

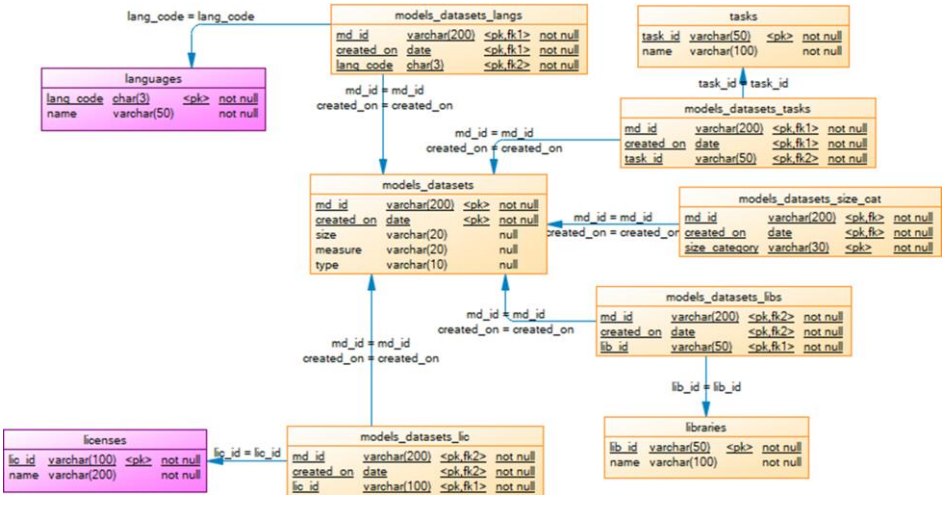
Разпределението на данните в таблиците е, както следва.

• Languages – номенклатурна таблица, съдържаща данни за езиците, използвани в ресурсите. Атрибутът lang_code съдържа код на езика по стандарта ISO 639; name – съдържа наименованието на езика; created_on – дата на въвеждане на данните в таблицата, на практика това е датата на извличането им от източника, като значението на този атрибут е същото и в

другите таблици, където се среща. Чрез него се отделят времевите периоди за сравнение на извлечените ресурси.

• Licenses – номенклатурна таблица, съдържаща данни за лицензите, с които се разпространяват езиковите ресурси и моделите.

• Models_datasets – съдържа данни за ресурсите от типа *Набор от данни*/*Модел*. Атрибутът type съдържа низ "dataset" или "model", съответно за различните ресурси; size – размер на ресурса; measure – мерна единица на размера.



Фигура 4. Таблици, съхраняващи метаданни за ресурси от типа *Набор от данни*/*Модел*

• Models_datasets_langs – представя взаимоотношение от тип много-към-много между *Набори от данни*/*Модели* и езици – за всеки език може да има множество ресурси, както и всеки ресурс може да бъде за множество езици.

• Models_datasets_lic – представя взаимоотношение от тип много-към-много между *Набори от данни*/*Модели* и лицензи – съдържа данни за лицензите, под които се разпространява ресурсът, както и за това кой лиценз в кои ресурси се използва.

• Models_datasets_size_cat – съдържа данни за категориите, в които се измерват ресурсите (тъй като категориите са много на брой и не са съотносими една към друга, тези данни не се визуализират, макар че се пазят в базата от данни).

• Tasks – номенклатурна таблица, която съдържа данни за категориите, които показват предназначението на ресурсите.

• Models_datasets_tasks – съдържа данни за предназначението (tasks) на ресурсите.

• Libraries – номенклатурна таблица, съдържаща видовете библиотеки (libraries), използвани в ресурсите.

• Models_datasets_libs – съдържа данни за библиотеките, използвани в ресурсите.

При евентуално разрастване на системата данните може да се съхраняват и в неструктуриран вид, което ще доведе до изискването за използване и на NoSQL база от данни, като подходяща за целта би била документно ориентирана база от данни.

5. Представяне на данните за потребителите

Статистиките в „Графана“ (Grafana) са представени с графика, която отразява данните спрямо стойностите на параметрите от филтъра в горната лява част на страницата.

Филтърът се състои от следните параметри.

• *Източник на данните* (*Source*). Към момента този параметър може да приема стойността *Hugging Face*. Може да се добави и друг източник и в такъв случай да бъде избирана само едната или повече стойности. Ако се изберат повече източници, графиката ще показва сборната информация.

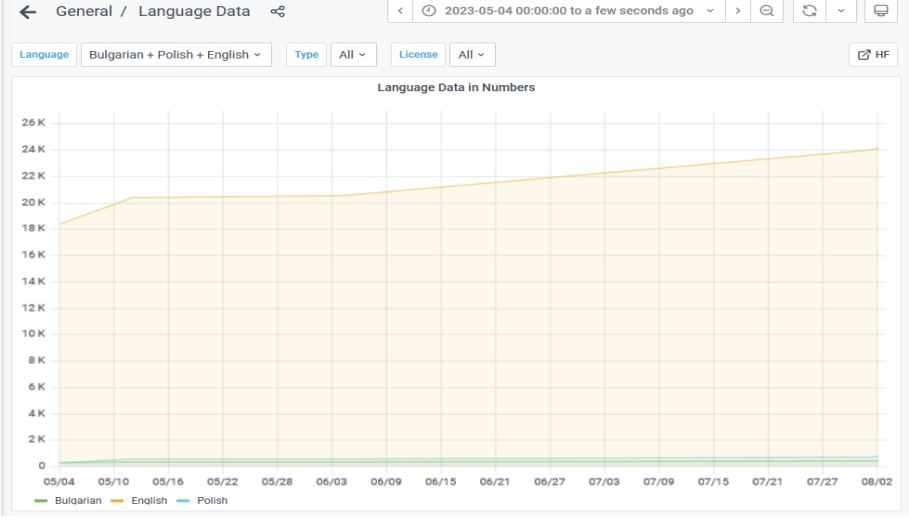
• *Език* (*Language*). Параметърът определя набора от езици, данните за които ще бъдат включени в статистиките. Възможно е да се избира един или повече от един език. При избор на повече от един език графиката показва информацията за данните с различни цветове. Под всяка графика има легенда с цветовото съответствие.

• *Тип* (*Type*). Тип на ресурса (набор от данни или модел), който се визуализира на графиката. Възможно е да бъдат избрани повече от един тип.

• *Лиценз* (*License*). С този параметър се филтрират допълнително езиковите ресурси по типа на лиценза, с който са достъпни. Може да се избира повече от един вид лиценз.

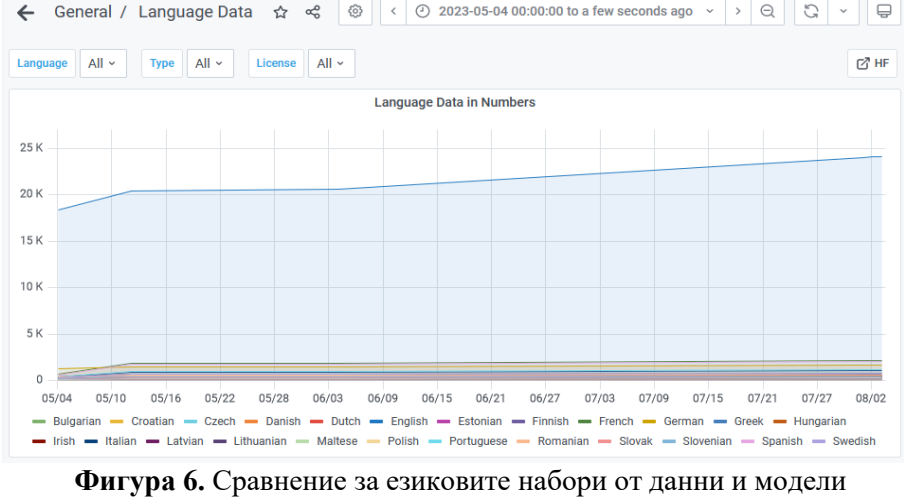
В горната дясна част на страницата има допълнителен филтър, който е свързан с времевия диапазон на данните, участващи в статистиките. Диапазонът от време може да се задава както с твърди стойности за начало и край, така и с предефинирани периоди за последните часове.

Въпреки че времевият диапазон може да се задава с точност до секунда, това не се препоръчва при създаването на системата и в конкретния случай е предвидено данните да се събират веднъж месечно. Диапазон от поне два месеца може да се приложи, след като минат няколко месеца от пускането на системата в експлоатация, за да се съберат достатъчно данни.



Фигура 5. Сравнение за езиковите набори от данни и модели за български, английски и полски език

Фигура 5 показва графика със следните параметри: избрано е сравнение между български, английски и полски за времеви диапазон от 4.5.2023 г. до 2.8.2023 г. Сравняват се едновременно езиковите набори от данни и модели, независимо с какъв лиценз се разпространяват. Ясно се вижда разликата между ресурсите за английски език и останалите два езика.



Фигура 6. Сравнение за езиковите набори от данни и модели

за 24 официални европейски езика

Фигура 6 показва сравнение между официалните езици на Европейския съюз, а времевият диапазон е от 4.5.2023 г. до 2.8.2023 г. Сравняват се едновременно езиковите набори от данни и модели, независимо с какъв лиценз се разпространяват. Тук също може да се види ясно доминацията на ресурсите за английски език спрямо останалите езици.

Сравнителните данни потвърждават направените през 2012 и 2022 година заключения, че само за някои европейски езици има технологичен напредък в областта на езиковите технологии. В случая, тъй като предназначението на хранилището „Хъгинг Фейс“ е основно за набори от данни и модели, свързани с изкуствен интелект, заключението може да се пренесе в тази насока. Системата е достъпна на адрес; https://dcl.bas.bg/aid2030/grafana/.

6. Заключение

Представената система илюстрира начин за събиране и визуализация на данни от интернет, в който са комбинирани достъпни и добре развити средства като „Ноуд-РЕД“, „МариаДиБи“, „Графана“. Представеният метод може да бъде приложен за различни задачи, целящи събирането и визуализацията на различен тип данни. В конкретния случай предназначението на системата е да обхожда и събира информация от хранилище за набори от данни и модели, които се използват основно за приложения в областта на изкуствения интелект. По този начин се илюстрира напредъкът в създаването на езикови модели и езикови набори от данни за претрениране и фокусирано трениране на големи езикови модели. Допълнително предимство на представената разработка е, че всеки един от използваните инструменти може да прилага за разнообразен тип задачи, а комбинацията им може да се използва при работата по ученически проекти в гимназиална форма на обучение.

БЕЛЕЖКИ

1. https://huggingface.co [прегледан на 11.9.2023]
2. https://nodered.org [прегледан на 7.7.2023]
3. https://mariadb.org [прегледан на 17.6.2023]
4. https://grafana.com [прегледан на 1.8.2023]
5. https://github.com [прегледан на 10.9.2023]
6. https://huggingface.co/bigscience/bloom [прегледан на 21.6.2023]
7. https://nodejs.org [прегледан на 1.6.2023]
8. Софтуерна архитектура за реализация на уебуслуги.

9. https://mariadb.com/kb/en/connectors/ [прегледан на 15.4.2023]
10. https://grafana.com/docs/grafana/latest [прегледан на 12.9.2023]
11. https://www.iso.org/iso-639-language-codes.html [прегледан на 27.5.2023]
12. Методът GET се използва за извличане на данни от даден сървър с помощта на дадено URL.
13. Търсеният div елемент има стил с идентификатор SVELTE_HYDRATER, оттам идва и странното наименование на възела.

REFERENCES

BLAGOEVA, D., KOEVA, S., MURDAROV, V., 2012. The Bulgarian Language in the Digital Age. *META-NET White Paper Series: Europe's Languages in the Digital Age.* Heidelberg etc.: Springer. http://www.meta-net.eu/whitepapers/volumes/bulgarian.

GASPARI, F., GRÜTZNER-ZAHN, A., REHM, G., GALLAGHER, O., GIAGKOU, M., PIPERIDIS, S., WAY, A., RIGAU, G., HAJIČ, J., 2022. *Deliverable d1.1 digital language equality (full specification)*. Project deliverable; EU project European Language Equality (ELE); Grant Agreement no. LC-01641480 – 101018166 ELE. https://european-language-equality.eu/wp-content/uploads/2022/03/ ELE_Deliverable_D1_3.pdf

GASPARI, F., WAY, A., DUNNE, J., REHM, G., PIPERIDIS, S., GIAGKOU, M., 2021. *Deliverable D1.1 Digital Language Equality (preliminary definition)*. https://european-language-equality.eu/wp-content/uploads/2021/05/ELE_ Deliverable_D1_1.pdf. Project deliverable; EU project European Language Equality (ELE); Grant Agreement no. LC-01641480 – 101018166 ELE.

GIAGKOU, M., LYNN, T., DUNNE, J., PIPERIDIS, S., REHM, G., 2023. European Language Technology in 2022/2023. In: *Rehm*, *G., Way*, A. (eds) *European Language Equality. Cognitive Technologies. Springer, Cham,* pp. 75 – 94. https://doi.org/10.1007/978-3-031-28819-7_4

KOEVA, S., 2023. Language Report Bulgarian. In: *Rehm*, *G., Way, A.* (eds) *European Language Equality. Cognitive Technologies. Springer, Cham*, pp. 103 – 106. https://doi.org/10.1007/978-3-031-28819-7_7

KOEVA, S., STEFANOVA, V., 2022. *Deliverable D1.5 Report on the Bulgarian Language. European Language Equality (ELE)*; EU project no. LC-01641480 – 101018166. https://european-language-equality.eu/reports/language-report-bulgarian.pdf.

SYSTEM FOR RETRIEVAL AND VISUALIZATION OF DATA FROM THE INTERNET

Abstract. The article presents a system that dynamically displays the availability of language datasets and language models found in large repositories such as Hugging Face. The goal of developing such a system is to demonstrate that, outside of English, the datasets and language models required for advancements based on or utilizing language technologies and artificial intelligence have either moderate or fragmented support. At the same time, the description of the system architecture introduces readers to easy-to-use instruments such as Node-RED, MariaDB, and Grafana, which offer a wide range of application opportunities in solving various tasks like crawling and collecting data from the Internet, storing information in a database, and data visualization in a clear and functional manner. Each of these tasks, as well as their combinations, can be used to carry out student projects at the senior high school level of education.

Keywords: automatic data collection; data visualization; language datasets; language models

✉ **Dr. Georgi Cholakov, Assist. Prof.**
ORCID iD: 0000-0001-7971-8434
Faculty of Mathematics and Informatics
Plovdiv University "Paisii Hilendarski", Plovdiv, Bulgaria
E-mail: gcholakov@uni-plovdiv.bg

✉ **Dr. Emil Doychev, Assoc. Prof.**
ORCID iD: 0000-0002-0306-0311
Faculty of Mathematics and Informatics
Plovdiv University "Paisii Hilendarski", Plovdiv, Bulgaria
E-mail: e.doychev@uni-plovdiv.bg

✉ **Prof. Dr. Svetla Koeva**
ORCID iD: 0000-0001-5947-8736
Department of Computational Linguistics
Institute for Bulgarian Language
Bulgarian Academy of Sciences
Sofia, Bulgaria
E-mail: svetla@dcl.bas.bg