https://doi.org/10.53656/math2024-1-1-cha

Science in Education
Научно-методически статии

ПРЕДИЗВИКАТЕЛСТВА ПРИ ОБХОЖДАНЕТО НА ИНТЕРНЕТ С ЦЕЛ ИЗВЛИЧАНЕ НА ДАННИ

Гл. ас. д-р Георги Чолаков[1)], доц. д-р Емил Дойчев[1)] ,
проф. д-р Светла Коева[2)]

[1)] Факултет по математика и информатика – ПУ „Паисий Хилендарски“ (България)
[2)] Институт за български език „Проф. Любомир Андрейчин“ – БАН (България)

Резюме. Статията представя предизвикателствата при реализацията на Система за извличане и визуализация на данни от интернет посредством обхождането на езикови ресурси от хранилището Hugging Face и извличането на данни, свързани с тях. Данните в системата периодично се обновяват, за да се проследява динамиката при създаването на езикови ресурси за различни периоди. Статията представя анализа на достъпните данни, тяхната структура и избрания начин за обхождане на страниците и извличане на данните. Споделеният опит при преодоляването на конкретните предизвикателства може да послужи при решаването на сходни проблеми, свързани с извличането на данни от интернет, задача, която често стои за разрешаване в различни проекти (включително ученически). Придобитият опит вследствие на описаната разработка показва, че подобен тип системи са силно зависими от имплементацията на обхождания източник на данни и при промяна в неговата структура на данни извличането трябва също да се актуализира, за да продължи да работи.

Ключови думи: уебобхождане; автоматично извличане на данни; набори от езикови данни

1. Увод

Статията представя предизвикателствата при реализацията на Система за извличане и визуализация на данни от интернет (Cholakov et al. 2023), по-точно обхождането на езикови ресурси от хранилището Hugging Face [1] и извличането на данни, свързани с тях. Крайната цел на системата е да визуализира в графичен вид наличните езикови ресурси за официалните европейски езици, използвани в областта на изкуствения интелект.

Метаданни за езиковите ресурси – набори от данни (datasets) и езикови модели (models), се извличат периодично (на конфигурируем интервал от време) от хранилището и се съхраняват в базата от данни на разработената система. Въз основа на броя на ресурсите (и на някои техни характеристики) се прави сравнение между официалните езици на Европейския съюз, демонстриращо наличие или потенциал за разработване на големи езикови модели, допринасящи за развитието в областта на изкуствения интелект.

Системата демонстрира относително голямата разлика между съществуващите набори от данни и езикови модели, които се базират на езикови технологии и се използват при създаването на приложения за изкуствен интелект, между английски език и останалите европейски езици, които имат средно добра (френски, немски, испански) или слаба технологична поддръжка (български, хърватски, словенски) (Giagkou et al. 2023, pp. 79 – 83).

Основен компонент от архитектурата на Системата е обхождането на хранилището Hugging Face и извличането на конкретните данни, които представляват интерес. Целта на настоящата статия е да представи: а) анализа на достъпните данни и тяхната структура; б) избрания начин за обхождане на страниците и извличане на данните. Споделеният опит при преодоляването на конкретните предизвикателства може да послужи при решаването на сходни проблеми, свързани с извличането на данни от интернет, задача, която в различни проекти (включително ученически) често стои за разрешаване.

2. Архитектура и използвани технологии в Системата за извличане и визуализация на данни от интернет

Различни проучвания показват съществен напредък в развитието на езиковите технологии както за български, така и за останалите европейски езици (Blagoeva et al. 2012; Koeva & Stefanova 2022). Предложени са методи за измерване на т.нар. цифрово езиково равенство – състоянието, в което за отделните (европейски) езици е налице достатъчна технологична поддръжка за решаването на предизвикателствата в дигиталната епоха (Gaspari et al. 2021, 2022). Различното при Системата за извличане и визуализация на данни от интернет (Cholakov et al. 2023) е, че данните се събират динамично чрез периодично обхождане на уебинтерфейса на хранилището Hugging Face, което е сред най-бързо развиващите се системи за съхранение и разпространение на набори от данни и езикови модели с приложение в изкуствения интелект.

Описанието на общата архитектура на системата, показано на фиг. 1, включва някои популярни софтуерни инструменти, предлагащи широ-

ки възможности за решаване на проблеми в областта на обхождането на данни в интернет, съхранението им в база от данни, както и визуализацията им в разбираем за потребителите формат. Следващите точки описват накратко компонентите от архитектурата.

2.1. Хранилището на данни Hugging Face

Hugging Face съдържа базирани на Git хранилища с функции, подобни на GitHub[2]; модели, с базиран на Git контрол на версиите; масиви от данни и уебприложения, предназначени за демонстрации на приложения за машинно обучение.

Фигура 1. Обща архитектура на системата

Целта на създадената система е обхождането на хранилището и събиране на метаданни за публикуваните езикови модели и езикови набори от данни, които се увеличават във времето. Сред тях са следните характеристики: 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, гигабайти и др.); езиците, за които се отнасят наборите от данни и моделите; лицензите, с които се разпространяват.

2.2. Инструмент за обхождане

За обхождане на хранилището Hugging Face се използва инструментът Node-RED[3], който е базиран на Node.js[4] и предоставя начин на работа, организиран с потоци, чиято идеология е да улесни работата с API (Application Programming Interface), дори без да се налага програмиране на по-ниско ниво, а посредством визуално дефиниране на елементите, съставящи работния поток. Чрез него лесно могат да бъдат обхождани уебстраници с REST (Representational State Transfer) заявки и резултатите от тях да бъдат обработвани спрямо целевата бизнес логика.

В описвания случай Node-RED позволява обхождане на метаданни по автоматизиран начин, подобно на обхождането на интернет съдържание, но без да се пише сложен програмен код, а с минимално програмиране на JavaScript. Събраните метаданни се съхраняват в базата от данни, от която на по-късен етап могат да бъдат извличани анализи и статистика.

2.3. База от данни

Извлечените данни се съхраняват в релационната БД MariaDB[5], версия 10.7.8, за операционната система Linux Ubuntu 20.04[6]. Сред аргументите за нейния избор е фактът, че тя е подходяща за работа с големи обеми от данни и е с подобрена ефективност при извличането на данни в сравнение с предшественика ѝ MySQL[7], което е важно за системата, защото извличането на данните се извършва посредством обединения и сечения на относително големи обеми от данни. Още един аргумент в подкрепа на избора ѝ е наличието на голяма общност от потребители, които я използват, което, от своя страна, улеснява намирането на решение при възникнали проблеми. От значение е също и възможността за съхранение на данни в неструктуриран вид, тъй като в процеса на усъвършенстване на системата е вероятно да се наложи и съхранение на неструктурирани данни.

2.4. Визуализация на данните

За визуализация на данните в графичен вид се използва системата Grafana[8], чрез която могат да се създават графики, базирани на извлечените данни. Grafana е предназначена за интерактивно визуализиране на аналитични данни и предоставя възможност за създаване на комплексни визуализации 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. В разглеждания случай данните, които се визуализират, са във вид на времеви серии (time series), показващи промяната на броя ресурси за избраните езици в различни периоди.

3. Реализация и предизвикателства, свързани с нея

При реализацията на системата първата стъпка беше да се намери начин за обхождане на ресурсите и събиране на метаданните за всеки

конкретен случай. За целта трябваше да бъдат дефинирани протокол за комуникация чрез заявки и отговори между разработваната система (играеща роля на клиент) и доставчика на данните (Hugging Face), както и самият формат на данните (идващи от интернет страниците на Hugging Face) – обичайно в процеса на софтуерна разработка това се договаря чрез дефиниция на API и документация, включваща заявки и връщани резултати. Но в разглеждания случай, тъй като разработката е едностранна, без участието на Hugging Face, беше необходимо да се направи подробен анализ на данните, които са налични в източника.

3.1. Анализ на данните

Анализът на данните има следните цели: а) да се дефинира комуникацията, чрез която разработваната система ще осигури достъп до данните; б) да се определи форматът на данните и по какъв начин ще бъдат разпознати и извлечени търсените стойности.

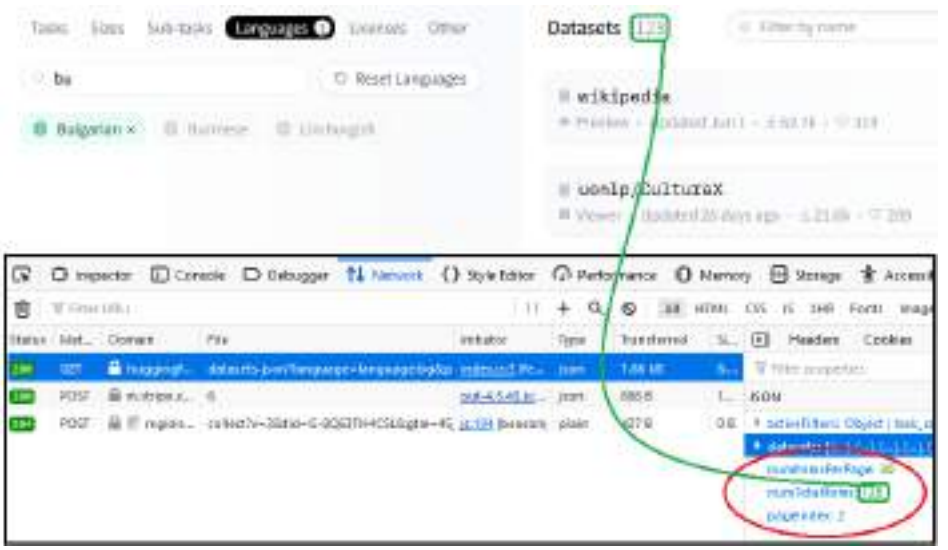
Достъп до данните

Липсата на официален интерфейс за извличане на данни от Hugging Face налага като единствен подход традиционното обхождане на съдържание на HTML страници, познато още като уебобхождане и използвано от всички водещи машини за търсене на информация в интернет (Google, Bing, и др.). Този подход е широко използван за най-различни цели (Abdurakhmonova and Ismailov 2022), включително и за хакерски атаки. Тук възниква първото от съществените предизвикателства – как да разграничим обхождането на разработваната система от останалите и то да работи така, че да не бъде маркирано автоматично (от анализиращ заявките софтуер на Хъгинг Фейс) или ръчно като опит за атака, например DoS (Denial-of-Service)[9] (Bergman and Popov 2023). Бяха разгледани различни методологии, архитектури (Najork 2017) и подходи (Sun et al. 2010), за да бъде подбран най-подходящият. При първоначалните тестове за извличане на данните, представляващи HTTP заявки, беше установено, че сървърът на Hugging Face се претоварва и спира да отговаря след около 4 минути. Добавено бе забавяне в цикъла за изпращане на заявките в работния поток в Node-RED – 1 секунда между заявките (установено емпирично) и по този начин първоначално извличанията бяха успешни. След няколко месеца се оказа, че процесът на извличане се прекъсва от Hugging Face, преди да са обходени половината от целевите ресурси. Към момента времетраенето между отделните HTTP заявки е увеличено на 5 секунди, което решава проблема с извличането, макар че с този интервал целият процес отнема около 2 дни.

Структура на данните

Липсата на официална документация наложи емпиричния подход за

анализ на формата на извлечените данни. За това от голяма помощ бяха вградените инструменти за разработка в браузърите Firefox и Chrom, както и Postman[10]. На фиг. 2 е илюстриран подходът за намиране на броя набори от данни (сходен е подходът и за езиковите модели) за български език в конзолата на браузъра Firefox. Извлечените данни служат за намиране на броя страници, които трябва да се обходят (по 30 ресурса на страница) – така се формира броят итерации, нужни за обхождане на всички страници. Това се повтаря за всеки от целевите езици, за които се събират данни, което формира най-външния цикъл с итерации. Параметризираният URL има вида
https://huggingface.co/datasets-json?language=
language:{{{lang}}}, където параметърът {{{lang}}} се заменя с код на езика – bg, en, de и т. н.



Фигура 2. Анализ на JSON за брой набори от данни

При извличането на ресурсите от всяка страница се използва отново HTTP GET заявка, която връща данни във вида, показан на фиг. 3. Анализът показа, че DIV елемент с клас SVELTE_HYDRATER съдържа атрибута data-props, в който са търсените метаданни за ресурса. Преобразуването на съдържанието на този атрибут в JSON значително улеснява обработката и извличането на данните. Предизвикателство се оказа фактът, че форматът на данните понякога се променя от разработчиците

– проблем, който беше констатиран още в ранен етап на разработването на системата. Метаданните, които се извличат, бяха преместени в поделемент tag_objs на атрибута data-props, което наложи отново анализиране на данните, за да бъде внесена корекцията.

Фигура 3. Анализ на атрибута data-props

Този проблем остана неразрешен – твърде сложно е да бъде създаден универсален алгоритъм за извличане на данните при всички възможни формати. Затова при забелязана промяна в Хъгинг Фейс, нашият код в Node-RED се коригира, за да продължи да извлича данните.

3.2. Обхождане на страниците и извличане на данните

Обхождането на страниците е реализирано с цикли в работните потоци в Ноуд-РЕД. Най-външният цикъл обхожда езиците, за които се извличат ресурси. За всеки език се изпълнява заявка HTTP GET, която има следния вид:

- За езиковите модели: https://huggingface.co/models?`language={{{lang}}}`
- За наборите от данни: https://huggingface.co/datasets -json?language=language:{{{lang}}}

Параметърът `{{{lang}}}` и в двете заявки се замества с конкретния език от итерацията, като за стойност се използва кодът ISO 14[11] на езика – bg, en, de, и т. н.

Резултатът от всяка заявка съдържа страница с езикови модели или набори от данни, респективно. Данните от страницата се извличат на база на описания по-горе анализ, за да се определи броят страници, които трябва да бъдат обходени – това е вложеният цикъл от първо ниво. При всяка негова итерация се обхожда една страница, която съдържа максимум 30 ресурса, отново с HTTP GET заявка. Резултатът от тази заявка съдържа масив с ресурсите от съответната страница, обхождането на който формира вложения цикъл от второ ниво. В него се обхожда всеки ресурс – езиков модел или набор от данни, за да се съберат метаданните, които представляват интерес, и да се съхранят в базата от данни.

Следващото предизвикателство е, че при извличане на страниците с набори от данни (datasets), ако сортировката е по подразбиране, т. е. адресът URL е формиран във вида https://huggingface.co/datasets?language=language:bg&sort=trending, например за български език, в ресурсите на страниците се срещат повторения. Такива бяха установени за следните ресурси на български: facebook/flores (2 пъти), multi_eurlex, setimes – 3 пъти. Такива повторения бяха установени и за немски език, вероятно е имало и за други езици, но това бе достатъчно, за да се потвърди наличието на програмен проблем. Тези повторения не влияеха на общия брой, но тъй като при съхранение в базата от данни повторенията се игнорират, то не всички ресурси биваха извличани, което доведе до изкривяване на резултатите в база от данни. Отново експериментално беше установено, че ако сортирането е по брой изтегляния на ресурсите, т. е. адресът URL е формиран като https://huggingface.co/datasets?language=language:bg&sort=downloads, дубликати не се срещат. Това поведение беше установено за периода юни – август, 2023 г. Към момента този проблем в Hugging Face вече изглежда отстранен, но за всеки случай се наложи да бъде приложена метрика върху резултатите при всяко ново извличане, на базата на която да бъде генерирано предупреждение до администраторите, ако условията на метриката бъдат изпълнени. Метриката е съвсем проста – ако новоизвлеченият брой на ресурсите за даден език е по-малък или равен на броя от предходното извличане, се генерира предупреждение – в момента то се изразява в просто изпращане на писмо, но може да бъде реализирано и като REST заявка до външна система с цел автоматизирана обработка на получаваните предупреждения.

4. Заключение

Реализацията на представената система демонстрира един от възможните подходи при имплементиране на решение за извличане на дан-

ни от източници в интернет в стил уебобхождане (web crawling). Обхождането по този начин, за съжаление, не е алгоритъм, който, реализиран веднъж, би могъл да бъде използван сравнително дълго, защото този подход е силно зависим от очаквания формат на данните за обхождане. Направената реализация също показва и част от предизвикателствата при подобно обхождане, на които алгоритъмът може да се натъкне – някои от тях практически невъзможни за решаване без промяна на програмния код. Изискванията на робота, индексиращ уебстраници за компанията Google, също показват, че практически универсално решение вероятно не съществува.

Предимство на представената реализация е фактът, че в основата си това решение може да се използва за обхождане на различни системи с различни цели и формати на извличаните данни, за което обаче ще е нужна адаптация на програмния код, извършващ обработка на данните.

БЕЛЕЖКИ

1. The AI community building the future. https://huggingface.co
2. GitHUB. https://github.com
3. Node-RED. Low-code programming for event-driven applications. https://nodered.org
4. Node.js. An open-source, cross-platform JavaScript runtime environment. https://nodejs.org
5. MariaDB Server: The open source relational database. https://mariadb.org
6. Canonical Ubuntu. https://ubuntu.com
7. MySQL. https://mysql.com
8. GrafanaLabs. https://grafana.com
9. Denial-of-service attack. https://en.wikipedia.org/wiki/Denial-of-service_attack
10. Postman. An API platform for building and using APIs. https://www.postman.com
11. ISO 639. Language Code. https://www.iso.org/iso-639-language-codes.html

REFERENCES

ABDURAKHMONOVA, N, ISMAILOV, A.S., 2022. Web Crawler Technologies as a Tool for Compiling Parallel Corpora. *XXV Akhanov readings International Scientific and Methodological Conference Sustainable Development Language. Intercultural Communication and Digital Technologies*. Almati, Kazakhstan.

BERGMAN, J, POPOV, O., 2023. Exploring Dark Web Crawlers: A Systematic Literature Review of Dark Web Crawlers and Their Implementation. https://ieeexplore.ieee.org/document/10064292.

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.

CHOLAKOV, G., DOYCHEV, E., KOEVA, S., 2023. System for retrieval and visualization of data from Internet. *The Mathematics and Informatics journal*, vol. 66, no. 5. [In Bulgarian]. https://azbuki.bg/uncategorized/sistema-za-izvlichane-i-vizualizacziya-na-danni-ot-internet/.

GASPARI, F., WAY, A., DUNNE, J., REHM, G., PIPERIDIS, S., GIAGKOU, M., 2021. *Deliverable D1.1 Digital Language Equality* (preliminary definition). Project deliverable; EU project European Language Equality (ELE); Grant Agreement no. LC-01641480 – 101018166 ELE. https://european-language-equality.eu/wp-content/uploads/2021/05/ELE_Deliverable_D1_1.pdf.

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.

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., 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.

NAJORK, M., 2017. Web Crawler Architecture. In: Liu, L., Özsu, M. (eds) *Encyclopedia of Database Systems. Springer, New York, NY*. https://doi.org/10.1007/978-1-4899-7993-3_457-3.

SUN, Y., COUNCILL, I., GILES, C., 2010. The Ethicality of Web Crawlers. *Proc. 2010 IEEE/WIC/ACM Int. Conf. on Web Intelligence, WI 2010*, pp. 668 – 675. https://doi.org/10.1109/WI-IAT.2010.316

CHALLENGES IN WEB CRAWLING FOR DATA COLLECTION

Abstract. The article presents the challenges of implementing a System for data retrieval and visualisation from the Internet by crawling language resources from the Hugging Face repository and extracting the associated data. The data in the system is updated at regular intervals to track the dynamics of language resource creation for different time periods. The article presents: a) the analysis of the available data and its structure; b) the chosen method for crawling the pages and extracting the data. The shared experience of overcoming the specific challenges can serve to solve similar problems related to the extraction of data from the Internet, a task that often has to be solved in various projects (including school projects).

Keywords: web crawling; automatic data extraction; linguistic datasets

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

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

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