

Umjetna inteligencija i *vibe coding* u službi izrade daktilografskih vježbi

Mario Marković, mag. educ. inf. et. mag. inf.

Srednja strukovna škola Samobor



Stručni rad

Sažetak

Rad opisuje kako se alati generativne umjetne inteligencije i pristup poznat kao *vibe coding* mogu koristiti za izradu prilagođenih daktilografskih vježbi bez naprednog programerskog znanja. U radu su uspoređeni odgovori alata ChatGPT, Gemini i Perplexity AI.

Ključne riječi: umjetna inteligencija, daktilografske vježbe, alat

Uvod

Razvoj alata generativne umjetne inteligencije omogućio je širem krugu korisnika računala brže pronalaženje informacija, izradu sadržaja te jednostavniju obradu i analizu podataka. Posebno se ističe mogućnost izrade programskih rješenja i bez naprednog programerskog predznanja. Pojam koji se u novije vrijeme često spominje jest *vibe coding*. Riječ je o pristupu programiranju u kojem korisnik alatu umjetne inteligencije opisuje željenu funkcionalnost na prirodnom jeziku, a alat na temelju upita generira programski kod. U ovom radu opisano je kako se alati umjetne inteligencije i pristup *vibe codinga* mogu biti korisni za izradu prilagođenih tekstova za učenje daktilografije, čak i onim korisnicima koji nemaju prethodno znanje o programiranju.

Problem pronalaženja relevantnih riječi za izradu daktilografskih vježbi

Kod izrade daktilografskih vježbi nastavnici se često susreću s problemom odabira odgovarajućih riječi i tekstova. Pritom je važan raspon riječi i znakova koje tekst sadrži, odnosno važno je da vježba obuhvaća velik broj različitih slova i znakova, a po mogućnosti i češću pojavnost onih znakova koji učenicima u procesu učenja predstavljaju poteškoće. Prema iskustvu autora, takvi su znakovi najčešće ć, ž, đ i š, no poteškoće se javljaju i pri pisanju brojeva te određenih interpunkcijskih znakova. Osim broja različitih i relevantnih znakova, važno je koristiti i semantički ispravan tekst koji je usklađen sa strukovnim kurikulumima škola. U kurikulumima objavljenima u okviru reforme obrazovanja iz 2025. godine ističe se važnost povezanosti nastavnih modula sa strukom i međupredmetnim temama. U tom se smislu umjetna inteligencija može koristiti za izradu tekstova prilagođenih učenicima na obje navedene razine. U ovom radu opisano je kako se umjetna inteligencija može koristiti pri generiranju specifičnih riječi na hrvatskom jeziku za izradu vježbi namijenjenih učenicima prve godine učenja daktilografije koji još uvijek nisu naučili pravilno tipkati sva slova.

Ograničenja generativne umjetne inteligencije u pronalaženju prikladnih riječi

Veliki jezični modeli dostupni u besplatnim verzijama imaju ograničenja u broju upita pa prosječan korisnik nailazi na svojevrsan zid prilikom rada s besplatnim alatima umjetne inteligencije. Stoga mnogi autori napominju važnost izrade dobro strukturiranih upita kako bi izbjegli halucinacije umjetne inteligencije i nepotpune odgovore. Ipak, prilikom izrade daktilografskih vježbi utemeljenih na specifičnim znakovima, ovi su se alati pokazali nedovoljno dobrima te pokazuju česte pogreške u odgovorima. U nastavku se nalazi primjer upita i odgovora komercijalne inačice ChatGPT modela, Gemini modela te Perplexity AI-a u besplatnoj verziji:

Upit:

Pronađi barem 10 riječi na hrvatskom jeziku po sljedećim uvjetima: riječi smiju sadržavati isključivo ova slova i niti jedno drugo slovo izvan ovog raspona: ASDFKLČ. Bilo

koja druga slova i interpunkcijski znakovi nisu dozvoljeni. Jezik mora biti isključivo hrvatski i ne smiju se koristiti kratice. Neka riječi budu duže od 5 znakova.

Odgovor ChatGPT:

sadaka, dasaka, sadaka, kasalj, čkalja, jasala, salaka, sklada, klada, sladak, sladka, kalasa, ...

Odgovor Gemini:

jadala, kasala, lakala, sadala, skakala, sladak, sakala, sadaka, kasada, falada, kasaka, jakada, ...

Odgovor Perplexity AI:

klad, klasa, klasak, flaskas, flaska, laškav, laskati, kladilac, kladiti, skladištar, skladištiti, ...

Kao što je vidljivo iz priložene tablice, daleko najbolje rezultate dao je ChatGPT model, iako riječi u odgovoru nisu često korištene i uobičajene hrvatske riječi. Rezultati Gemini alata još su upitniji – neke riječi iz odgovora ne postoje u hrvatskom jeziku. Kod alata Perplexity dobili smo velik broj riječi koje sadržavaju i druga slova, ali su, osim toga, i gramatički neispravne (npr. laškav). Pretpostavka jest da se s više različitih upita ipak može doći do traženih riječi. Međutim, radi veće preciznosti i šireg opsega dobivenih riječi te radi jednostavnosti upita i ograničenosti besplatnih modela ipak je preporuka izraditi jednostavno programsko rješenje koje će iz postojeće baze riječi pronaći samo one relevantne i to s neograničenim brojem upita.

Izrada skripte i priprema korpusa hrvatskih riječi

Korpus hrvatskih riječi preuzet je s repozitorija na GitHubu. Autor tog repozitorija je dr. Goran Igaly, viši predavač na Prirodoslovno matematičkom fakultetu - Matematičkom odsjeku Sveučilišta u Zagrebu. Rječnik hrvatskoga jezika dio je sveobuhvatnog projekta čiji je cilj izraditi i učiniti slobodno dostupnima resurse potrebne za promicanje hrvatskoga jezika. Ovaj rječnik sadrži više od 602.185 riječi i pojava koje se pojavljuju u različitim tekstovima, od knjiga i novinskih članaka pa do izvora na internetu. Za potrebe ovog zadatka preuzeta je datoteka pod nazivom HR_Txt-624.txt koja sadrži kompletnu bazu riječi te je premještena u mapu u koju će se kasnije

postaviti i Python skripta (program) za pronalazak riječi. Potom je tekstualna datoteka preimenovana u baza_rijeci.txt radi jednostavnijeg korištenja.

Za izradu programske skripte potrebno je instalirati programski jezik Python na računalo. Postupak je vrlo jednostavan: potrebno je preuzeti i instalirati program sa službene stranice python.org. Zatim je potrebno u istoj mapi, gdje se nalazi baza, izraditi datoteku program.py, koju potom otvaramo desnim klikom, a zatim opcijom Edit with IDLE, ili otvoriti IDLE na računalu pa zatim otvoriti ovu praznu datoteku. Ovdje ćemo kasnije zalijepiti programski kod generiran od strane Chat GPT modela.

Vibe coding i izrada skripte

Sa svim materijalima spremnima postavljamo upit alatu umjetne inteligencije kako bismo izradili program koji će pronaći relevantne riječi prema našem upitu. Upit je potrebno konkretno i precizno strukturirati kako bismo izbjegli moguće halucinacije. U upitu je potrebno modelu napomenuti da se riječi pretražuju u navedenoj datoteci tog naziva.

Uvjeti koje bi program trebao sadržavati su sljedeći:

- Program iz baze treba pronaći samo riječi koje sadržavaju određena slova (npr. *a, s, d, f, j, k, l, č*) – sve riječi koje sadržavaju druga slova program treba ignorirati.
- Program treba ponuditi opciju da pronađene riječi sadrže samo određena slova (npr. *a, s, d, f, j, k, l, č* te, uz taj niz, sve riječi trebaju obavezno sadržavati slova *a, č*, a ostale riječi tog niza koje ne sadrže *a, č* se ignoriraju).
- Program treba imati opciju da se unese i traženi niz ili slog koji sadrži određena riječ (npr. sve riječi koje sadrže slog *ije*).
- Program treba imati opciju da se odredi duljina riječi.
- Program treba ispisati sve rezultate i broj riječi.

Odgovor modela ChatGPT:

```
import re
```

```
allowed_letters = input(
```

```

    "Unesite dozvoljena slova (npr.
    ASDFJKLČERUICVNM) ili upišite SVE za
    sva slova: "
).upper()

min_length = int(input("Unesite
    minimalnu duljinu riječi: "))

required_letters = input(
    "Unesite obavezna slova koja svaka
    riječ mora sadržavati (npr. ČAFJ).
    Ako nema, samo pritisnite Enter: "
).upper()

required_sequence = input(
    "Unesite obavezan niz slova koji
    riječ mora sadržavati (npr. IJE). Ako
    nema, samo pritisnite Enter: "
).upper()

required_set = set(required_letters)

allow_all_letters = allowed_letters ==
    "SVE"

if not allow_all_letters:
    allowed_set = set(allowed_letters)

    if not required_set <= allowed_set:
        print("Greška: obavezna slova
        moraju biti unutar skupa dozvoljenih
        slova.")
        exit()

    if not set(required_sequence) <=
        allowed_set:
        print("Greška: sva slova iz
        obaveznog niza moraju biti unutar
        skupa dozvoljenih slova.")
        exit()

file_name = "baza_rijeci.txt"

```

```

try:
    with open(file_name, "r",
        encoding="utf-8") as file:
        content = file.read()
except FileNotFoundError:
    print(f"Datoteka '{file_name}'
    nije pronađena!")
    exit()

words = re.findall(r'\b\w+\b',
    content)

if allow_all_letters:
    filtered_words = {
        word for word in words
        if len(word) >= min_length
        and required_set <=
        set(word.upper())
        and required_sequence in
        word.upper()
    }
else:
    filtered_words = {
        word for word in words
        if set(word.upper()) <=
        allowed_set
        and len(word) >= min_length
        and required_set <=
        set(word.upper())
        and required_sequence in
        word.upper()
    }

# Ispis rezultata
print("Riječi koje zadovoljavaju
    uvjet:")
print(",
    ".join(sorted(filtered_words)))
print("Broj pronađenih riječi:",
    len(filtered_words))

```

Potom je provjerena ispravnost koda i testirana na različitim primjerima. Uz nekoliko nadopuna dobiven je gotov program koji pronalazi riječi s točno postavljenim uvjetima. U nastavku je prikazan primjer u kojem su tražene riječi isključivo u rasponu slova osnovnog položaja te zahvata na slova *e, i, r, u*, uvjet je da riječ bude dulja od 6 znakova, da obavezno sadrži slova *e, r, u, i* te da obavezno sadrži niz *ije*:

Prikaz u programskom sučelju:

```
Unesite dozvoljena slova (npr.
ASDFJKLČERUICVNM) ili upišite SVE za
sva slova: ASDFJKLČERUI
Unesite minimalnu duljinu riječi: 6
Unesite obavezna slova koja svaka riječ
mora sadržavati (npr. ČAFJ). Ako nema,
samo pritisnite Enter: ERUI
Unesite obavezan niz slova koji riječ
mora sadržavati (npr. IJE). Ako nema,
samo pritisnite Enter: IJE
```

Riječi koje zadovoljavaju uvjet:
 Rusije, dječurlije, drukčije, furije,
 karijeru, karijesu, krijesu,
 kukurijek, kukurijeka, kukurijekali,
 kukurijeke, kukurijeku, kukuriječe,
 kukuriječu, kurije, lijeru, rasijeku,
 rijeku, riječju, riječku, srijedu,

Broj pronađenih riječi: 25

Iz dobivenog odgovora možemo zaključiti da su sve riječi isključivo hrvatskog jezika, gramatički ispravne te iskoristive za konkretne vježbe zahvata na slova *E, I i R, U*. Odgovor je bolji i precizniji je je program otvoren za usavršavanje prema potrebama korisnika. Također, broj upita uz ovakvo jednostavno programsko rješenje je neograničen.

Zaključak

Razvoj alata umjetne inteligencije omogućio je i prosječnim korisnicima računala izradu jednostavnih programskih rješenja bez naprednog programerskog predznanja. Na primjeru prikazanom u ovom radu vidi se kako se pristup tzv. *vibe coding*-a može praktično

primijeniti u obrazovanju, i to za rješavanje konkretnog problema iz nastavne prakse. Umjesto ručnog pretraživanja i osmišljavanja riječi, nastavnik uz pomoć jednostavne skripte može brzo izdvojiti one riječi koje u potpunosti odgovaraju zadanim slovnim i jezičnim uvjetima.

Takav pristup posebno je koristan u nastavi daktilografije jer omogućuje bolju prilagodbu vježbi u skladu s individualnim potrebama učenika. Učenici tako mogu češće vježbati upravo ona slova, skupove slova ili slogove koji im stvaraju najviše poteškoća, dok nastavnik dobiva alat kojim može preciznije i učinkovitije pripremati nastavne materijale. Time umjetna inteligencija ne preuzima ulogu nastavnika, nego postaje sredstvo koje mu olakšava rad i proširuje mogućnosti prilagodbe nastave.

Literatura

Google (b.d.) Gemini.

<https://gemini.google.com/> (17. 4. 2026.).

Igaly, G. (b.d.) Rječnik hrvatskih jezika. GitHub repozitorij. <https://github.com/gigaly/rjecnik-hrvatskih-jezika/> (17. 4. 2026.).

OpenAI (b.d.) ChatGPT.

<https://chatgpt.com/> (Pristupljeno: 17. 4. 2026.).

Perplexity (b.d.) Perplexity.

<https://www.perplexity.ai/> (17. 4. 2026.).

Plavljanić, B. (2026) 'Vibe coding: Revolucija u razvoju softvera ili prolazni trend?', PC CHIP

<https://pcchip.hr/softver/korisni/vibe-coding-revolucija-u-razvoju-softvera-ili-prolazni-trend/> (17. 4. 2026.).

Wawrzynek, A. (b.d.) Najčešće greške u daktilografiji i kako ih izbjeći. PDF. Preuzeto s: CARNET.hr