

D6.4 - VeMo-IaC

Druga različica programske opreme

31.8.2025

Podatki o dokumentu	
Različica	1.0
Pričakovan datum oddaje	31.8.2025
Datum oddaje	31.8.2025

RRP	6	Oznaka izročka	D6.4
Povezani izročki	D6.3	Dostop	Javno
Glavni urednik	Daniel Vladušič	Glavni avtor	Nikola Sekulovski
Ostali	Nejc Bat, Jure Medvešek		

Ključne besede
Programska oprema, Veliki jezikovni modeli (LLM), IaC

Kazalo

1	Uvod	6
1.1	Namen dokumenta in povezava z drugimi deli projekta	6
1.2	Struktura dokumenta.....	6
2	Izbrani pristopi in eksperimenti.....	7
2.1	Opis izbranih pristopov	7
2.2	Eksperimenti opravljeni z izbranimi pristopi	7
2.3	Izvedbe učenja/eksperimenti	8
2.3.1	Model + Steampunk	8
2.3.2	Model zgrajen iz Steampunk množice z nadvzorčenjem (oversampling)	9
2.3.3	Model + Steampunk + sintetični podatki + podatki enotnih testov.....	9
2.3.4	Model + Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje.....	10
2.3.5	Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje + izpopolnjeni GitHub podatki	10
2.4	Evaluacijski postopek.....	11
3	Rezultati.....	12
4	Vrednotenje rezultatov eksperimentov.....	13
4.1	Steampunk	13
4.2	Model + Steampunk + nadvzorčenje (oversampling)	15
4.3	Model + Steampunk + sintetični podatki + podatki enotnih testov	16
4.4	Model + Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje	18
4.4.1	Začetni poskusi bogatenja opisov in ponovnega maskiranja vrednosti parametrov.....	18
4.4.2	Manjši in bolj osredotočen podatkovni nabor z avtomatskim in ročnim bogatenjem Ansible nalog ter opisov/pozivov	20
4.5	Model + Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje + izpopolnjeni GitHub podatki.....	22
5	Zaključek.....	25

6	Literatura	26
---	------------------	----

Kazalo Tabel

Tabela 1: Rezultati vsake inačice modela	12
--	----

Kazalo Slik

Slika 1: Izguba pri učenju glede na korake med doučevanjem na podatkovnem naboru Steampunk	8
Slika 2: Validacijska izguba glede na korake med doučevanjem na podatkovnem naboru Steampunk	8
Slika 3: Izguba pri učenju glede na korake med doučevanjem na podatkovnem naboru Steampunk s prekomernim vzorčenjem.	9
Slika 4: Validacijska izguba glede na korake med doučevanjem na podatkovnem naboru Steampunk s prekomernim vzorčenjem.	9
Slika 5: Izguba pri učenju glede na korake med doučevanjem na podatkovnem naboru Steampunk + enotni testi + sintetični podatki.	9
Slika 6: Validacijska izguba glede na korake med doučevanjem na podatkovnem naboru Steampunk + enotni testi + sintetični podatki.	9
Slika 7: Izguba pri učenju glede na korake med doučevanjem na izbranem podnaboru podatkov iz nabora Steampunk z začetnim ponovnim maskiranjem in bogatenjem.	10
Slika 8: Validacijska izguba glede na korake med doučevanjem na izbranem podnaboru podatkov iz nabora Steampunk z začetnim ponovnim maskiranjem in bogatenjem.	10
Slika 9: Izguba pri učenju glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov s ponovnim maskiranjem in bogatenjem.	10
Slika 10: Validacijska izguba glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov s ponovnim maskiranjem in bogatenjem.	10
Slika 11: Izguba pri učenju glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov z ponovnim maskiranjem in bogatenjem + izpopolnjene GitHub podatke.	11
Slika 12: Validacijska izguba glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov z ponovnim maskiranjem in bogatenjem + izpopolnjene GitHub podatke.	11

Povzetek

Izroček D6.4 predstavlja nadaljevanje dela, opisanega v D6.3 in D6.2, s poudarkom na učenju in evaluaciji modela z različnimi iteracijami podatkovnih množic. V dokumentu so predstavljeni rezultati, ki smo jih pridobili z rabo podatkovnih množic, zgrajenih v D6.3. Vsaka iteracija modela je bila ovrednotena s pomočjo orodja Spotter, dodatne ocene pa je podal tudi GPT-4o-mini in Ansible strokovnjaki.

Rezultati kažejo postopno izboljševanje kakovosti modela, pri čemer se indeks resnosti večinoma znižuje, semantična kakovost pa se povečuje. Najboljši rezultat je bil dosežen z uporabo kombinacije Steampunk množice, obogatenih opisov, sintetičnih in enotskih podatkov ter izpopolnjenih GitHub podatkov. Ta različica modela je dosegla skoraj brezhibne rezultate na evaluacijskem naboru ter izkazala zanesljivo sledenje zahtevnejšim navodilom.

Kot zelo pomembno izpostavljamo dejstvo, da je model ocenjeval tudi Ansible strokovnjak in da je trenutno v interni uporabi.

1 Uvod

1.1 Namen dokumenta in povezava z drugimi deli projekta

Ta dokument dopolnjuje delo, opravljeno v poročilu **D6.3**, saj podrobneje razlaga razvoj modela in njegovo učinkovitost z uporabo novejših podatkov, oz. podatkovnih množic, ki so bile zgrajene od M12 do M24 in so v njem opisane.

1.2 Struktura dokumenta

Dokument je sestavljen iz naslednjih poglavij:

- Izbrani pristopi in eksperimenti – poglavje 1 – razlaga za izbiro našega modela.
- Rezultati – poglavje 2 – rezultati različnih iteracij modela na evaluacijskem podatkovnem naboru.
- Vrednotenje rezultatov – poglavje 3 – podrobnejša razlaga zmogljivosti posameznih iteracij modela.

2 Izbrani pristopi in eksperimenti

2.1 Opis izbranih pristopov

Kot smo opisali v D6.2, je CodeLlama-7B-Instruct-hf dosegel najbolj obetavne rezultate glede natančnosti in je bil zato izbran kot osnova za nadaljnji razvoj. V tem obdobju smo se osredotočili predvsem na gradnjo in iterativno izboljševanje podatkovnih množic ter na prilagajanje modela z učenjem po navodilih. Vsaka različica je bila avtomatsko evaluirana, vendar smo uvedli tudi testiranje z eksperti. Strokovnjaki za Ansible so testirali zgrajene modele in podali oceno.

2.2 Eksperimenti opravljeni z izbranimi pristopi

V tej sekciji podajamo podroben pregled učnih parametrov, obdelave podatkovnih naborov strategije učenja, ki smo jih uporabili pri šestih eksperimentalnih izvedbah, opisanih spodaj.

Učni parametri so bili pri vseh izvedbah enaki in so povzeti tako:

- Število epoh: 2
- Velikost paketa za učenje in validacijo: 2–5
- Začetna stopnja učenja: 2e-5
- Tip razporejevalnika (scheduler): kosinus

Število epoh je bilo nastavljeno na dve, da se je vsak učni primer obdelal več kot enkrat. Izognili smo se večjemu številu epoh pri učenju, saj zasledujemo cilj, da je takšno učenje pri stranki relativno hitro in poceni, prav tako pa se s tem izognemo pretiranemu prilagajanju (overfitting). Velikosti paketov so se razlikovale glede na uporabljeno različico podatkovnega nabora. Izbira razporejevalnika v kombinaciji z stopnjo učenja 2e-5 je omogočala stabilnost in gladko konvergenco skozi učenje.

Razdelitev podatkov je bila izvedena dinamično – 95 % podatkov smo namenili učenju in 5 % validaciji v vsaki iteraciji. Ker so bile te razdelitve ustvarjene sproti, se velikosti učnih in validacijskih naborov med iteracijami razlikujejo. Posledično validacijske izgube med različnimi različicami podatkovnih naborov niso neposredno primerljive.

V primerjavi z začetnim pristopom doučevanja, opisanim v D6.2, obstaja nekaj ključnih razlik. Čeprav je naš cilj ostal enak: čim bolj povečati natančnost modela in njegovo sposobnost sledenja navodilom, so nas omejitve (čas, računska moč) prisilile v določene kompromise. Tako kot v D6.2 smo izkoristili prednosti pristopa LoRA (Low-Rank Adaptation), ki omogoča velike prihranke pri računski porabi in hkrati ohranja močno prilagodljivost domeni. Vendar pa smo se v tej fazi odločili odstraniti 4-bitno kvantizacijo (Li et al., 2025). Čeprav

kvantizacija zmanjša pomnilniške zahteve, lahko negativno vpliva na natančnost. Glede na razpoložljive vire je bil sam LoRA zadosten kompromis med učinkovitostjo in zmogljivostjo. Ta odločitev tako predstavlja zavesten premik v prid večji natančnosti na račun večje porabe virov.

Za nadaljnje izboljšanje prilagoditve domeni in splošne zmogljivosti smo izpopolnili konfiguracijo LoRA (Hu et al., 2021). Konkretno smo povečali dimenzijo rank, faktor skaliranja (alpha), uvedli izgubo za zmanjšanje pretiranega prilagajanja in razširili LoRA tako, da je zajemal tudi večplastno nevronske omrežje (MLP) poleg pozornostih (attention) plasti.

Končna konfiguracija LoRA je bila naslednja:

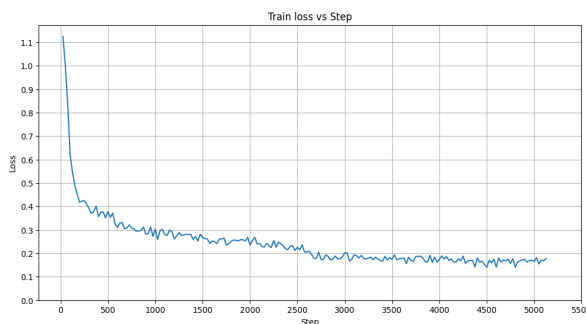
- Lora rank: 64
- Lora alpha: 128
- Lora izguba: 0.1
- Ciljni moduli: pozornostni (attention) in MLP plasti

2.3 Izvedbe učenja/eksperimenti

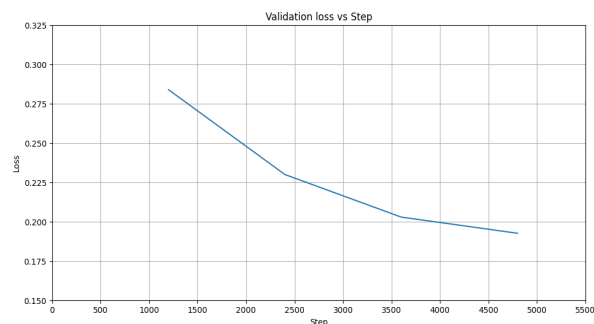
V tem poglavju predstavljamo učne in validacijske krivulje modela na vsaki različici podatkovnega nabora. Kot lahko vidimo, se učna in validacijska izguba lepo zmanjšujeta z vsakim korakom pri obeh epohah doučevanja.

Ker pa podatkovni nabori med sabo niso enaki, teh rezultatov ni smiselno združevati in neposredno primerjati med različnimi izvedbami – primerjava izgub različnih podatkovnih naborov namreč ne bi bila upravičena.

2.3.1 Model + Steampunk

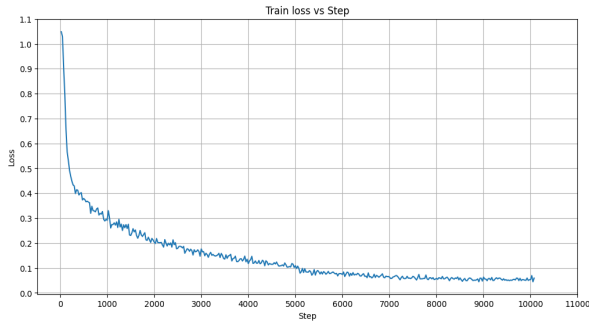


Slika 1: Izguba pri učenju glede na korake med doučevanjem na podatkovnem naboru Steampunk

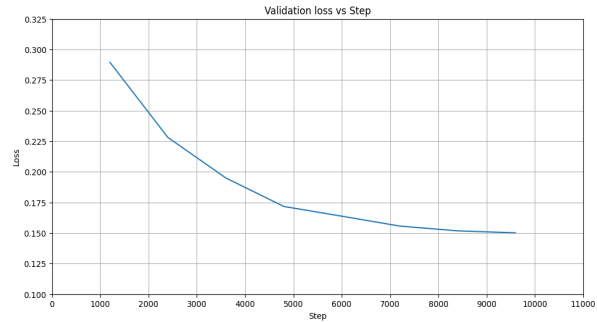


Slika 2: Validacijska izguba glede na korake med doučevanjem na podatkovnem naboru Steampunk

2.3.2 Model zgrajen iz Steampunk množice z nadzorčenjem (oversampling)

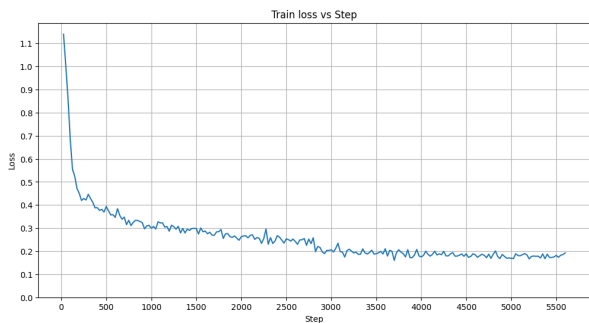


Slika 3: Izguba pri učenju glede na korake med doučevanjem na podatkovnem naboru Steampunk s prekomernim vzorčenjem.

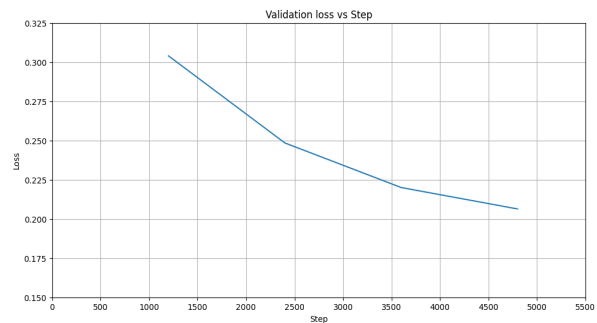


Slika 4: Validacijska izguba glede na korake med doučevanjem na podatkovnem naboru Steampunk s prekomernim vzorčenjem.

2.3.3 Model + Steampunk + sintetični podatki + podatki enotnih testov



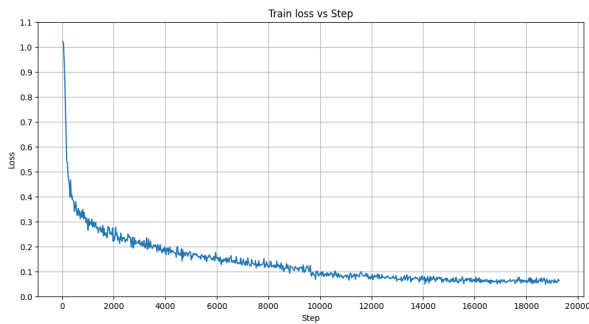
Slika 5: Izguba pri učenju glede na korake med doučevanjem na podatkovnem naboru Steampunk + enotni testi + sintetični podatki.



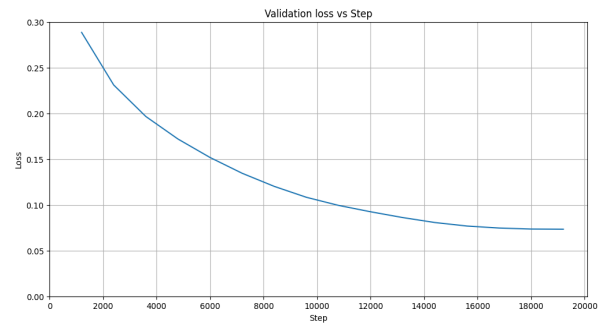
Slika 6: Validacijska izguba glede na korake med doučevanjem na podatkovnem naboru Steampunk + enotni testi + sintetični podatki.

2.3.4 Model + Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje

2.3.4.1 Začetni poskusi bogatenja opisov in ponovnega maskiranja vrednosti parametrov



Slika 7: Izguba pri učenju glede na korake med doučevanjem na izbranem podnaboru podatkov iz nabora Steampunk z začetnim ponovnim maskiranjem in bogatenjem.

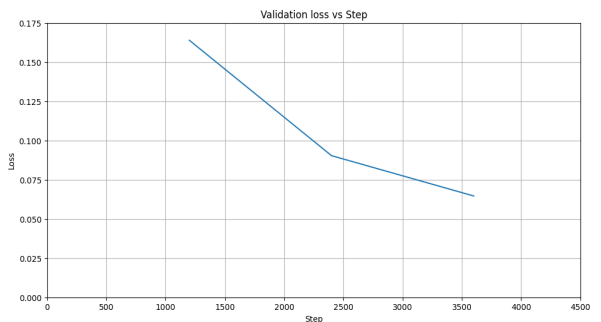


Slika 8: Validacijska izguba glede na korake med doučevanjem na izbranem podnaboru podatkov iz nabora Steampunk z začetnim ponovnim maskiranjem in bogatenjem.

2.3.4.2 Manjši in bolj osredotočen podatkovni nabor z avtomatskim in ročnim bogatenjem Ansible nalog ter opisov/pozivov

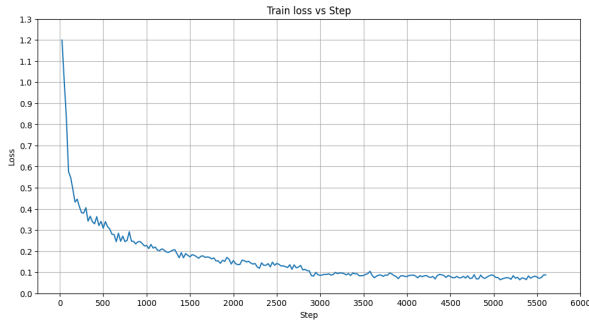


Slika 9: Izguba pri učenju glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov s ponovnim maskiranjem in bogatenjem.

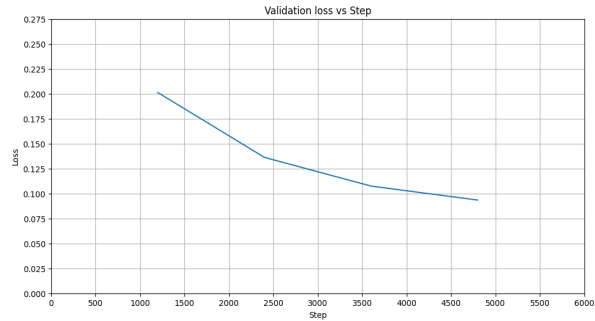


Slika 10: Validacijska izguba glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov s ponovnim maskiranjem in bogatenjem.

2.3.5 Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje + izpopolnjeni GitHub podatki



Slika 11: Izguba pri učenju glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov z ponovnim maskiranjem in bogatenjem + izpopolnjene GitHub podatke.



Slika 12: Validacijska izguba glede na korake med doučevanjem na izbranem podnaboru iz osredotočenega podatkovnega nabora, ki je vključeval Steampunk + sintetične podatke + podatke enotnih testov z ponovnim maskiranjem in bogatenjem + izpopolnjene GitHub podatke.

2.4 Evaluacijski postopek

Evaluacija vsake različice modela je bila izvedena na sintetično ustvarjenem podatkovnem naboru, ki je vseboval 32 Ansible nalog in ustrezne pozive (prompte), izbranih iz 22 različnih modulov. Za vsak od teh 32 pozivov je moral doučen model generirati Ansible nalogo, ki je bila nato ocenjena z dvema dopolnjujočima se pristopoma:

1. Spotter – ta orodje oceni vsako nalogo posebej in poda napake, opozorila, namige ali brez težav.
2. GPT-4o-mini – dobil je vnos z generirano nalogo in Spotterjevim izhodom ter moral dodeliti numerično oceno v razponu od 0 do 10.

Za kvantificiranje zmogljivosti smo upoštevali skupno število opozoril in napak, ki jih je vrnil Spotter. Uvedli smo indeks resnosti: uteženo vsoto vseh opozoril in napak, kjer so imele napake utež 10, opozorila pa utež 1 – saj so napake bistveno bolj kritične. Ta metrika je omogočila podrobnejšo oceno učinkovitosti modela čez celoten evaluacijski podatkovni niz.

Pomembno je omeniti, da so vsi pozivi, podani modelu, izrecno vključevali FQCN (fully qualified collection name) modula, ki ga je bilo treba uporabiti. V prihodnjih iteracijah želimo to omejitev odpraviti, s čimer bi izboljšali uporabnost in evaluacijski postopek še bolj približali resničnim scenarijem uporabe.

3 Rezultati

Tabela 1: Rezultati vsake inačice modela

	Napake	Opozorila	Indeks resnosti	GPT ocena
Steampunk	2	2	22	6.75
Steampunk + nadzorčenje	2	3	23	6.84
Steampunk + sintetični podatki + podatki enotnih testov	1	5	15	6.84
Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje	3	3	33	6.98
Manjši in bolj osredotočen podatkovni nabor z avtomatskim in ročnim bogatenjem Ansible nalog ter opisov/pozivov	1	2	12	7.67
Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje + izpopolnjeni GitHub podatki	0	1	1	8.09

V Tabela 1 so prikazane ocene posamezne različice modela na našem evaluacijskem podatkovnem naboru. Vidimo lahko, da je – razen pri poskusu s prekomernim vzorčenjem in pri prvem bogatenju opisov (description augmentation) – vsaka nova različica podatkovnega nabora (in s tem modela) dosegla boljše rezultate kot prejšnja. To je jasno razvidno iz indeksa resnosti, ki se večinoma zmanjšuje skozi različne izvedbe (izjema sta nadzorčenje in začetni poskus z bogatenjem opisov).

Rezultati GPT-4o-mini pa nekoliko odstopajo – ocene se ne izboljšujejo vedno v skladu z nižjim indeksom resnosti. Na primer, GPT je pri začetnem poskusu bogatenja opisov dodelil višjo povprečno oceno kot pri vseh prejšnjih izvedbah do tiste točke. To lahko pomeni bodisi, da ta metrika ni povsem zanesljiva, bodisi da GPT pri ocenjevanju ni upošteval le Spotterjevega rezultata (tj. skladnosti s sintakso), temveč tudi semantično kakovost nalog.

Najboljši rezultat – tako glede indeksa resnosti kot glede GPT ocene – je bil dosežen z zadnjo različico modela, ki je bila doučena na podatkovnem nizu, dopolnjenem z izpopolnjenimi GitHub podatki.

4 Vrednotenje rezultatov eksperimentov

4.1 Steampunk

V tej različici modela so bile napake na našem evaluacijskem podatkovnem nizu večinoma povezane z manjkajočimi obveznimi parametri ali z generiranjem medsebojno izključujočih se parametrov. Opozorila pa so se večinoma nanašala na nepravilne podatkovne tipe, dodeljene vrednostim parametrov.

Slab primer:

```
Input:
Install the nginx package and ensure it is present on systems belonging to the
Debian OS family.

-----
- Ideal output:
- name: Install a package based on OS family
  ansible.builtin apt:
    package: nginx
    state: present
    when: ansible_os_family == 'Debian'

-----
Generated Task:
- name: Install the nginx package and ensure it is present on systems belonging
to the Debian OS family.
  ansible.builtin apt:
    name: nginx
    state: present
    deb: '{{url}}'

-----
Spotter Evaluation:
Status:    error
Errors:    1
Warnings:  0

-----
ERROR: mutually-exclusive-arguments [E1200] (line: 2, column:3)
Arguments ['deb', 'name'] in module 'ansible.builtin apt' are mutually
exclusive.

-----
Evaluation Scores:
- GPT Judge Score: 4.0
```

Ta različica modela je sicer sposobna slediti enostavnejšim navodilom. Pravilno zna dodeliti vrednosti parametrov, omenjene v pozivu, k njihovim ustreznim imenom parametrov. Kljub temu pa občasno generira nekaj dodatnih parametrov, ki v pozivu niso bili izrecno navedeni, vendar naloga kljub temu ostane pravilna. Spodaj je prikazan primer.

Dober primer:

```
Input:
Copy the nginx configuration file from /localpath/nginx.conf to the remote
server at /etc/nginx/nginx.conf, ensuring the destination file has the correct
permissions and ownership for nginx to function properly.

-----
- Ideal output:
- name: Copy the nginx.conf file to the remote server
  ansible.builtin.copy:
    src: /localpath/nginx.conf
    dest: /etc/nginx/nginx.conf

-----
Generated Task:
- name: Copy the nginx configuration file from /localpath/nginx.conf to the
remote server at /etc/nginx/nginx.conf, ensuring the destination file has the
correct permissions and ownership for nginx to function properly.
  ansible.builtin.copy:
    src: /localpath/nginx.conf
    dest: /etc/nginx/nginx.conf
    owner: nginx
    group: nginx
    mode: '{{value}}'

-----
Spotter Evaluation:
Status:    warning
Errors:    0
Warnings:  0

-----
Evaluation Scores:
- GPT Judge Score: 7.0
```

Sposobnost modela, da sledi navodilom, se hitro poslabša, ko je soočen z bolj zapletenimi pozivi. V takih primerih skoraj povsem prezre navodilo in namesto tega generira ustrezne parametre iz modula z uporabo nadomestnih vrednosti. Čeprav je rezultat (Spotterjev izhod) sintaktično pravilen, je semantično napačen, saj ne ustreza zelenemu izidu. Spodaj je prikazana naloga v vprašanju.

Primer s slabim sledenjem navodilom:

```
Input:
Mount the device /dev/xvdb1 to the directory /data with the filesystem type
ext4, ensuring it is in a mounted state.

-----

- Ideal output:
- name: Mount /dev/xvdb1 to /data
  ansible.builtin.mount:
    path: /data
    src: /dev/xvdb1
    fstype: ext4
    state: mounted

-----

Generated Task:
- name: Mount the device /dev/xvdb1 to the directory /data with the filesystem
type ext4, ensuring it is in a mounted state.
  ansible.builtin.mount:
    path: '{{value}}'
    src: '{{value}}'
    fstype: '{{value}}'
    state: mounted

-----

Spotter Evaluation:
Status:    warning
Errors:    0
Warnings: 0

-----

Evaluation Scores:
- GPT Judge Score: 5.0
```

4.2 Model + Steampunk + nadzorčenje (oversampling)

Ker smo z nadzorčenjem povečali število učnih primerov z nalogami, ki so vsebovale več kot šest parametrov (glej D6.3), je bil model pogosteje izpostavljen manj zastopanim modulom in se je morda naučil določenih parametrov, s katerimi sicer ne bi bil seznanjen. Zato smo na našem evaluacijskem podatkovnem naboru zaznali dve napaki: ena je bila enaka napaki glede medsebojno izključujočih se argumentov kot pri različici modela iz poglavja 4.1, druga pa je bila povezana z manjkajočimi obveznimi parametri. Opozorila so se nanašala na nepravilne podatkovne tipe vrednosti parametrov. Čeprav se je skupna ocena izboljšala, je sposobnost modela za sledenje navodilom – oziroma pomanjkanje te

sposobnosti – ostala nespremenjena kot v poglavju 4.1, kar pomeni, da model ni sposoben slediti bolj zapletenim navodilom oziroma vsaj ne v celoti.

Primer s slabim sledenjem navodilom:

```
Input:
'Start an EC2 instance named instance in subnet subnet-5scalable with type
t3.small and image ami-123456, tagging it with Environment: Testing.'
```

4.3 Model + Steampunk + sintetični podatki + podatki enotnih testov

Ta različica je ustvarila le eno napako in pet opozoril - kar predstavlja precejšnje izboljšanje natančnosti na evalucijskem podatkovnem nizu. Napaka je bila povezana z generiranjem

medsebojno izključujočih se parametrov, opozorila pa so se nanašala na nepravilne podatkovne tipe pri določenih parametrih. Sposobnost sledenja navodilom pa je bila podobna kot pri različicah modela iz poglavij 4.1 in 4.2.

Slab primer:

Input:

Generate the nginx configuration file at /etc/nginx/nginx.conf using the template nginx.conf.j2 based on the host-specific variables.

- Ideal output:

- name: Template nginx config based on host variables

ansible.builtin.template:

src: nginx.conf.j2

dest: /etc/nginx/nginx.conf

Generated Task:

- name: Generate the nginx configuration file at /etc/nginx/nginx.conf using the template nginx.conf.j2 based on the host-specific variables.

ansible.builtin.template:

src: nginx.conf.j2

dest: /etc/nginx/nginx.conf

owner: root

group: root

mode: '{{value}}'

Spotter Evaluation:

Status: warning

Errors: 0

Warnings: 0

Evaluation Scores:

- GPT Judge Score: 7.0

4.4 Model + Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje

4.4.1 Začetni poskusi bogatenja opisov in ponovnega maskiranja vrednosti parametrov

Bogatenje pozivov/opisov (prompt/description augmentation) je bilo namenjeno izboljšanju sposobnosti sledenja navodilom, in sicer z doučevanjem modela na daljših in bolj zapletenih pozivih. Vendar pa so bili rezultati na evaluacijskem podatkovnem nizu razočarajoči, saj je ta različica dosegla najslabšo učinkovitost med vsemi modeli. Napake so bile večinoma povezane z manjkajočimi obveznimi parametri, napačnimi vrednostmi parametrov ali parametri, ki sploh niso pripadali ustreznemu modulu. Tako kot v prejšnjih iteracijah so bila opozorila večinoma povezana z nepravilnimi podatkovnimi tipi. Spodaj je podan ponazoritveni primer.

Slab primer:

```
Input:
Schedule a cron job to back up the nginx configuration every Sunday by creating
a tar.gz archive at /backup/nginx-config.tar.gz from /etc/nginx/

-----

- Ideal output:
- name: Add a cron job to backup nginx config every Sunday
  ansible.builtin.cron:
    name: 'Backup nginx config'
    weekday: '7'
    job: 'tar czf /backup/nginx-config.tar.gz /etc/nginx/'

-----

Generated Task:
- name: Schedule a cron job to back up the nginx configuration every Sunday by
creating a tar.gz archive at /backup/nginx-config.tar.gz from /etc/nginx/
  ansible.builtin.cron:
    name: backup nginx config
    special_time: Sunday
    job: tar -czf /backup/nginx-config.tar.gz /etc/nginx/

-----

Spotter Evaluation:
Status: error
Errors: 1
Warnings: 0

-----

ERROR: value-not-in-enum-choices [E601] (line: 2, column:3)
```

```
Value of 'special_time' in module 'ansible.builtin.cron' is set to 'Sunday'  
which does not match any of the possible choices listed in ['reboot', 'yearly',  
'annually', 'monthly', 'weekly', 'daily', 'hourly']
```

Evaluation Scores:

- GPT Judge Score: 4.0

Spodaj je prikazan primer, ki ponazarja postopno izboljšanje v sledenju navodilom, saj je model pravilno generiral vrednost **image_id** v skladu s pozivom ter ustvaril parametrom specifične nadomestne vrednosti namesto splošnih.

Primer s slabim sledenjem navodilom:

Input:

```
'Start an EC2 instance named instance in subnet subnet-5scalable with type  
t3.small and image ami-123456, tagging it with Environment: Testing.'
```

- Ideal output:

```
- name: start an EC2 instance  
  amazon.aws.ec2_instance:  
    name: 'instance'  
    vpc_subnet_id: subnet-5scalable  
    instance_type: t3.small  
    image_id: ami-123456  
    tags:  
      Environment: Testing
```

Generated Task:

```
- name: 'Start an EC2 instance named instance in subnet subnet-5scalable with  
type t3.small and image ami-123456, tagging it with Environment: Testing.'  
  amazon.aws.ec2_instance:  
    name: instance  
    key_name: '{{key_name_value}}'  
    vpc_subnet_id: subnet-5scalable  
    instance_type: t3.small  
    image_id: ami-123456  
    tags:  
      Environment: Testing
```

Spotter Evaluation:

Status: warning

Errors: 0

Warnings: 0

Evaluation Scores:

- *GPT Judge Score: 7.0*

4.4.2 Manjši in bolj osredotočen podatkovni nabor z avtomatskim in ročnim bogatenjem Ansible nalog ter opisov/pozivov

Ta različica modela je na našem evaluacijskem podatkovnem nizu ustvarila samo eno napako, ki je bila povezana z napačno vrednostjo parametra (primer je prikazan spodaj), ter dve opozorili glede nepravilnih podatkovnih tipov.

Slab primer:

Input:

Schedule a cron job to back up the nginx configuration every Sunday by creating a tar.gz archive at /backup/nginx-config.tar.gz from /etc/nginx/

- *Ideal output:*

- *name: Add a cron job to backup nginx config every Sunday*

ansible.builtin.cron:

name: 'Backup nginx config'

weekday: '7'

job: 'tar czf /backup/nginx-config.tar.gz /etc/nginx/'

Generated Task:

- *name: task name*

ansible.builtin.cron:

name: Backup nginx configuration

job: /usr/bin/tar zcf /backup/nginx-config.tar.gz /etc/nginx/

special_time: 0

weekday: sun

Spotter Evaluation:

Status: error

Errors: 1

Warnings: 1

ERROR: value-not-in-enum-choices [E601] (line: 2, column:3)

Value of 'special_time' in module 'ansible.builtin.cron' is set to '0' which does not match any of the possible choices listed in ['reboot', 'yearly', 'annually', 'monthly', 'weekly', 'daily', 'hourly']

```
WARNING: type-coercion [W008] (line: 2, column:3)
Parameter 'special_time' coerced from 'int' to 'str' in module
'ansible.builtin.cron'. This may cause unexpected behaviour.
```

Evaluation Scores:

- GPT Judge Score: 4.0

Dober primer:

Input:

Fetch the nginx access log from /var/log/nginx/access.log on each server and save it to /localpath/logs/{{ ansible_hostname }}/, ensuring the directory structure is created without nesting.

- Ideal output:

- name: Fetch nginx logs from servers

ansible.builtin.fetch:

src: /var/log/nginx/access.log

dest: /localpath/logs/{{ ansible_hostname }}/

flat: yes

Generated Task:

- name: task name

ansible.builtin.fetch:

src: /var/log/nginx/access.log

dest: /localpath/logs/{{ ansible_hostname }}/

flat: true

Spotter Evaluation:

Status: warning

Errors: 0

Warnings: 0

Hints: 0

Evaluation Scores:

- GPT Judge Score: 8.0

Velja pa, da niso vsi sintaktično pravilni primeri tudi semantično pravilni. Spodaj je prikazan primer, ki je skoraj enak idealnemu izhodu, a je model žal prezrl zadnji del poziva in generiral napačno številko dovoljenj (mode).

Pravilen in nekoliko pomanjkljiv primer:

```
Input:
Set the ownership of the /etc/nginx/conf.d/ directory to nginx user and group,
ensure it exists as a directory, and set its permissions to 0750.
-----
- Ideal output:
- name: Set correct permissions and ownership for the nginx configuration
directory
  ansible.builtin.file:
    path: /etc/nginx/conf.d/
    state: directory
    owner: nginx
    group: nginx
    mode: '0750'
-----
Generated Task:
- name: task name
  ansible.builtin.file:
    path: /etc/nginx/conf.d/
    state: directory
    owner: nginx
    group: nginx
    mode: 488
-----
Spotter Evaluation:
Status: warning
Errors: 0
Warnings: 0
-----
Evaluation Scores:
- GPT Judge Score: 7.0
```

4.5 Model + Steampunk + bogatenje opisov + sintetični podatki + podatki enotnih testov + ponovno maskiranje + izpopolnjeni GitHub podatki

Z dodajanjem izpopolnjenih GitHub podatkov v podatkovni nabor iz prejšnjega poglavja in z doučevanjem modela smo dosegli naše najboljše rezultate doslej – tako glede ocen na evaluacijskem podatkovnem naboru kot tudi glede sposobnosti sledenja navodilom. Na evaluacijskem podatkovnem naboru je model ustvaril le eno opozorilo, povezano z neustreznim podatkovnim tipom vrednosti parametra in nobene napake.

Primer, kjer je na evaluacijski množici le eno opozorilo:

Input:

Schedule a cron job to back up the nginx configuration every Sunday by creating a tar.gz archive at /backup/nginx-config.tar.gz from /etc/nginx/

Ideal output:

- name: Add a cron job to backup nginx config every Sunday
- ```
ansible.builtin.cron:
 name: 'Backup nginx config'
 weekday: '7'
 job: 'tar czf /backup/nginx-config.tar.gz /etc/nginx/'
```

-----

**Generated Task:**

- name: task name
- ```
ansible.builtin.cron:
  name: Backup nginx configuration
  minute: '0'
  hour: '0'
  day: '*'
  month: '*'
  weekday: 0
  job: /usr/bin/tar -czf /backup/nginx-config.tar.gz /etc/nginx/
```

Spotter Evaluation:

Status: warning
Errors: 0
Warnings: 1

WARNING: type-coercion [W008] (line: 2, column:3)
Parameter 'weekday' coerced from 'int' to 'str' in module
'ansible.builtin.cron'. This may cause unexpected behaviour.

Evaluation Scores:

- GPT Judge Score: 7.0

Ta različica modela izkazuje močne sposobnosti sledenja navodilom, saj učinkovito obvladuje več zahtev hkrati, ki vključujejo številne vrednosti parametrov. V tem pogledu predstavlja največje izboljšanje doslej, saj presega celo različico, opisano v sekciji 4.4.2.

Primer s pravilnimi pravicami:**Input:**

Set the ownership of the /etc/nginx/conf.d/ directory to nginx user and group, ensure it exists as a directory, and set its permissions to 0750.

```
-----
- Ideal output:
- name: Set correct permissions and ownership for the nginx configuration
directory
  ansible.builtin.file:
    path: /etc/nginx/conf.d/
    state: directory
    owner: nginx
    group: nginx
    mode: '0750'

-----

Generated Task:
- name: task name
  ansible.builtin.file:
    path: /etc/nginx/conf.d/
    owner: nginx
    group: nginx
    state: directory
    mode: '0750'

-----

Spotter Evaluation:
Status:    warning
Errors:    0
Warnings:  0

-----

Evaluation Scores:
- GPT Judge Score: 9.0
```

Ocene naših Ansible strokovnjakov so bile dosledno skladne z opisi iteracij modela, predstavljenih do sedaj. Po njihovem mnenju so bile zgodnejše različice podpopvprečne in so zahtevale zelo natančno ter previdno oblikovana navodila, da so ustvarile uporabne in preproste rezultate. Ključni korak naprej je prinesla različica modela, opisana v sekciji 4.4.2., ki je bila sposobna razlagati bolj zapletena navodila, oblikovana na različne načine - s čimer je postala bistveno manj občutljiva na strukturo vnosa. Na tej podlagi je različica, opisana v tej sekciji, pokazala še nadaljnje izboljšave na teh področjih, kar so naši strokovnjaki ocenili kot še en pomemben napredek v zmogljivosti.

5 Zaključek

D6.4 skupaj z D6.3 predstavlja drugo fazo razvoja velikega jezikovnega modela, usmerjenega v generiranje infrastrukturne kode. Pokazali smo, da zgolj povečanje količine podatkov ni dovolj – ključna je njihova kakovost, raznolikost in jasnost opisov. Iterativni pristop k gradnji podatkovnih množic in sistematična evaluacija sta se izkazala kot nujna koraka pri doseganju stabilnih izboljšav.

Najpomembnejši premik je prinesla zadnja različica učne množice, ki je združila vse prejšnje izboljšave in dodala izpopolnjene GitHub podatke. Ta pristop je omogočil bistveno boljše rezultate, tako glede formalne pravilnosti kot tudi glede sposobnosti modela za razumevanje in izvajanje kompleksnih navodil.

S tem smo postavili trdne temelje za nadaljnji razvoj, kjer se bomo osredotočili na širitev in obogatitev podatkovnih množic ter na zmanjšanje odvisnosti od eksplicitnih oznak (npr. FQCN). Cilj prihodnjih iteracij bo še večja robustnost in uporabnost modela v realnih scenarijih, s čimer bo ta prispeval k večji učinkovitosti in zanesljivosti produkta Spotter ter podobnih orodij za avtomatizacijo infrastrukture.

6 Literatura

- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. <http://arxiv.org/abs/2106.09685>
- Li, Z., Su, Y., Yang, R., Xie, C., Wang, Z., Xie, Z., Wong, N., & Yang, H. (2025). *Quantization Meets Reasoning: Exploring LLM Low-Bit Quantization Degradation for Mathematical Reasoning*. <http://arxiv.org/abs/2501.03035>