Софийски университет “Св. Климент Охридски”

Факултет по математика и информатика

**Курсов проект**

По "Граф бази данни”

Валидация на достоверност на информация

Изготвили:

Стоян Николов, Технологии за големи данни

Николай Шиваров, Технологии за големи данни

Съдържание

[Въведение 2](#_Toc169783272)

[Основни функционалности 2](#_Toc169783273)

[1. Въвеждане на текст и извличане на фактите от него 2](#_Toc169783274)

[2. Мапване на триплите към Wikidata обекти и релации и проверка на тяхната достоверност 2](#_Toc169783274)

[3. Forward chaining 2](#_Toc169783274)

4. Добавяне на подобни релации 2

Как да се подкара 3

## 

## Въведение

Целта на проекта е да се разработи софтуерно приложение, което валидира текст чрез съпоставяне на фактите в него с факти в Wikidata.

## Основни функционалности

### Въвеждане на текст и извличане на фактите от него

Ползвали сме VsCode. Приложението се стартира в командния ред. Потребителите въвеждат текстова информаци. Този текст се обработва автоматично, за да се извлекат фактите в него под формата на трипли. Това става с помощта на голям езиков модел. В случая сме използвали llama3-70b чрез LlamaAPI с намалена температура до 0.1 за по добра генерация. За генерирането на фактите сме използвали следните инструкции:

* For the given text provide all concepts and relations between them. Ignore the correctnes of the text. Present the result in a python list of tuples - subject, relation, object. Please use Wikidata labels for the results. Don't use Wikidata IRIs for the result. Don't add comments.

Като след това се добавя текста, който потребителя е въвел. Големият езиков модел обработва текста и връща отговор под формата на трипли.

### Мапване на триплите към Wikidata обекти и релации и проверка на тяхната достоверност

След като вече сме получили триплите с помощта на функция ги мапваме към съществуващи wikidata обекти и релации. После проверяваме тяхната достоверност с друга функция, която проверява за всеки subject predicate дали object е в съответния за предиката списък. Резултата се генерира във файл result.txt, където се вижда дали факта е правилен.

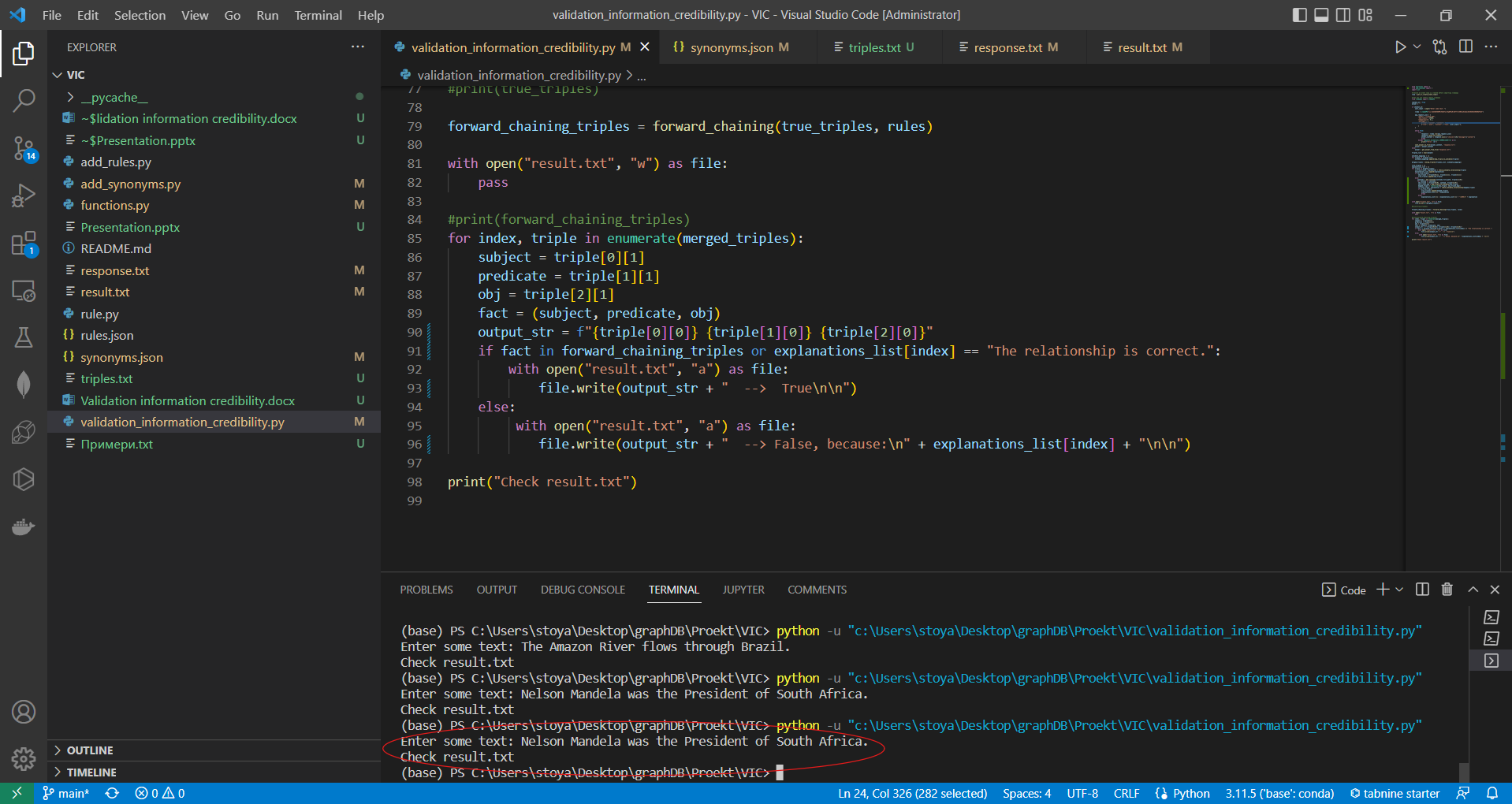
### Forward chaining

В някои случаи може да има факти, които следват от други факти. Например, когато кажем „Sofia is the capital of Bulgaria“, ще получим отговор ‘Sofia capital of Bulgaria’ – TRUE ‘Sofia is capital’ – FALSE. Това се случва, тъй като втория факт го няма в wikidata, но всъщност той е верен и следва от първия. За да решим този проблем сме създали файл “rules.json”, където пазим правила от такъв тип. Всяко правило се състои от n на брой условия и един извод, като всички условия и изводи са трипли. Добавянето на нови правила е изключително лесно за user-a, като той просто трябва да изпълни файла “add\_rules.py” и да въведе условията и извода. За прилагането на правилата използваме права изводимост(forward chaining) от логическото програмиране и така има шанс да намерим нови верни факти, които ги няма в wikidata.

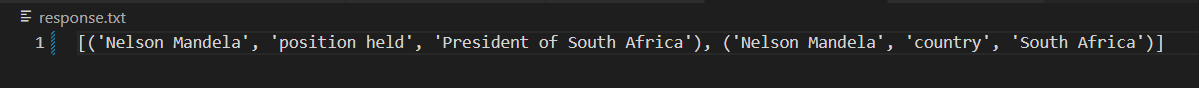
1. Добавяне на подобни релации

Имаме добавена функционалност, която добавя трипли с близки релации на генерираните от езиковия модел, които, ако са верни значи и първоначалния факт е верен. Например за релация location добавяме релация country, защото в wikidata в локацията нямаме информация за странната, където се намира.

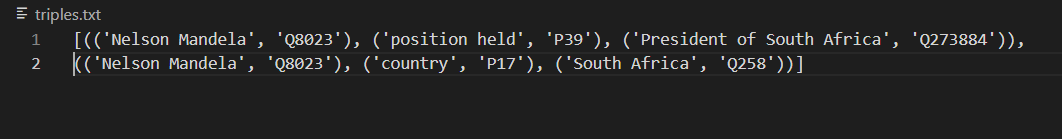
Как се подкарва

Пуска се validation\_information\_credibility.py Въвежда се текст в терминала. И се проверяват резултатите в results.txt.

Така изглежда отговора на езиковия модел:



Така изглеждат мапнатите трипли:



Така изглежда резултата от изпълнението на програмата:

