Молим вас да не будете превише критични према могућим граматичким грешкама, јер у свакодневном говору истовремено користим три језика :)

Virtual Buddy

План развоја корак по корак

1

Припрема документа:

- **Дељење на chunks:** Дужи документи треба поделити на мање делове величине 100–1000 речи, како би се омогућило ефикасно претраживање и обрада(**NLTK** или **SpaCy**).
- **Креирање embeddings:** Коришћење моделе (**OpenAI Emb., fastText, Sentence Transformers**) за претварање текстуалних chunks у векторске репрезентације које одражавају њихов семантички садржај. Коришћење **MTEB**.
- **Метаподаци:** Сачувајемо информације о структури документа (на пример, наслове, одељке) у метаподацима ради побољшања претраге и навигације.
- **Чување у векторској бази података:** Сачувајемо векторске репрезентације у специјализованој бази података (**Weaviate**, **Chroma**) ради брзог претраживања по сличности.

2

Избор и имплементација претраживања:

- **Векторска претрага:** Коришћење густински претраживач (dense retriever) који проналази релевантне делове текста на основу семантичке сличности у векторском простору (**FAISS** или **Weaviate**).
- **Индексирање:** Након векторизације, учитајемо векторе у векторску базу података, омогућавајући брзу претрагу.
- **Претрага релевантних делова:** Треба претворити упит у векторски облик и извршити претрагу најближих суседа у бази података како бисте пронашли најрелевантније chunks.

3

Избор генеративног модела:

- **Модели:** Размотримо употребу модела као што су **GPT** или **Llama**, који подржавају генерисање текста. Ови модели су способни да генеришу кохерентне и контекстуално релевантне одговоре ћирилицом и латиницом, посебно када имају додатне информације из retrieval-компоненте. Они су обучени на великим корпусима података на различитим језицима, што омогућава генерисање тачног одговора. Такође важна је могућност финог подешавања (fine-tuning) омогућава прилагођавање модела специфичним доменима и интеграцију са системима за претраживање информација.
- **Имплементација система:** Обједињавање са retrieval-системом, прослеђивање извучених chunks заједно са корисничким упитом генеративном моделу ради формирања коначног одговора.

4

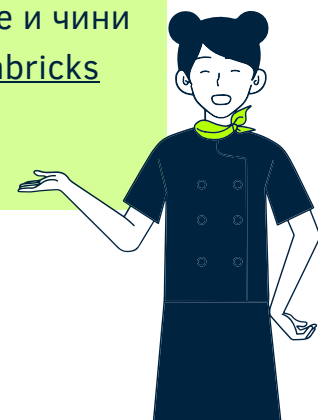
Архитектура система:

- **Модул за претходну обраду података:** Одговоран за чишћење и сегментацију текстова.
- **Векторско складиште:** Чува векторске репрезентације и омогућава брзо претраживање.
- **Модул за претраживање:** Извлачи релевантне чанкове на основу упита.
- **Генеративни модел:** Формира одговор користећи упит и извучене податке.
- **Коришћење фрејмворка:** Размотримо примену **LangChain** и **LlamaIndex** за поједностављење интеграције компоненти и оркестрације процеса.

5

Процена и метрике:

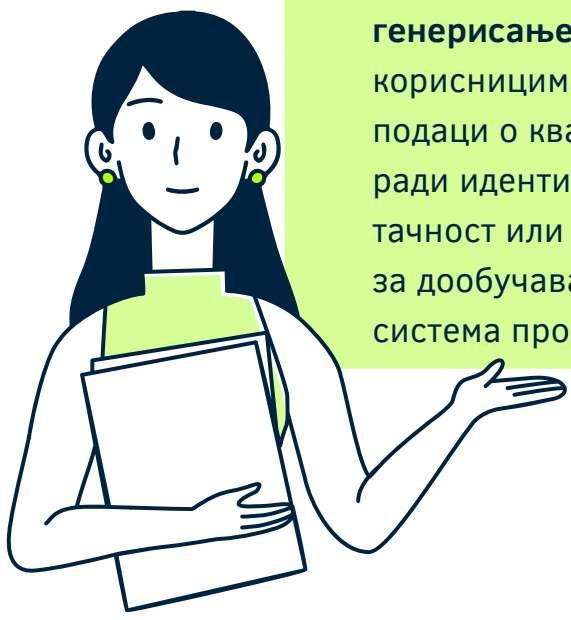
- **Оцењивање компоненти претраживања:** Колико добро систем проналази релевантне информације у бази знања? За то се користе стандардне метрике претраживања, као што су **DCG** и **nDCG**, које процењују квалитет рангирања резултата.
- **Оцењивање компоненти генерације:** Користимо другу велику језичку моделу као „судију“. Ова LLM ће анализирати одговоре RAG-система и дати оцену њиховог квалитета. Овакав приступ омогућава аутоматизацију евалуације и чини је објективнијом. Примере промптова за евалуацију: Prometheus или Databricks (MLFlow).
- **RAGAs:** Семантичка сличност одговора, тачност одговора.



6

Оптимизација и унапређење система:

- **Побољшање тачности проналажења и релевантности одговора током времена:** Редовно дообучавање модела на актуелним подацима специфичним за предметну област како би се повећала њихова способност проналажења релевантних информација. Коришћење хибридних метода претраге, комбиновањем семантичке и лексичке претраге, како би се обезбедило прецизније проналажење информација. Периодична провера и ажурирање базе знања, уклањањем застарелих информација и додавањем нових података, како би се одржала актуелност одговора.
- **Повећање брзине одговора и смањење кашњења:** Коришћење високоперформансних сервера и дистрибуираних рачунања за убрзање обраде захтева. Имплементација кеширања често тражених података како би се смањило време обраде поновљених захтева. Увођење паралелне обраде задатака, као што су истовремено проналажење и генерисање, ради смањења укупног кашњења система.
- **Увођење паралелне обраде задатака, као што су истовремено проналажење и генерисање, ради смањења укупног кашњења система:** Омогућавање корисницима да оцењују одговоре и остављају коментаре како би се прикупили подаци о квалитету рада система. Анализа добијених повратних информација ради идентификовања образаца и области које захтевају побољшање, као што су тачност или брзина одговора. Коришћење прикупљених повратних информација за дообучавање модела и корекцију алгоритама, обезбеђујући прилагођавање система променљивим потребама корисника.



Напред! У свет високих технологија!
Хвала на пажњи!