



Semestrální práce z KKY/HDS

Tvorba hlasového dialogového systému

Dominik Zappe – A23N0011P
(zapped99@students.zcu.cz)

Květen 2024

Obsah

1	Zadání	1
2	Příprava dat (SP3)	2
3	Implementace dialogového manažera (SP4)	3
3.1	Detekce důležitých částí otázky	5
3.2	Transformace detekovaných částí	5
3.3	Problém s API pro rok 2024	5
4	Závěr	6

1 Zadání

Obecné požadavky

Připravte data pro natrénování jazykového modelu pro úlohu rozpoznávání s velkým slovníkem (LVCSR). Cílová doména dialogového systému bude z oblasti **statistiky dopravních nehod v ČR**, případně vyučujícím schválené individuální zadání.

Vytvořte hlasový dialogový systém využívající systém rozpoznávání řeči natrénovaný z dat vytvořených v rámci SP3. Hlasový dialogový systém bude podávat informace o statistice dopravních nehod, přičemž bude využívat statistiky Policie ČR přístupné z oficiálních stránek, příp. ze stránek Českého rozhlasu.

Pro řešení použijte hlasovou platformu SpeechCloud, jako programovací jazyk můžete zvolit JavaScript nebo Python. Při implementaci postupujte dle dokumentace k SpeechCloud API.

Při návrhu a implementaci dodržujte následující pokyny:

- Navrhněte hlasový dialog se smíšenou iniciativou a s řízením založeným na rámcových strukturách
- Porozumění řeči vhodně implementujte buď vlastním algoritmem nebo s využitím SpeechCloud API
- V dokumentaci popište vámi použitý přístup, vhodně doplňte ilustrací, příkladem, vývojovým diagramem
- Dbejte na správně formulované výzvy, potvrzování stavu dialogu, nápovědy a ošetření situací, kdy uživatel nereaguje
- Při implementaci používejte standardní výstup pouze pro zobrazení logovacích zpráv, hlasový dialog musí být možné realizovat i bez zobrazení tohoto logu (tj. aniž by jej uživatel sledoval)

Požadovaná funkcionalita

Požadavky na funkcionalitu výsledného dialogového systému (kurzívou příklady promluv):

- Přehled nehod v republice pro daný den
 - Kolik nehod bylo včera?
- Přehled nehod v konkrétním kraji

- Kolik nehod bylo v pondělí v Plzeňském kraji?
- Zodpovězení otázek na jednotlivé statistiky
 - Byly včera nehody s těžkým zraněním?
- Změna aktuálního stavu dialogu, např. jiný kraj, jiný den.
 - Kolik lehce zraněných bylo včera? (...) A kolik předevčírem?
 - Jaké škody byly v Moravskoslezském kraji? (...) Jaké byly v Plzeňském kraji?
- Doplnění nezbytných informací do stavu
 - Kolik nehod bylo v Plzeňském kraji?
[Který den vás zajímá?]
Pondělí.

Požadované sémantické entity

Je třeba rozumět následujícím entitám:

- dny v týdnu (po-ne), relativní den (včera, pozítří)
- kraje
- sledované statistiky (počet mrtvých, těžce/lehce zraněných, jízda pod vlivem, nedání přednosti, nesprávné předjíždění, nepřiměřená rychlost, nesprávný způsob jízdy, jiné příčiny, škody)
- dotazy na sledované statistiky (Kolik bylo ...?) vč. zjišťovacích otázek (s odpovědi ano/ne)

2 Příprava dat (SP3)

V rámci SP3 bylo připraveno několik textových souborů s vymyšlenými a vygenerovanými větami. Konkrétně byl vytvořen soubor pro pozdravy, začátky otázek, typy nehod, údaje o místě (kraje) a údaje o čase (měsíce, dny, dekorátory dnů a čísla – datумы). Většina těchto dat byla vymyšlena nebo vygenerována pomocí ChatGPT 3.5 a GitHub Copilot Chat. Následně byl navržen postup, jak tyto jednotlivé údaje spojovat do vět – skript byl napsán v Pythonu. Jednalo se prakticky o kombinatorickou explozi, kde ne každá otázka musí obsahovat pozdrav, každá otázka také nemusí obsahovat údaj o místě, či naopak o čase. Čas celkově může být nahrazen jak dnem, tak dekorovaným dnem, měsícem či dekorovaným měsícem, taktéž je možné zeptat se pomocí datumu. Dekorátorem jsou zde myšleny slova jako „minulý“, „předminulý“ atp.

3 Implementace dialogového manažera (SP4)

Veškerá implementace je v jazyce Python ve verzi 3.11. Implementovaný manažer běží na virtuálním stroji na IP adrese 147.228.173.117. Klient je přístupný na adrese <https://speechcloud.kky.zcu.cz:9444/index.html?edu-hds-zappe>.

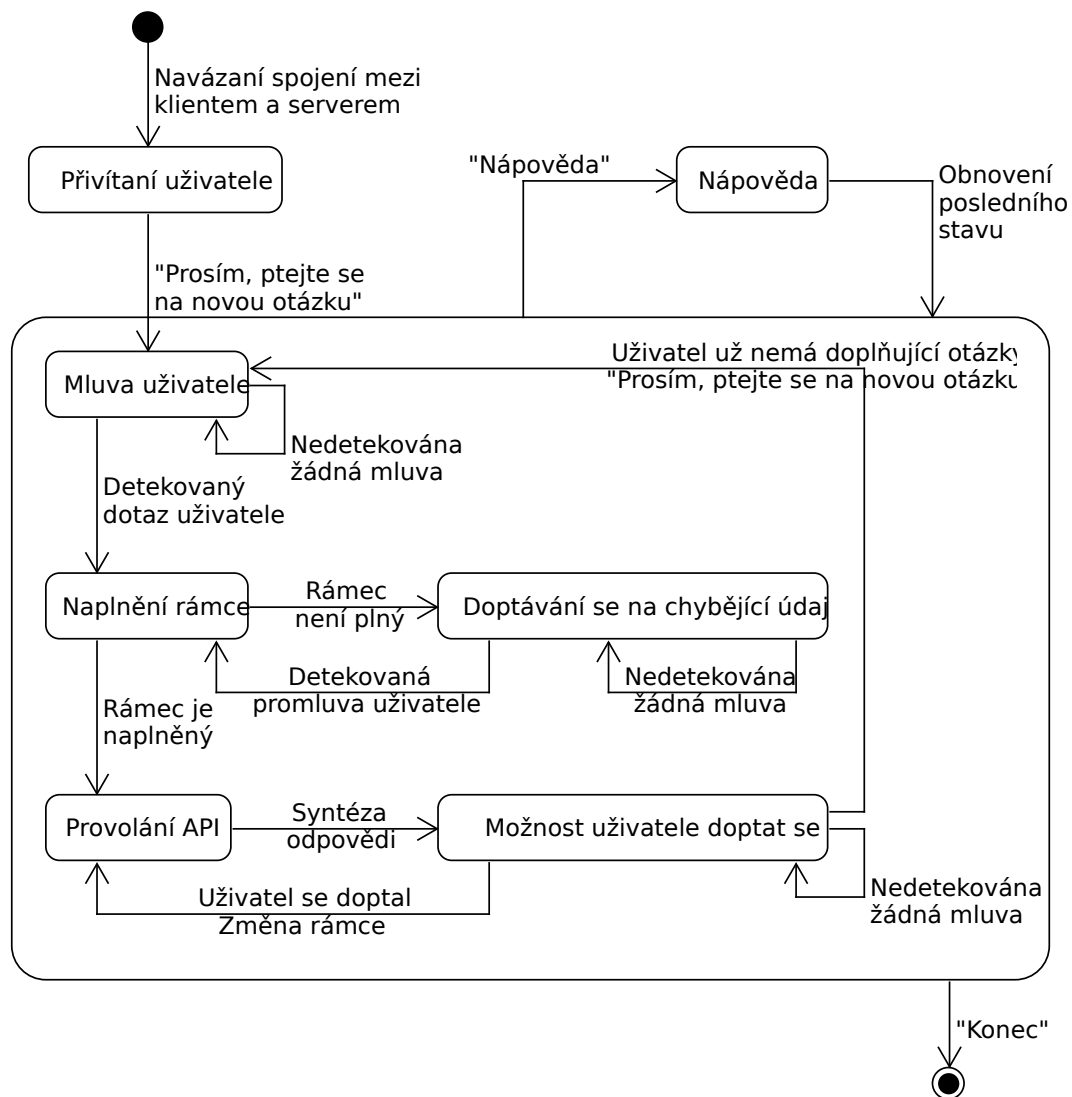
Dialogový manažer je navržený tak, že hlasový dialogový systém nejprve uživatele přivítá a po zaznění „Prosím, ptejte se na novou otázku“ předává iniciativu uživateli. Následně jsou detekovány z načteného uživatelského vstupu čtyři věci:

- Typ nehody
- Místo
- Čas
- Zda-li se jedná o otázku zjišťovací (odpověď ano/ne)

Nejsou-li detekovány všechny údaje, hlasový dialogový systém o tom dá uživateli vědět a konkrétně mu řekne, co mu chybí, a bude se doptávat tak dlouho, dokud si rámec o těchto čtyřech údajích nedoplní. Po naplnění rámce je poslán dotaz na API ze zadání a uživatel je o tom informován odpovědí na jeho otázku. Následně se hlasový dialogový systém uživatele zeptá, zda-li má ještě nějaké doplňující otázky. Pokud se uživatel dále bude doptávat, bude se dialog takto cyklit tak dlouho, jak uživatel bude chtít. Až uživatel odpoví záporně, na doptávající se otázku, tedy např. „Ne, už nemám další otázky.“ – rámec se vynuluje a dialog se opakuje – tedy zazní „Prosím, ptejte se na novou otázku“.

V průběhu dialogu je kdykoliv možné se zeptat na nápovědu nebo dialog ukončit. Je to intuitivně možné např. pomocí „Nápověda“, „Potřeboval bych pomoc“, nebo „Ukončit“, „Konec“ atp.

Výše popsaná dialogová smyčka je znázorněna na diagramu na Obrázku 1



Obrázek 1: Hlavní smyčka hlasového dialogového systému

3.1 Detekce důležitých částí otázky

V implementaci nebylo užito SLU gramatik. Osobně mi přišla implementace pomalá a navíc při užití SLU funkcí není možné dostat „raw string“ toho, co uživatel doopravdy řekl – není tedy možné rozeznat mezi tichem a neznámými slovy.

Otročinné práci s přepisováním textových souborů ze SP3 se bohužel vyhnout nedalo. Nebyla natvrdo definována SLU gramatika, ale bylo užito podobných konstrukcí, které jsou řízeny ručně. Tomuto účelu slouží soubor **data.py**. V tomto zdrojovém souboru je přepsána většina obsahu textových souborů ze SP3 – protože je nutné detekovat v uživatelském dotazu správně místo, čas a typ nehody. Pro detekci byly použity datové struktury *pole* a *mapy*. Detekce typu otázky – obyčejná vs. zjišťovací – probíhá detekcí slov „kolik“, „počet“ a „jaká“. Pokud v otázce tato slova nejsou, automaticky se otázka považuje za zjišťovací – tedy odpověď ano, ne.

Samotná detekce pak probíhá porovnáváním známých řetězců s detekovanou uživatelskou promluvou. Pokud se v promluvě například nachází nějaký známý tvar slova „Plzeň“, je detekováno místo jako „Plzeňský kraj“. Jedná se víceméně o přístup hrubou silou a možnými vylepšeními do budoucna by mohly být lepší metody zpracování přirozeného jazyka – minimálně alespoň lepší preprocessing a normalizace (např. lematizace či stemming).

3.2 Transformace detekovaných částí

Transformace místa a typu nehody je jednoduchá, neboť API přímo definuje vlastní zkratky, které musí být užity. Transformace času je však o něco složitější, už jen proto, že čas může být zadán různě – „dnes“, „předevčím“, „minulý rok“, „předminulý únor“ atp. Pro tento účel byla definována řada transformačních funkcí ve zdrojovém souboru **utils.py**. Veškerá transformace času probíhá poměrně naivně a metodou hrubé síly – bohužel nebylo nalezeno rozumnější řešení. Datумы nejsou podporovány vůbec, neboť detekce uživatelské řeči nevrací datумы v nějaké rozumné formě.

3.3 Problém s API pro rok 2024

Formát odpovědi API, které má být v práci použito, byl změněn po roce 2024. Hlasový dialogový systém na tuto skutečnost uživatele upozorní při uvítání a při nápovědě taktéž. Problém byl vyřešen odečtením roku o jedna, tedy automaticky se jakéhokoliv dne stává o rok starší datum – např. dnes je 25. 5. 2024 → 25. 5. 2023.

4 Závěr

Výstupem semestrální práce je hlasový dialogový systém, který odpovídá na otázky o nehodách v ČR pomocí rámců.

Osobně pro mě byla semestrální práce spíše otrockou prací, kde bylo nutné natvrdo definovat mnoho výrazů a posléze je hrubou silou hledat v uživatelských promluvách a na základě nich reagovat a upravovat si rámce.

Práce by se do budoucna dala vylepšit sofistikovanějším způsobem detekce důležitých částí v otázkách a následnou lepší transformací. Dále by se práce dala rozšířit o detekci datumů a následnou práci s nimi.