Diplomová práce

Reprezentace stavu hlasových dialogových systémů/State representation for spoken dialog systems

Filip Polák

Akademický rok 2020/2021

Poděkování

1 Abstrakt

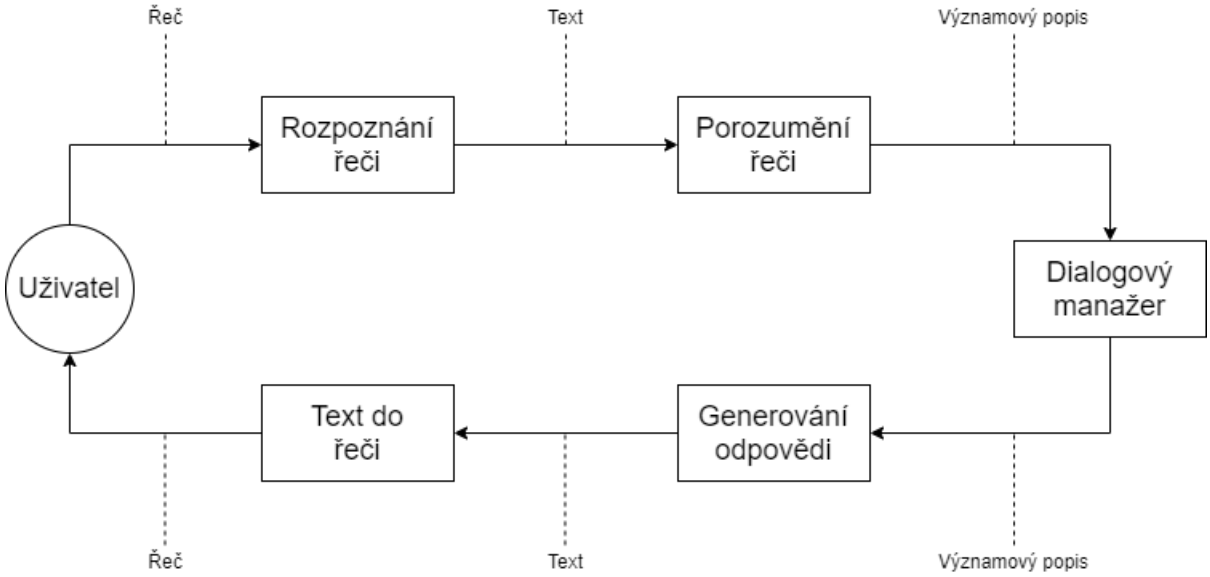
Hlasové dialogové systémy slouží k zprostředkování komunikace mezi člověkem a počítačem. Tato komunikace může ulehčit život nejenom tělesně postiženým lidem (slepota, tělesné postižení), ale i aktivitám v běžném životě, jako například zarezervování místa v restauraci, chytrá domácnost nebo zakoupení letenky. Pro správnou funkčnost těchto systémů je klíčová správně reprezentovat jejich stav, což je hlavním cílem této diplomové práce. Reprezentaci stavu si lze představit jako představu systému o tom, co uživatel řekl a co je jeho cílem. V práci jsou uvedeny různé metody a přístupy k reprezentaci stavu. Poté je provedena analýza softwarové knihovny jazyka Python sloužící k vhodné reprezentaci hlasového dialogového systému, jejíž funkčnost je poté demonstrovaná na několika hlasových systémech.

Obsah

2 Úvod

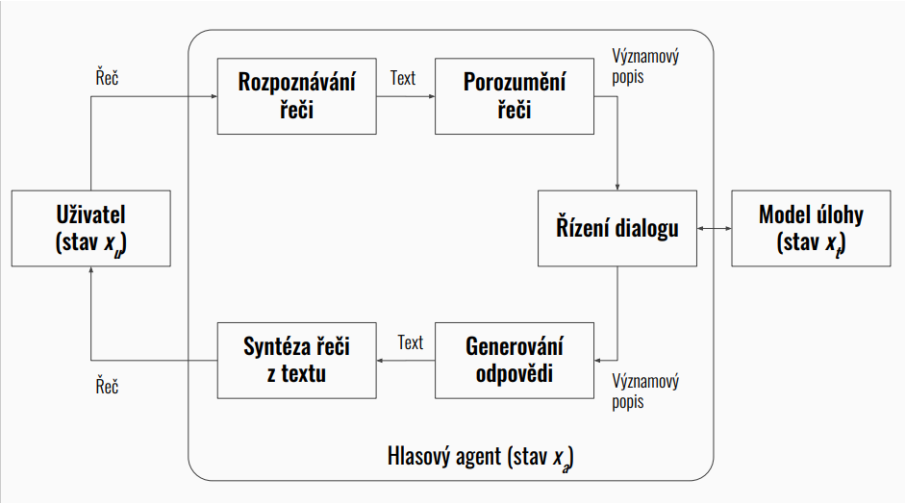
Komunikace (z latinského *communicare* - sdílet), je proces, kdy si dvě entity předávají informace za účelem splnění určitého cíle. Komunikace, a především ústní komunikace, je základním pilířem každé vyspělé společnosti. O tom, kdy spolu lidé začali komunikovat slovy a ne jenom gesty, se stále vedou debaty. První předpoklady zasazovaly začátek ústní komunikace cca 50.000 let do minulosti, kdy se na světě už několik tisíc let objevoval moderní člověk homo sapiens. Novější výzkumy s pomocí skenů mozků (článek https://www.sciencemag.org/news/2013/08/striking-patterns-skill-forming-tools-and-words-evolved-together) ovšem ukazují, že existuje korelace mezi výrobou nástrojů a vytvářením slov. Lidé tedy spolu mohli komunikovat pomocí slov již před 2 miliony lety, kdy se v Africe procházeli zástupci homo erectus.
Hlasové dialogové systémy přesouvají verbální komunikaci z domény člověk-člověk do domény člověk-počítač. Od nastavování budíku či zjištění informací o počasí po asistované učení matematiky nebo zakoupení letenky, hlasové dialogové systémy jsou omezené pouze výpočetní náročností, která se v dnešní době pojí s jejich implementací. Za několik desetiletí bude možné realizovat autopiloty jako ve Star Treku nebo operační systémy jako Jarvis v obleku Iron Mana. Nezávisle na doméně jsou hlasové dialogové systémy užívány z jednoduchého důvodu - využít v každodenních operacích možnost použít mnohonásobně rychlejší alternativu. Místo toho, aby člověk zašel stáhnout žaluzie, stačí, aby to řekl rychlé domácnosti. Místo toho, aby člověk musel zdlouhavě prohlížet výkony svých akcií, stačí, aby se na jejich stav zeptal hlasového dialogového systému, kterému by pak mohl přikázat, aby je případně prodal nebo ještě podržel. Hlasové dialogové systémy můžou být také využívány jako pomoc lidem s tělesnými postiženími, například slepota nebo ochrnuté končetiny.
Ačkoliv je komunikace pomocí řeči součástí každodenního života všech lidí, je důležité si uvědomit, že s sebou přináší i řadu úskalí. Jedním z nich je například jazyková bariéra, která narozdíl od gest brání komunikaci dvou lidí z opačných konců zeměkoule a není tedy překvapivé, že většina hlasových dialogových systémů nabízí možnosti alespoň těch největších světových jazyků. Dalším může tzv. sériová povaha předávaných informací - davkove dane info, muze se ztratit dulezite info. V neposlední řadě je dalším neduhem ústní komunikace omezení možnosti při přeskakování nedůležitých informací, které obírají o drahocenný čas jak lidi samotné, tak i hlasový dialogový systém.
Jedním z řešením těchto problémů je využití tzv. multimodálního přístupu k návrhu dialogu, který tyto problémy kompenzuje a nebo je plně odstraní - vyuziti propojeni vykonneho mobilu atd, vyuziti textu atd.

3 Hlasový dialogový systém



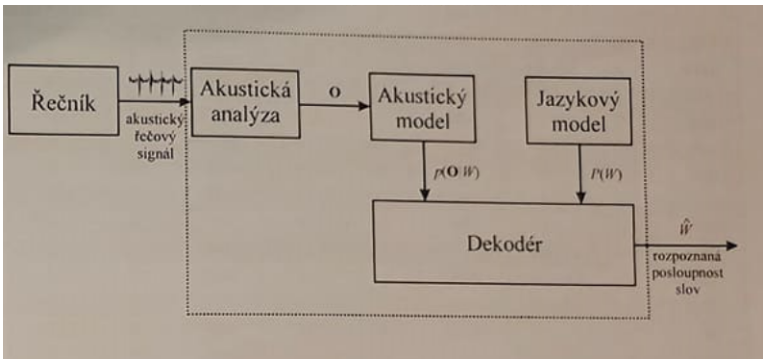
Obrázek 1: Klasické schéma hlasového dialogového systému

Schéma 1 představuje klasické schéma hlasového dialogového systému. **Uživatel** promluví, může jít o odpověď nebo otázku, provede se **rozpoznání řeči**, kde se řeč převede na text, ten je poté vstupem bloku **porozumění řeči**, jehož výstupem je sémantický tvar textu, neboli strojová reprezentace textu. Podle té řídí **dialogový manažer** samotný dialog a popřípadě řízenou úlohu a generuje významový popis, který je **generátorem odpovědi** převeden do textové odpovědi, která je poté **převedena do řeči** a prezentována uživateli, který na ni může podle potřeby reagovat.
Důležitým faktem, který je třeba zdůraznit, je, že hlasový dialogový systém se skládá ze "spojení"uživatele, hlasového agenta a řešené úlohy. Toto spojení lze pozorovat na schématu 2.



Obrázek 2: Spojení uživatele, hlasového agenta a řešené úlohy - TODO VLASTNÍ

3.1 Ropoznání řeči - Automatic Speech Recongnition (ASR)



Obrázek 3: Ropoznání řeči TODO VLASTNÍ

Rozpoznání řeči je převod řeči, respektive řečového signálu, na text.
Rozpoznávání řeči funguje podle následujicí rovnice 1:

$$W^* = \arg\max_W P(W|A) = \arg\max_W \frac{P(W,A)}{P(A)} = \arg\max_W \frac{P(A|W) \cdot P(W)}{P(A)} \quad (1)$$

kde A vyjadřuje posloupnosti vektorů pozorování, neboli posloupnosti akustických příznaků, W vyjadřuje posloupnost slov, $P(A|W)$ vyjadřuje akustický model, $P(W)$ vyjadřuje jazykový model, $\arg\max_W$ vyjadřuje prohledávácí strategii přes všechna "možná slova"a W^* vyjadřuje posloupnost slov, která řečovému signálu nejlépe odpovídala.
Výstupem rozpoznání řeči je tedy posloupnost slov, která měla podle analýzy největší

shodu s původním řečovým signálem, a nazývá se **1-best hypotéza**. Ostatní posloupnosti slov, které se s původním řečovým signálem neshodovaly úplně, se ovšem NEZAHAZUJÍ, ale je možné je taky využít. Z ropoznávání řeči se tedy vybírá **n-best hypotéz**, kde **n** odpovídá počtu posloupností slov, které postupují dále (do další analýzy?).

Akustický model (Acoustic Model - AM) modeluje pravděpodobnost pozorování A při hypotéze W, tedy $P(A|W)$. Zjednodušeně - jaká je pravděpodobnost akustického příznaku či posloupnosti akustických příznaků A, pokud bylo řečené určité slovo či posloupnost slov W. Model je trénován z dvojic *akustický signál - posloupnost akustických jednotek*, kde jsou akustické jednotky fonémy, grafémy, slabiky atd. a mění se podle typu modelu. Historicky byly akustické modely modelovány jako Gaussovské směsi představující pravděpodobnost pozorování, jimž následovaly skryté Markovské procesy (Hidden Markov Model), a aktuálně je nejčastější využití hlubokých neuronových sítí ve spojení s HMM.

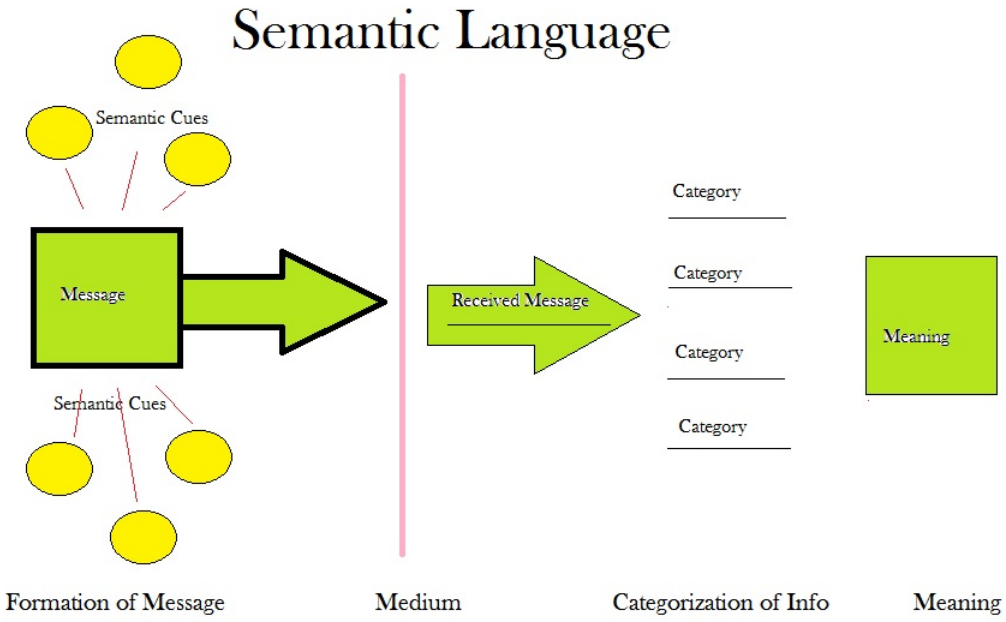
Jazykový model (Language Model - LM) modeluje apriorní pravděpodobnost posloupnosti slov W, k čemuž využívá n-gramové jazykové modely, kde **n-gram** představuje **n** po sobě jdoucích slov.

N-gramové jazykové modely jsou vyjádřeny pomocí rovnice 2:

$$P(W) = P(\omega_i, \omega_{i-1}, \omega_{i-2}...\omega_0) = \prod_{i=0}^{n} P(\omega_i|\omega_{i-1}, \omega_{i-2}...\omega_0) \approx \prod_{i=0}^{n} P(\omega_i|\omega_{i-1}, \omega_{i-2}...\omega_{i-N+1}) \tag{2}$$

Rovnice 2 tedy vyjadřuje, jaká je pravděpodobnost slova ω_i, pokud mu předchází posloupnost slov $\omega_{i-1}, \omega_{i-2}...$, což lze aproximovat pomocí pouze N předchozích slov. Jazykové modely jsou v dnešní době taktéž založeny na neuronových sítích a jsou trénovány z rozsáhlých textových korpusů.

3.2 Porozumění řeči - Spoken Language Understanding (SLU)



Obrázek 4: Porozumění řeči - TODO VLASTNÍ

V bloku porozumění řeči se řeč v podobě textu, která přišla z ASR bloku, převádí na tzv. významový popis řeči. Jde tedy o převod lexikální podoby řeči do strojové reprezentace, zjednodušeně - tomu počítač rozumí.
Porozumění rozpoznává globální význam promluvy, jedná se tedy o sémantické koncepty, a lokální význam, tedy sémantické entity. Sémantické koncepty vyjadřují, zdali je celá promluva nebo její část souhlasem/nesouhlasem, otázkou/odpovědí atd. Sémantické entity se zaměřují na jednotlivá slova či několik slov a zdali se jedná například o čas, datum, místo atd. Stejně jako v ASR je možné využít n-best hypotéz, díky čemuž se může zvýšit přesnost porozumění, jelikož se generuje několik alternativních hypotéz.
K modelování porozumění řeči se klasicky přistupuje dvěma způsoby, statisticky a znalostně. Tomu, než se ale jeden z těchto přístupů zvolí, ovšem předchází tzv. 0. krok, kde musí dojít k definici domény, ve které bude dialog probíhat, a doménově závislých konceptů a entit.
K definici domény patří pravděpodobně ještě důležitější krok a tím je definice uživatele, což může znít po definici domény jako nadbytečná a nepotřebná věc. Pokud přeci pracuje hlasový agent v určité doméně a je tím pádem omezen na velmi specifický obor otázek a odpovědí, není pak tolik podstatné, kdo je uživatelem a jak na hlasového agenta mluví. Naprostý opak je ovšem pravdou. Uživatel je nejdůležitější částí hlasového dialogového systému už jen z toho důvodu, že celá konverzace závisí na jeho otázkách a odpovědích. Proto je podstatné, aby byl v hlasovém dialogovém systému správně nadefinován uživatel a jeho možný způsob

mluvy. Tato definice může být provedena i pouze mentálně v hlavě konstruktéra systému, který se musí zamyslet, pro jaké uživatele svůj hlasový systém konstruuje. Jaký obor otázek bude systém "ochotný"zpracovávat, jestli bude zpracovávat pouze spisovnou nebo i hovorovou řeč, jestli bude chtít po každé operaci potvrzení uživatele a tak dále. Iron Manův oblek bude mít určitě jiného uživatele než uživatele zajímající se pouze o hlasové ovládání hlasitosti rádia.??????????
S tím souvisí, že se musí nadefinovat možný tvar vstupu od uživatele a tvar výstupu agenta, jež jsou dále použity v dialogovém manažeru.
Statistický přístup - TODO
Znalostní přístup - TODO

3.3 Dialogový manažer - Dialog Manager (DM)

Dialogový manažer (v angličtině Manažer i Management) je ve své podstatě mozkem celého hlasového dialogové systému, jelikož řídí konverzaci z pohledu hlasového agenta. Dialogoý manažer pozoruje stav uživatele prostřednictvím významového popisu řeči a na základě tohoto pozorování a aktuálního stavu agenta generuje významovou reprezentaci odpovědi, která je poté v dalších krocích převedena na řeč. Dialogový manažer ale jenom negeneruje samotnou odpověď, ale může ovlivňovat řešenou úlohu (zapne/vypne světlo, zvýší/sníží hlasitost, posune robota, stáhne si data z databáze) a ovlivňovat stav agenta. Řízení hlasového dialogového systému dělí dle použitých metod.

3.3.1 Podle iniciativy

Iniciativa hlasového agenta je takový způsob řízení dialogu, kdy agent pokládá otázky a nabízí uživateli způsob řešení úlohy. **Iniciativa uživatele** je řízení založené na příkazech od uživatele, který jimi přímo řídí agenta a řešenou úlohu. Kombinací obou předešlých přístupů je **smíšená iniciativa**, kdy může v jakékoliv fázi dialogu převzít iniciativu jedna či druhá strana. Uživatel například položí otázka (iniciativa uživatele), hlasový agent se může doptat na chybějící informace a prezentuje výsledek uživateli (iniciativa agenta) a uživatel se může ptát dále či s výsledkem dále pokračovat (iniciativa uživatele.)

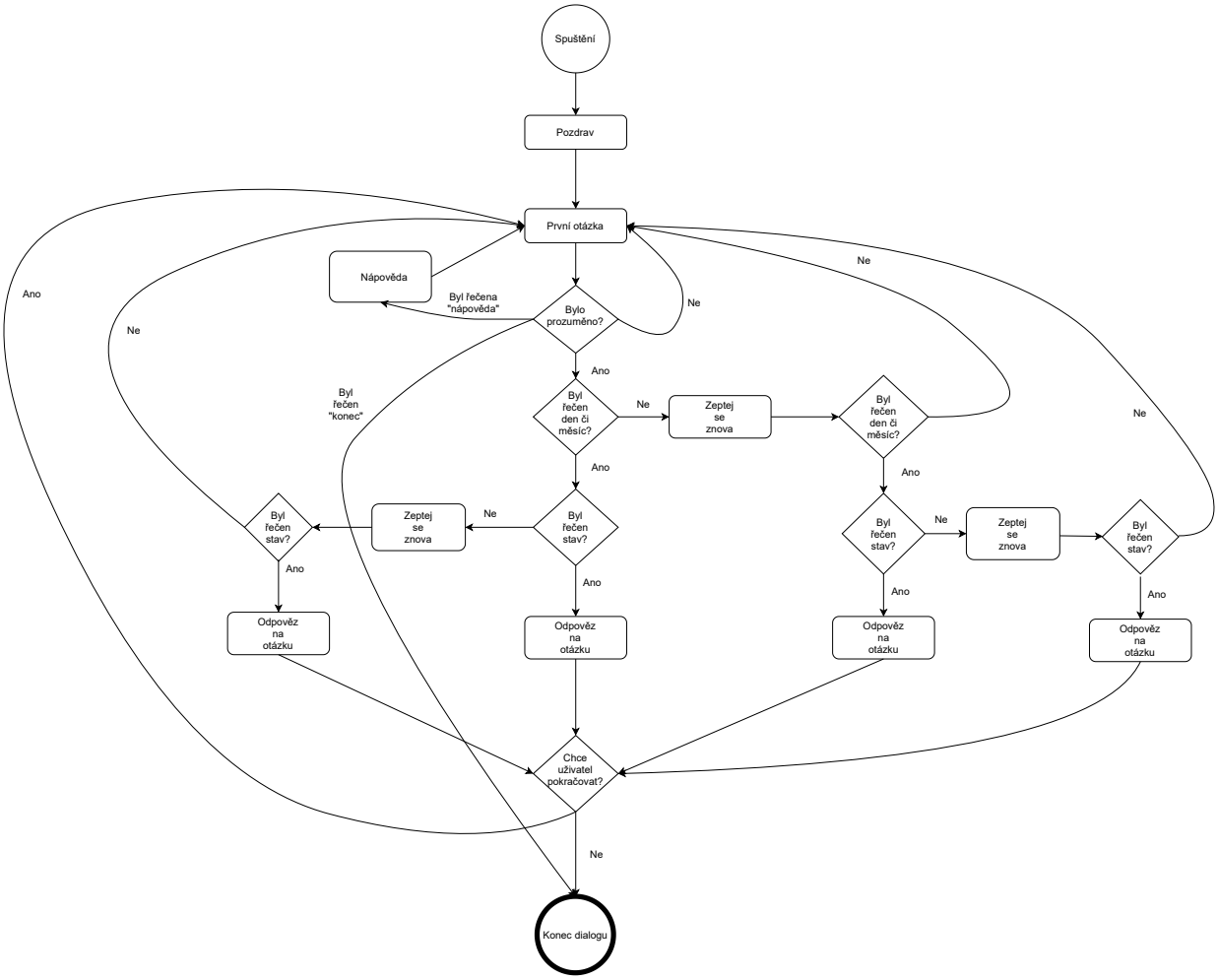
3.3.2 Podle struktury dialogu

Turn based dialog je, jak název napovídá, řízení dialogu prováděné po krocích či obrátkách (turn). Agent vždy čeká na úplný vstup uživatele (nemá tedy možnost tzv. barge-in - skoku do řeči) a až poté vygeneruje odpověď a odprezentuje ji uživateli. Jednu dialogovou obrátku v tomto přístupu představuje dvojice otázka uživatele + odpověď agenta. Stav hlasového dialogového systému se mění pouze v diskrétních krocích.
Přístup podobnější reálným konverzacím mezi lidmi je **inkrementální dialog**, kde agent průběžně zpracovává vstup uživatele a může na něj v případě nejednoznačnosti reagovat, neboli má možnost provést barge-in, jež má samozřejmě možnost provést

i uživatel. V teorii hlasových dialogových systémů se stav, kdy uživatel nebo agent mluví, nazývá, že uživatel nebo agent drží *floor*.

3.3.3 Řízení dialogu s konečným počtem stavů

Dialog je v tomto přístupu reprezentován jako orientovaný graf. **Uzly** představují stavy dialogu a **přechody** mezi uzly představují změnu stavu dialogu podmíněnou vstupem uživatele.



Obrázek 5: Řízení dialogu s konečným počtem stavů

3.3.4 Řízení dialogu založené na rámcových strukturách

Stav dialogu se v tomto přístupu nazývá **rámec** a lze ho v programovacích jazycích reprezentovat např. jako slovník nebo pole, kde každá položka pole/slovníku představuje jeden **slot**. Slot je vyplněn jednou nebo více hodnotami a ke každému slotu může ještě příslušet info o potvrzení a důvěra (confidence) systému v to, jestli bylo řečeno to, co si systém uložil. Slot je na základě přiřazených hodnot:

- **nevyplněný** - nemá hodnotu
- **nepotvrzený** - hodnota nebyla potvrzená uživatelem
- **potvrzený** - hodnota byla potvrzená uživatelem
- **nekonzistentní** - více možných hodnot

Každý slot má předem danou důležitost, což má vliv na algoritmus řízení dialogu:

1. Vyber povinné **nevyplněnné sloty** a dotaž se na jejich hodnoty
2. Vyber **nepotvrzené sloty** a vyžádej si potvrzení
3. Vyber **nekonzistentní sloty** a nabídni uživateli řešení nekonzistence
4. Z hodnot **potvrzených slotů** sestav řízení úlohy a na základě výstupu úlohy vyplň sloty
5. Prezentuj vybrané sloty uživateli

3.3.5 Promluvy v hlasových dialogových systémech

Při komunikaci s hlasových dialogovým systémem se odpovědi a otázky uživatele či agenta rozdělují na:

- **Vyzvání** druhé strany k předání informace a může být i doplněno, jaký druh informace se od druhé strany očekává.
  - "Pro **kolik** lidí chcete zarezervovat stůl?"
  - "Do jaké **destinace** chcete koupit letenku?"
  - "Na **kdy** chcete nastavit budík?"
- **Potvrzení** umožňující zjištění správnosti předané informace.
  - "Chcete rezervovat stůl pro **dva**, v **čínské** restauraci, v **20:00**?
  - "Chcete objednat letenku z **Praha** do **Berlín** v **pondělí**?
  - "Chcete nastavit budík na **7:00**?
- **Zotavení z chyby** pro opravení nebo opakování vstupu z důvodu nesprávně předané informace.
  - "Omlouvám se, nerozuměl jsem Vám."
  - "Prosím, zopakujte Vaši odpověď."
  - "Neslyšel jsem, do jaké **destinace** chcete koupit letenku."
- **Opětovná pobídka**, pokud byla očekávána promluva, ale ta nebyla předána.
  - "Řekněte mi, prosím, *pro **kolik** lidí chcete zarezervovat stůl*.

  - "Neřekl jste, *do jaké* ***destinace*** *chcete koupit letenku.*
  - "*Na* ***kdy*** *chcete nastavit budík?*"

- **Dokončení** pro předání požadované informace.
  - "Děkuji za potvrzení. Hledám stůl volný stůl pro **dva**, v **čínské** restauraci, v **20:00**."
  - "Letenka z **Praha** do **Berlín** v **pondělí** byla zakoupena."
  - "Budík byl nastaven na **7:30**."

3.3.6 Potvrzování v hlasovém dialogovém systému

Jelikož uživatel ani hlasový dialogový systém nejsou schopni pozorovat své stavy přímo, ale pouze prostřednictvím převodu stavu na řeč a přenosem řečovým kanálem, je důležité, aby hlasový agent udržoval uživatele "v obraze"ohledně svého vnitřního stavu, díky čemuž si může uživatel vytvořit mentální odraz toho, jak hlasový agent funguje a co od uživatele potřebuje. Toho je dosaženo pomocí výzev a potvrzování.

3.3.7 Uzavřené výzvy

Uživatel může volit pouze z nabízených možností.

- "Pro zrušení účtu řeknětě ano, nebo ne."
- "Pro potrvzení rezervace řeknětě ano, nebo ne."

3.3.8 Otevřené výzvy

Uživatel může odpovědět libovolně, ačkoliv je jeho možná odpověď omezena seznamem příkladů, které zná i hlasový agent. Agent může seznam uživateli seznam příkladů říct??? několika způsoby

- **okamžitě** - "Dobrý den, můžete se zeptat na faktury nebo stav účtu."
- **po prodlevě** - "Dobrý den,..."- 3s - "Můžete se zeptat na faktury nebo stav účtu."
- **na vyžádání uživatele** - "Na co se můžu zeptat?"

3.3.9 Explicitní potvrzování

Hlasový agent očekává od uživatele odpověď **ano** či **ne**.

- "Chcete odstranit seznam skladeb?"
- "Chcete zapnout světlo"

3.3.10 Implicitní potvrzování

Agentem požadovaná informace je představena jako součást výzvy.

- "Řekněte jméno seznamku skladeb, který chcete odstranit."
- "Řeknětě, do jakého typu restaurace chcete udělat rezervaci."
- "Řekněte, na kdy chcete nastavit budík."

3.4 Generování odpovědi - Natural Language Generation (NLG) a Text do řeči - Text-to-speech (TTS)

Generování odpovědi a text do řeči úzce navazují na výstup dialogového manažera, jímž je významový popis řeči. Ten může např. pro potvrzení (confirm) výchozí vlakové stanice a dotaz (request) na cílovou stanici vypadat jako (*confirm, departure_station, "Praha"*) + (*request, arrival_station*) a generování odpovědi tento popis převede např. do textu "Kam chcete jet z Prahy?". Uživatelsky přívětivé jsou také různé styly otázek a jejich průběžné změny, tedy že se agent nebude pouze ptát "Kam chcete jet z Prahy?", ale třeba "Chcete jet z Prahy, kam to bude?", "Chcete jet z Prahy? A kam to dnes bude?". Především v českém jazyce je potřeba dávat velký důraz na správné skloňování, časování, množná čísla a předložky. Takto vygenerovaný text poté vstupuje do bloku převádějící text do řeči, který jej vygeneruje řeč, která je pak prezentována uživateli.

4 Metody a přístupy k reprezentaci stavu

discriminative, generative - statistic atd.

4.1 Scalable multio-domain dialogue state tracking

Reprezentace stavu:

- **candidate set** pro slot - oproti klasickému diskriminativnímu přístupu - sdružená distribuce přes všechny sloty, kde byl problém se scalabilitou pro tasky s hodně daty
  - candidate set - set dvojic values pro určitý slot a jejich skóre odpovídající preferenci uživatele pro danou value
- **samotná reprezentace stavu** - místo všech možných hodnot pro slot s v kroce t omezeno na $\rightarrow V_s'^t = C_s^t \cup \delta_s, \phi_s$ - candidate set, dontcare value, null value
  - konstatní délka K + 2 (délka cand. + 2 dummy values), kde K = 7 je maxim. délka candidate setu, která lze vůbec reálně naplnit

Dialogue state tracking:

- popis modelu - neuronová síť - styl vypočítávání parametrů str. 563/3.1
- feature extraction - používání delexikovaných promluv (což je výstup dial. sys. - klasika) - (využívání deep learning based LU model) - *delex(s)* v kroce t je posláno do GRU network (fig3)
- výhody:
  - utterance, slot a candidate related features
  - lehká reprezentace negate(time), affirm..
  - parameter sharing and transfer learning - parametry neuronky se můžou přenášet do jiných domén $\rightarrow$ není třeba pokaždé přetrénovat neuronku
- nevýhoda
  - potřeba olabelovaných dat

Takeaway message:

- candidate set slot hodnot - omezení DST (po krocích) na tento set
- z toho důvodu možné velké až neomezené value sety
- parameter sharing - ačkoliv je třeba neuronku dotrénovat na určitou doménu
- výstupem (vstupem dial. sys.) je pst každé candidate hodnoty pro každý slot - nastaví se threshold developera

4.2 Discriminative state tracking for spoken dialog systems

Statistical state tracking - trackování posteriorní distribuce přes hidden dialog states (basically HMM). Dva typy:

- discriminative - využití podmíněných modelů (p(Y|X)) trénovaných diskriminativní stylem - můžou trackovat pouze několik hypotéz
- generative - generativní modely (p(X,Y)) k zaznamenání, jak jsou SLU výsledky generovány z hidden dialog states - problém s velkými sety dat

Cíl paperu - možnost mít zvolený počet hypotéz a stále vypočítat správnou distribuci
Reprezetance stavu:

- každý turn je 1 datapoint - vstupem datapointu:
  - set K featur popisující aktuální kontext dialogu - z ASR a SLU
  - G hypotéz ohledně stavu dialogu - mění se v krocích dialogu, jsou mezi sebou disjoint
  - M featur popisující každou hypotézu
- cíl - přiřadit hyptézám pst + pst metahypotéze, pokud nejsou žádné hypotézy správně - na základě nejpst. hypotézy odpovídá dialog uživateli
- analogie - ASR N-best list

Nalezení správné (nejpst.) hypotézy:

- maximum entropy models - popis výpočtu 4-4.1
- features použité v trénování modelu - *general* - turn specific, a *hypothesis-specific* - base (aktuál. turn), history (předešlé turns), confusion (pst erroru - background noise, stejné názvy atd.)

Fixed-length discriminative state tracking

- vybírá se subset $\tilde{G}$ hypotéz na skórování - nejlepší výsledky s $\tilde{G} = 6$ (jedna aktuální, dvě z t-1, tři z t-2) - výpočet v 4.3
- problém s tím, že se sdílí všechny featury přes všechny hypotézy -
  - těžké trénování a rozhodování
  - položky dole v N-best listu nikdy nebudou brány v potaz - špatné odhadování poster. psti
  - se zvyšujícím $\tilde{G}$ kvadraticky narůstá počet feature functions, proto musí být $\tilde{G}$ malé - upper-bound ohraničení pro úspěšnost

Dynamic discriminative state tracking

- využití feature functions, které linkují hypothesis-specific featur s korespondující hypotézou - zbavení se závislosti množství vah na učení na počtu hypotéz
- konkatenace M hyp-spec featur a K general featur (stejné pro všechny hypotézy) + rest hypotéza, kde M má nedefinované hodnoty
- x_i^g je i-tá featura hypotézy g, i = 1,...,M+K, g = 1,...,G+1
- featury jsou dynamicky měněné v každém kroku
- set M+K vah je λ_i, které se maximum entropy model učí
- možný problém - jelikož se general features duplikují, změna λ_i by měnila stejně všechny skóre → kombinace general a hyp-spec features
  - DiscDyn1 - bez změny - M + K vah
  - DiscDyn2 - přidání encodingu ordinálních hodnot hyp-spec featur - vektor indikátorů (boolean hodnot), kde je indikátor nenulový pokud je rank = 1 atd → detailněji popsání ordinal-valued hyp-spec featur, ale pořád ignoruje general featurs → 2(M+K) vah na naučení
  - DiscDyn3 - indikátory i pro general featury → cca 10(M+K) vah na naučení
  - DiscInd - 2 binární klasifikátory - jeden vypočítá skóruje hypotézu podle M a K, druhý dává pst podle K, že meta-hypotéza je správná

Takeaway message:

- z výsledků paperu - diskriminativní lepší než generativní
- DisDyn modely jsou lepší než fixed length - lepší modelovat dynamicky každou hypotézu stavu, než jen subset
- malý rozdíl mezi 1-3, lze tedy použít jednodušší encoding

4.3 PyDial: A multi-domain statistical dialogue system toolkit

Toolkit možný použít k discriminative state tracking? - je v pythonu, je open-source: https://github.com/vmishra04/Pydial

4.4 Mycroft

vytváření SKILLů? rozběhnout a zkusit SKILLy

4.5 A finite-state turn-taking model for spoken dialog systems

Turn-taking - účastníci konverzace mění mluvení a ticho.
Historie:

- čtyřstavový Markovský model - A mluví, B mluví, oba mluví, oba jsou zticha → příliš jednoduchý, není možno rozeznat switching pauzy z A do B nebo B do A
- stavy musí deterministické

Rozšířeno na:

- šestistavový model - USER, SYSTEM (jeden z nich přebírá iniciativu), $FREE_S$, $FREE_U$ (nikdo nepřebírá iniciativu - index ukazuje, po koho iniciativě došlo k uvolnění), $BOTH_S$, $BOTH_U$ (mluví oba - index ukazuje, po koho iniciativě došlo k souběžnému mluvení)
- ne všechny přechody jsou validní
- čtyři možné akce - Grab the floor, Release the floor, Wait while not claiming the floor, Keep the floor
- (R,W) - systém první, uživatel druhý
- typické turn-taking akce:
  - turn transitions with gap - z jednoho účastníka na druhého
  - turn transitions with overlap - účastník přebírá floor, i když ho má druhý účastník
  - failed interruptions - účastník skočí do řeči, ale stáhne se před tím, než stihne druhý účastník floor opustit
  - time outs - stejné jako turn transitions with gap, ale druhý účastník nepřebere floor
- cost matrix:
  - cena, která vyřeší gap nebo overlap je 0
  - cena, která vytvoří nechtěnou gap nebo overlap je konstanta
  - cena, která prodlouží gap nebo overlap je konstanta nebo funkce závislá na čase stráveném v gapu/overlapu
- potřeba vypočítat $C(A) = \sum_{S \in \Sigma} P(s = S|O) \cdot C(A, S)$, kde C(A) je očekáváná cena akce A, Σ je set stavů, O jsou pozorovatelné featury světa???, C(A,S) je cena z matice pro stav S a akci A. $P(s = S|O)$ je pravděpodobnost, že uživatel drží floor, a ta je třeba odhadnout - TODO - dopsat výpočet

Takeaway message - výhody:

- zrychlení a větší jistota průběhu konverzace
- vypočítané pomocí víceméně jednoduchých pravděpodobnostních pravidel
- potřeba pouze odhadnout P(F|O) a $P(d \geq \tau|O, U)$

Takeaway message - nevýhody:

- potřeba trénovací set pauz