

РЕФЕРАТ

Пояснювальна записка до дипломного проекту « Розробка системи віртуального голосового помічника для персональних комп'ютерів »

Голосові помічники, управління комп'ютером, фреймворки, тести, штучний інтелект.

Предметом дослідження був голосовий помічник та методика його перевірки навички та здібності.

Програмний модуль голосового помічника для управління комп'ютером

Завдяки навичкам голосових помічників метою дисертації є підвищення швидкості та якості комп'ютерне керування .

Метод дослідження полягає в розробці модулі голосового помічника та системи автоматизації управління комп'ютером.

Голосовий помічник *Алекса* , помічник *Google Аліса* використовується як приклад для проведення поглибленого аналізу процесу розробки голосового помічника . В результаті ручне тестування можливостей розробки займає багато часу і може містити багато інших помилок.

Результати можуть бути використані для розвитку навичок, що дозволяють голосовим помічникам керувати комп'ютером у *середовищі Python* для автоматизації роботи на комп'ютері, що скоротить час виконання завдання та покращить його якість.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

МН - машинне навчання

DCT - довготривала короткочасна пам'ять

OPM - обробка природної мови

CI - командний інтерфейс

ООП - об'єктно-орієнтоване програмування

UAT - приймальне тестування користувача

OAT - експлуатаційні приймальні випробування

ШІ - штучний інтелект

СУХИ - не повторюйся

ISP - принцип сегрегації інтерфейсу

HTTPS – безпечний протокол передачі гіпертексту

JSON – нотація об'єктів JavaScript

URL – уніфікований локатор ресурсів

DLL - бібліотека динамічних посилань

API - публічний інтерфейс програми

SSL - рівень безпечних сокетів

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ МОЖЛИВОСТІ ГОЛОСОВОГО ПОМІЧНИКА	10
1.1. Загальні відомості про голосового асистента на прикладі програм ...	10
1.2. UX-патерни в голосі.....	Error! Bookmark not defined.
1.3. Проектування інтерфейсу в сім етапів	Error! Bookmark not defined.
1.4. Висновки.....	Error! Bookmark not defined.
РОЗДІЛ 2 ТЕХНІЧНІ ХАРАКТЕРИСТИКИ	Error! Bookmark not defined.
2.1. Постановка задачі.....	18
2.1.1. Характеристика об'єкта автоматизації... Error! Bookmark not defined.	Error! Bookmark not defined.
2.1.2. Вимоги до програми	22
2.2. Отримання даних з звукових сигналів	23
2.3. Вимоги до програми	24
2.4. Висновки.....	25
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЮ	26
3.1. Вибір програмних засобів та операційної системи	26
3.2. Мова програмування	26
3.3. Інтегроване середовище розробки.....	29
3.4. Розробка програми	30
3.5. Етапи налагодження..... Error! Bookmark not defined.	Error! Bookmark not defined.
3.5. Висновки.....	31
ВИСНОВКИ	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33
ДОДАТОК А	

Комісія ПОД та ПЗ				ВСП КРФК НАУ 22 12 25 000 ПЗ			
<i>Виконав</i>	<i>Ринденко В.А.</i>			Зміст	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Дроздова А.М.</i>					3	40
<i>Консульт.</i>	<i>Рубан Т.М.</i>				3-087 121		
<i>Консульт.</i>							
<i>Голова ЦК</i>	<i>Кравчатий А.В.</i>						

ВСТУП

Для більшості з нас було б чудово мати помічника, який завжди вислухає вас, передбачить усі ваші потреби та вживе необхідних заходів для їх задоволення. Все це тепер доступно завдяки помічникам зі штучним інтелектом, а саме голосовим помічникам.

Кожен п'ятий житель Сполучених Штатів має розумний динамік, а це 47 000 000 людей. Помічник може створювати нагадування, список справ, будильник, таймер, читати новини, вмикати музику, подкаст, замовляти доставку, купувати квитки в кіно та викликати таксі. Все це «навички» чи «навички» помічників. Їх ще називають голосовими програмами. У 2018 році для Alexa і Google Assistant було розроблено 70 000 таких програм.

У 2017 році Starbucks запустила функцію замовлення кави додому для Amazon Alexa. Крім того, всі можливі ЗМІ писали про зростання замовлень на доставку, створюючи чудовий піар. Наприклад, Starbucks пішла в Uber, Domino's, MacDonald's, а пральний порошок Tide розвинув власну майстерність для Alexa.

Як і у Starbucks, голосова програма виконує одну-дві функції: замовлення кави, встановлення будильника або виклик кур'єра. Вам не потрібно бути міжконтинентальною корпорацією, щоб щось спроектувати. Ідея, дизайн, тестування, розробка та випуск схожі на подібні етапи у світі мобільної розробки, але з голосовими функціями. Павло Хвай детально розповів про процес: від ідеї до публікації з прикладами реальних ігор, з історичними вставками та аналізом світу розвитку голосу.

У розмовній програмі взаємодії з користувачем будується через розмову: усний - за допомогою розумної колонки або через мікрофон, наприклад, за допомогою Google Assistant. Крім динаміка, пристроєм взаємодії може бути екран, тому розмовні програми також є графічними.

Правильно сказати розмовний додаток, а не голосовий, але це усталений термін, і я його також буду вживати.

Голосові програми мають важливу перевагу перед мобільними пристроями: їх не потрібно завантажувати і встановлювати. Досить знати ім'я, і помічник сам все почне.

Це тому, що немає нічого для завантаження - включаючи розпізнавання мовлення та бізнес-логіку - і вся програма знаходиться в хмарі. Це величезна перевага перед мобільними додатками. Але це не дозволяє працювати з ними в автономному режимі.

Історія голосових помічників почалася з інтерактивних голосових відповідей, інтерактивної системи для запису голосових відповідей. Можливо, ніхто про це не чув. Цей термін, але кожен стикається з ним, коли викликає технічну підтримку і чує про роботу: «Натисніть 1, щоб увійти в головне меню». Хоча вони вже є частиною історії, вони можуть навчити нас чомусь.

Більшість людей намагаються зв'язатися з оператором при взаємодії з системою IVR. Це пояснюється тим, що UX смоче при взаємодії на основі жорстких команд, що незручно.

Це підводить нас до основного правила хорошої розмовної програми.

Хороша розмовна програма взаємодіє з користувачами не через суворі команди, а через живу, природну розмову, схожу на спілкування між людьми.

Спілкування з додатком має бути схоже на виклик піцерії для замовлення, ніж спілкування з командами чат-ботів. Досягти такої гнучкості, як у розмовах між людьми, не вдасться, але розмовляти з додатком комфортно і природною мовою цілком можливо.

Це також перевага голосу перед графічними програмами: вам Не потрібно вчитися користуватися. Моя бабуся не знала, як зайти на сайт або замовити піцу, тому що програму, але зможе зателефонувати доставку через графу. Нам потрібно скористатися цим і адаптуватися до того, що говорять люди, замість того, щоб навчити їх розмовляти з нашим додатком.

Світ голосу обертається навколо віртуальних помічників: Google Assistant, Amazon Alexa та Alice.

Все влаштовано майже так, як ніби це було в мобільному світі, просто замінюючи платформи iOS і Android, тут Аліса, Google Assistant і Alexa, а не графічні програми. - голосові, зі своїми іменами або іменами, і кожен помічник має свій внутрішній сховище голосових додатків . Знову ж таки, говорити «додаток» неправильно, оскільки кожна платформа має свій термін: Аліса має «навички», Алекс має «навички», а Google має «дії».

Щоб розпочати навик, я прошу помічника: «Алекс, скажи Starbucks, що я хочу кави!» Алекс знайде програму для кав'ярні у своєму магазині та поговорить з ним. Тоді не між Олексою і користувачем, а між користувачем і програмою. Багато людей розгублені і думають, що помічник продовжує розмовляти з ними, хоча програма має інший голос.

Для користувача в програмі немає графічної частини - все виглядає як набір діалогових вікон. З боку може здатися, що програма проста річ, її легко створити, але це не так.

Розвиток ділиться на дві частини: розвиток розуміння мови та логічне письмо. Перші два етапи є специфічними, оскільки програми розмовні, а останні два стандартні.

РОЗДІЛ 1

АНАЛІЗ МОЖЛИВОСТЕЙ ГОЛОСОВИХ АСИСТЕНТІВ

1.1. Загальні відомості про голосового асистента на прикладі програми Amazon Alexa

Дизайнери UX і UI скептично ставляться до голосових інтерфейсів. Здається, що це маркетинговий ажітаж, який швидко зникає. Інші не використовують голосові помічники, тому визначають, що звук незручний і неприродний. Але, незважаючи на їх скептицизм, існує самостійна сфера знань – їх секрети, закономірності і механізми (і навіть ринок праці).

Але почнемо з трохи ретроспективи. Двірі у 1995 році. Microsoft випускає Windows 95, і світ різко змінюється. Я пам'ятаю своїх батьків та інших дорослі вони розмовляли про переваги і жаль комп'ютер. І в вихідні ми вся родина пішов відвідати до друзів щоб грати в "Косинці" тому «Сапер».

Nokia 3310 був випущений в 2000 році. Світ знову змінився, цього разу з телефонами з нажимна Кнопка інтерфейс. Пізніше з'явився смартфони з стилуси. У мене теж був такий. Було дуже круто тикати стилусом в екран. Ви відразу виділяєтеся на тлі людей з кнопковими телефонами. Але «найкращий стилус — це ваш власний палець», я говорив Стів Робота. IN 2007 рік рік яблуко помер продай iPhone - і від тих людей чіпають Тара і Swire, і тикання стилусом вже давно швидко.

Комісія ПОД та ПЗ				ВСП КРФК НАУ 22 12 25 001 ПЗ			
Виконав	Ринденко В.А.			Характеристика та аналіз предметної області	Літ.	Арк.	Аркуші
Керівник	Дроздова А.М.					7	40
Консульт.	Рубан Т.М.				3-087 121		
Консульт.							
Голова ЦК	Кравчатий А.В.						

А потім кинувся: 2011 - Apple представляє Siri , у 2014 випускає Amazon Alexa я Amazon Echo , в 2016 рік Виявляється Google помічник , в 2017 рік -

«Аліса» від «Яндекса» ... Для розробників і бізнесу відкрили Як і раніше, в магазині мобільних додатків двері в екосистему помічників. Незабаром тільки лінь (і недалекоглядність) Ні. зроблю навички для голос помічники.

Неможливо уявити досвід користувача (UX, досвід користувача) перебувала в стадії будівництва в вакуум. Вокал звичка народжується в конкретна ситуація, там, де це працює, я органічно - немає сенсу створювати мовний скрипт, а потім розглядати, де б його застосовувати. значний визначити цю ситуацію і вміло її подолати.

Voice зручніше, ніж мобільні веб-додатки або, коли нам потрібна конкретна функція для вирішення конкретного завдання. Чому? Тому що це не потрібно чекати завантажити сайт, прокручувати сторінки, Пошук на меню, натисніть кнопку . Сайти та програми багатофункціональні. Голосова майстерність слід бути ув'язнений під сингл випадку під «тут я зараз».

У лютому 2019 американський ТНТ транслював матчі НБА розриви між ігри «Лос-Анджелес Лейкерс » я «Бостон Селтікс » коментатор Ерні Джонсон я буду оголошувати що почалося продаж обмежений кросівки Nike і що глядачі може запитай Google помічник резерв для їх Тато за 350 доларів.

За шість хвилин кросівки зник. Більше 15 тисячі Люди бронювали їх за допомогою голосового помічника (кількість замовлень перевищила кількість наявних товарів). Реальна ситуація з продажу баскетбольного взуття була реальною гра.

Більше в 2017 рік рік Starbucks придумав як за допомогти голосовий помічник для скорочення часу очікування замовлення і

шляху клієнта до потрібної чашки кави. Можливо, можливо сказати «Олекса, замовлення ми Starbucks » («Олекса, наказів мій « Starbucks

») я брати улюблений пити в найближчий установ вже за кілька хвилин. Ні необхідно стояти в черги я чекай, Коли бариста вигукує ваш ім'я. Вам не потрібно входити в програму, шукайте своє звичайне замовлення, підтвердьте вибір. Потім це г випадок бити Форд: Alexa вбудований в Мультимедійна система SYNC3 , завдяки чому водії мають можливість замовляти та оплачувати голосом ваш Starbucks прямий в салоні автомобіль, Ні відволікаючи від дороги.

Але колеги Алана ШІ розповіли мені, як знайти цей додаток. Історії користувачів призначені вже не для маркетингу, а для завдань ертеgrise. У США технікам, які супроводжують ліфти, необхідно прочитати багато документів, заповнити деякі форми, ввести дані про технічне обслуговування, відзвітувати про виконання завдань — все це робиться в різних системах бухгалтерського обліку. Писання займає багато часу, коли ліфт працював мало. Alan AI використовує голосову технологію штучного інтелекту, щоб дозволити користувачам ліфтера заповнювати форми голос протягом робота або під час подорожі на об'єкті.

Voice UX - це не картинка, а фон до ситуації. Завдання дизайнера - детально вивчити контекст і зрозуміти, чого хоче користувач в тій чи іншій ситуації.

1.2. УН-візерунки в голос

Отже, одна навичка є функцією. Якщо водій замовляв каву, він вирішив цю проблему. Після кави хочете дізнатися, як далеко від Місяця ви знаходитесь? Ще одна вокальна техніка - це те, що викликає це. Зробіть швейцарський ніж в одній навичці – поганий трюк.

Дизайнери Scroll.UX і UI сперечаються з маркетологами про те, скільки інформації відображати на першому екрані. Немає нічого гіршого, ніж чекати, поки вміст завантажуватиметься та нескінченно прокручується. Діалогові інтерфейси також мають перший екран і, на відміну від Інтернету, немає прокрутки. Вам це не потрібно, адже голосовий помічник – це набір навичок. Користувач керував активним навиком і запуслав одну синю функцію.

Модальні вікна та кнопки. Друга назва модального вікна - Dialog. Що таке модальне вікно? Переконайтеся, що або відмовляється виконувати операцію. У реальному житті люди висловлюють свої наміри, так чи ні, без необхідності кнопки.

Мультимодальний. «Розумні» колонки та екрани, смартфони, Іграшки для розмовних дітей, «розумні» будиночки, приладові панелі в машині - пристрої визначають середовище, в якому вони використовуються. Завдяки розумним динамікам те, що користувачі можуть комфортно робити вдома, буде відрізнятися від використання голосового помічника в автомобілі. Один і той же помічник на різних платформах має різні набори навичок.

Визначає контекст використання та пристрій. Перевірте, чи можна об'єднати голос з іншими поданнями на цьому пристрої.

Кросплатформний. Ще один аспект - як зробити навички для декількох помічників. Механізми і логіка можуть бути збережені, але диявол буде прихований в деталях реалізації і шляху, освоєному користувачем. Давайте подивимося, як виглядають навички Аліси та односторонні операції Google Assistant на платформі.

«Аліса» на iOS. Користувачі Аліси повинні спочатку завантажити програму Яндекс.Браузер, дозволити їй використовувати локацію та мікрофон, натиснути на значок помічника і вимовити фразу активації навику. Наприклад, «Виконати навик «Так, сер». Другий і подальший

користувальницький досвід складається з трьох кроків: відкрийте «Яндекс.Браузер», натисніть кнопку виклику помічника, промовте фразу активації.

Google Асистент в iOS. Користувачі завантажують Google Assistant, входять за допомогою облікового запису, дозволяють відправляти повідомлення і дають згоду на доступ до мікрофона. Якщо ви вибираєте англійську за замовчуванням на своєму телефоні, вам потрібно вибрати російську в інтерфейсі програми, попросити помічника змінити мову в голосі і вимовити активну фразу для виклику дії. Тут по-іншому - Поговоріть з додатком Так,. Потім шлях скорочується - вдруге користувач відкриє програму Google Assistant і видасть команду.

Якщо ви не знаєте фразу запуску (а команди, які викликають одну і ту ж навичку, будуть відрізнятися в різних екосистемах), навичка не почнеться - з точки зору UX, це одна зі слабких сторін голосових помічників. Але Google, Яндекс та інші компанії зараз працюють над виявленням навичок зручності (як правильно доносити інформацію про нові навички) до споживачів, полегшуючи пошук корисних і актуальних навичок пізніше.

І, до речі, дизайнери VUI наполягають, що навички з однаковою механікою для різних помічників – це все ж окремі проекти.

1.3. проектування інтерфейс в сім етапи

Я запитав VUI дизайнерів, колег з Just AI та користувачів конструктора Аймілогічні навички, з яких зазвичай починають роботу над сценарієм. Всі кажуть «за ідеєю». Ви можете вільно описати її в текстовому редакторі.

Павло Хвай, засновник логотипу інструменту для розробки голосовий додаток: Найшвидший і найдешевший спосіб навчитися будувати розмову між користувачем і програмою - це написати парадигму розмови. Це текстовий файл, який описує роботу будь-якого потоку. За форматом приклад діалогу схожий на сценарій фільму, в якому всі репродукції малюються відповідно до персонажів.

Розкажіть, що робить бот. Використовуйте ємні фрази середньої довжини. Завершіть речення закритим запитанням: щоб користувач зрозумів, чого він хоче.

bad- "Привіт! Я Активний. Я не можу прожити і дня без спорту. Він - моє все, моє життя, моє натхнення! А в мене великий досвід тренера і сотні вправ на базі! З Я з радістю поділюся з вами своїми улюбленими вправами!"

good- "Привіт! Я Активний, бот-спортсмен. З Я із задоволенням порекомендую тобі комплекс вправ. Хочеш, я розповім тобі про йогу?"

Опрацюйте шляхи користувачів

У мовах програмістів намір є наміром. Моделювання з класичним інтерфейсом є формальним неформальним запитом або. Формалізована робота простіше, а неформальна робота може перетворити життя в кошмар.

Розробники використовують блок-схему під час роботи з користувацькими шляхами. Завдання завдання - допомогти описати логіку програми. Блок-схема складається з кроків діалогу від імені користувача і системи, іноді в неї додаються логічні елементи - виклики API, робота з контекстом.

Блок-схема в Міго

Часто блок-схема описує основні напрямки голосової майстерності. Деякі дизайнери описують кожну деталь навіку за допомогою блок-

схеми. Павлу Хваю це робити не рекомендую, бо скоро перестане читатися, а будь-які поправки затягнуться.

Найпростіший спосіб пропрацювати шлях користувача — пройти з ним від початку до кінця і подивитися, де з'являться умови та відхилення від основного сценарію.

створити навігацію

Незалежно від шостого етапу, користувач завжди повинен мати можливість почати все спочатку, повернутися назад, рухатися вперед і реагувати по-різному. Не змушуйте його запам'ятовувати команди.

Приклад з життя. Вже згаданий навик «Так, сер» (у нього в «Алісі» аудиторія 650 000 чоловік, крім того, вона стала однією з найпопулярніших ігор Google Assistant не тільки в Почаї, але і в світі) - це атмосферна гра з музикою, яку варто синтезувати мовою, цікавим сюжетом, ігровою механікою на початку використання, тільки за допомогою Yes, No and Enough для управління грою. За сюжетом, гравець - майстер, поміщик і фермер, входить в «Ваша Величність! Велика кількість покладів міді була знайдена на нашій родючій землі! Хотите почати будувати шахту? Ви можете відповісти «так», але хочеться звикнути до персонажа і говорити як велика людина.

- Ваша милість! На наших родючих землях виявлено великі поклади міді! Хотите почати будувати шахту? - звільнений!

VUI дизайнер виявлено це момент і вчив звичка краще розуміти користувачів, надаючи їм більше свободи у відповіді.

Напишіть приклади діалогів і зробіть відповіді бота більш різноманітними

Мої колеги з Just AI записують приклади діалогів у таблиці. Їм так зручно, але це не найзручніший варіант - опишіть логіку і можливі перетворення. Хтось пише сценарій у програмі Word. Не існує єдиного формату та правила опису.

Користувачі дуже дратуються, коли помічник починає повторювати. Гіпотезу підтверджує Неллі Камаєва, дизайнер Alan AI. Перевіряючи навички для дітей, вона побачила, як швидко хлопці втратили інтерес, наштовхнувшись на ту саму відповідь.

Необхідно надати кілька синонімічних зауважень, які користувач почує, перейшовши на той самий крок сценарію. Дизайнери VUI рекомендують використовувати від трьох до десяти варіантів однієї фрази.

Подивіться на смітник, щоб знайти нерозпізнані наміри

Сміття, або всеосяжний на мові програмістів, — це місце, де користувацькі фрази потрапляють з різних причин. - З ким я розмовляю? Ви робот? — Ти мене вкусив. Бажаєте продовжити спілкування? Фраза «З ким я взагалі розмовляю? Ти робот?! "буде перехоплено (якщо ви не надали сценарій)". Ти мене вкусив. Бажаєте продовжити спілкування? "- стандартна підказка в таких ситуаціях Подумайте заздалегідь, ніж допомагати користувачеві, який потерпів невдачу в усвідомленні всього.

Відгадайте співрозмовників, які вам набридли. Мене це не хвилює, я не хочу продовжувати говорити, ми називаємо їх нудними. Те ж саме стосується і комунікативних навичок. Аліса, Алекса та Google Assistant мають власні голосові набори, які містять різноманітні чоловічі та жіночі голоси, акценти та звукові ефекти, щоб оживити мову робота, таку як кашель та нюхання.

Якщо ви хочете справити враження на користувача, залучіть його до розмови, подумайте про стиль розмови, зробіть синтез: розмістіть акценти, паузи і інтонацію. Це була важка робота, але повірте, вона того варта. У внутрішній воркшопі колега створив навик для Алекси — слухайте цей уривок:

- Гей, хлопець. Жахлива погода, так? Хочеш напитися? Нехай ваші старі друзі Джо змішають вас і хороший ірландський напій. В порядку? —

А тепер, коли тобі захочеться клацнути та йди в паб Джо. Там ти ніколи не будеш пити сам. Я чекаю на вас.

Інший спосіб роботи з голосом — запис аудіо з професійними акторами. Довго, дорого, ви втрачаєте гнучкість, якщо хочете щось додати до сценарію, але ефективно. Уміння може говорити голосами знаменитостей, політиків та кіногероїв (ну, ви вже могли почути наш голос за кадром «Світу Лавкрафта», але ось як Google Assistant говорить голосом володаря Греммі Джона Ледженда).

Для одного з бізнес-кейсів ми також використовували голосові записи від професійних акторів. Лише 0,5% клієнтів змогли запідозрити (або навіть здогадатися, але скоріше підозрювати), що бот розмовляє з ними.

Ви можете підбирати звуки з готових бібліотек або, щоб створити свої власні (Алеха вимагає розширення файлів, тому вам доведеться поводитися з перетворенням). У тому ж рпі «Так, сер» для «Аліси» використовуються звуки для створення атмосфери: іржання коня, гомін натовпу, зловісний сміх, волинка. А нещодавно Аліса отримала навик Росії Birds, де можна почути спів і щебетання жайворонків, горобців та інших птахів у розмові.

Дизайн голосового інтерфейсу не обмежується блок-схемами і сухим текстом. У дизайнерів є все, щоб створити навички для залучення користувачів до процесу.

І інші поради від дизайнерів VUI

Не вивчайте інтерфейс. Мова – це знайомий і зрозумілий інтерфейс. Не потрібно вчити людину говорити. Він може це зробити.

погано- Щоб знову прослухати повідомлення, скажіть «Послухай ще раз». Щоб перейти до наступного повідомлення, двічі натисніть наступне повідомлення.

Гаразд - "Прослухайте повідомлення ще раз і перейдіть до наступного?" Давайте закриємо тарілку. Я рекомендую уникати відкритих питань і відкритих заяв ботів, де користувачі повинні бути спрямовані в дію.

bad- "Привіт! Я Symphony, любитель музики-ботів. Буду радий порекомендувати вам альбом і розповім про нього."

good- "Привіт! Я Symphony, любитель музики-ботів. З Я з радістю порекомендую вам альбом і розповім про нього. Хочете дізнатися про найкращий трек дня?"

Уникайте бюрократії. Очевидна рекомендація, якої мало хто дотримується. Ніхто з нас не хоче читати складний і перевантажений текст, а тим більше слухати його.

Погано - «Важливо пам'ятати, що наступний альбом артиста буде платиновим, що приводить нас до висновку, що цей дебютний альбом є засобом виходу на міжнародну сцену».

добре - дебютний альбом привернув увагу всього світу. Не дивно, що його наступні альбоми двічі вигравали платинову!

У тиші, на вулиці, в галасливому приміщенні, говоріть в різних тонах і на різних швидкостях перевіряйте свої навички. Навіть у найспокійніших місцях можуть виникнути проблеми. Деякі люди можуть вважати процес тестування нудним, але я запевняю вас, що це не так.

Навички вимагають реального краш-тестування!

З особистого досвіду. На одному з воркшопів я розробив фітнес-навик: за планом Алекси я спочатку проводив інструктаж, потім вмикав музику, і людина повторювала вправу. Я пристрасно все випробував на собі: стрибати і бігати, щоб обрізати музику, змінювати довжину треків, багаторазово повторювати вправи і, нарешті, навички UX мене задовольнили.

Користувачі є провокаторами. Вони вивчать реакції навичок, не пов'язаних з темою: вміння замовляти піцу, а користувач – як би дуже хитрий – розпитувати про землю. Придумайте відповідь. У Just AI з килимком контрольного списку, який використовується під час тестування.

Кажі все, що придумаєш. Слухайте все, що почує користувач. Попросіть колег прочитати і навіть розіграти сценарій. Запишіть мову, поверніться до записів, експериментуйте.

Але з першого разу не все передбачиш у майстерності. Прийміть це і прийміть це. Ваша майстерність потребує додаткового навчання. Читайте діалоги та аналізуйте журнали. Як це зробити за допомогою Python, в одному з випусків Alice School розповідає Дарія Сердюк, інженер-дослідник НЛП Just AI.

1.4. Висновки в розділі

Здійснено голосові принципи побудови голосового помічника. Описано на основі інформації про голосових помічників.

Інструменти, доступні для автоматизованого тестування навичок голосового помічника, були проаналізовані та виявлені у вузькій області розробки програмного забезпечення.

В результаті аналізу було прийнято рішення про необхідність розробки власної методики перевірки навичок голосового помічника, а також розробки платформи для автоматизації перевірки навичок відповідно до розробленої методики.

РОЗДІЛ 2

ТЕХНІЧНІ ХАРАКТЕРИСТИКИ

2.1. Постановка задачі

2.1.1 Характеристика об'єкта автоматизації

Об'єктом автоматизації є виконання цих голосових команд.

На даний момент багато хто користується голосовими помічниками, тому що це досить швидкий варіант - використовувати голос для пошуку інформації або для запуску програм.

Тому виникає необхідність створити голосового помічника, який:

- шукав програми на комп'ютері та завантажував їх;
- шукали в Інтернеті;
- завантажувати потрібні користувачеві сайти;
- мали можливість закривати непотрібні програми

Цей список можна довго розширювати, за замовчуванням у програмі більше 15 можливих операцій, і їх можна додати, якщо знадобиться щось своє.

Сам процес перегляне доступні команди запиту для цієї програми, які зможуть відповісти і виконати цей запит. Голосові помічники зможуть відповідати індивідуальним вимогам користувачів на практиці, так як код дуже гнучкий і вимагає мінімальних знань програмування для редагування або додавання нових функцій. Завдяки підходу до розкладання розширення системи не є складним завданням.

Комісія ПОД та ПЗ				ВСП КРФК НАУ 22 12 25 002 ПЗ			
Виконав	Ринденко В.А.			Проектування інформаційної системи	Літ.	Арк.	Аркуші
Керівник	Дроздова А.М.					18	40
Консульт.	Рубан Т.М.				3-087 121		
Консульт.							
Голова ЦК	Кравчатий А.В.						

Великі компанії, які пропонують своїх голосових помічників (Google, Apple та інші), зі списком з десятків тисяч команд, які постійно оновлюються, не можуть опублікувати повний список усіх команд, які є в програмі.

Тому була введена програма Help. При використанні програми слід пам'ятати, що доступність відповідної команди для даного запиту гарантована, а не конкретні побажання користувача.

Ви можете скористатися кнопкою Довідка безпосередньо та через папку, де розташована програма, щоб знайти потрібну команду. Це також буде реалізовано в цій програмі.

2.1.2 Вимоги до програми

З попереднього опису предметної області статті потрібно було розробити програмний продукт, щоб спростити процес пошуку необхідної інформації і програм на ПК. Крім того, запланований продукт повинен включати такі функції, як збереження команд вводу та виведення завершених команд.

Користувач є клієнтом з доступом до всіх команд, і він вводить голосовий запит для пошуку. У сучасних умовах створення процедури, яка автоматизує її діяльність, є нагальним і необхідним завданням.

Програма передбачає розробку глосарію (далі – глосарій) для збереження ключових слів. Також надає зручний інтерфейс.

Словник повинен містити часто вживані вирази: відкрити, запустити, закрити, шукати тощо.

Інтерфейс повинен бути зручним і комфортним і не створювати проблем для користувача. Програма повинна володіти всіма якостями зручного людино-машинного інтерфейсу, такими як: дружелюбність,

природність, послідовність, не надмірність і повинна бути інтуїтивно зрозумілою.

2.2 Опис математичного методу розв'язування задачі

Взагалі, проектування слід робити методом декомпозиції, тобто поділу однієї великої задачі на трохи простіші, кожна з яких також ділиться на підзадачі. Глибина декомпозиції, тобто глибина розділу, визначається обсягом завдання, що утворюється після розбиття на основні розділи. Вважати атомарною (неподільною) для проектування завдання, яке повністю виконує одну логічно пов'язану дію.

Декомпозиція — це процес розбиття програми на дві частини. Неважко зрозуміти, навіщо потрібно проводити правильне розкладання: адже до вирішення об'ємної і складної задачі залучена велика кількість людей, і ця кількість зростає пропорційно складності задачі. Можливо, регулярні контакти між цими людьми будуть проблемою. Основне завдання слід розділити на частини, які можна розробити окремо і які легко об'єднати в одну систему. Такий поділ дозволить самостійно займатися різними частинами проекту. Так, і головне - значно спростить майбутній процес налагодження.

Ви можете використовувати шаблон схеми класу, щоб створити новий клас за допомогою ООП.

Об'єктно-орієнтоване програмування (ООП) — одна з парадигм програмування, яка розглядає програму як набір «об'єктів», що взаємодіють один з одним. Він використовує кілька технологій з попередніх парадигм, включаючи успадкування, модульність, поліморфізм та інкапсуляцію.

Метою глосарію є лише збереження окремих, не пов'язаних між собою виразів, його структура дуже проста. Однак одна з головних вимог до організації глосарію – вміти здійснювати пошук за значенням. Справжні глосарії часто містять сотні або навіть тисячі виразів, теоретично їх може бути більше мільярда. Наявність такої величезної кількості виразів визначає корисність.

2.3 Отримання даних із звукових сигналів

Розпізнавання мов є однією з багатьох програм штучного інтелекту, які можна використовувати в різноманітних повсякденних проєктах, щоб підвищити їх доступність. Google Assistant і Amazon Alexa очолюють список найпопулярніших способів розпізнавання мови. Але з ними є одна проблема: їхні API для розпізнавання мови доступні лише через Інтернет і неможливо отримати доступ без підключення до Інтернету.

Це все добре, оскільки розпізнавання мови найчастіше використовується в домашній техніці, і в цю «епоху онлайн» (трохи більше, ніж у 2020 році) буде важко знайти дім без підключення до Інтернету. Але якщо ви хочете використовувати офлайнове розпізнавання мовлення? На щастя, у нас є готове рішення, яке не вимагає особливих зусиль з вашого боку!

Vosk — це безкоштовний набір інструментів з відкритим кодом, який використовується для розпізнавання мовлення в автономному режимі. Він підтримує розпізнавання мов 16 мовами, включаючи англійську, індійську, англійську, французьку, іспанську, португальську, китайську, російську, турецьку, в'єтнамську, італійську, голландську, арабську, грецьку, а також нещодавно додану німецьку, каталонську та фарсі.

Оскільки API Vosk підтримує кілька мов, у нас є моделі для кожної з цих мов. Це невеликі моделі (50 МБ кожна), але ви можете знайти і більші

моделі серверів. Моделі мають різноманітні функції, такі як ідентифікація динаміка, API потокової передачі та словник, який можна переналаштувати.

Іншою важливою особливістю інструментарію є його доступність для різних платформ, включаючи настільні сервери з Python, Java, C#, Node, а також легкі пристрої, такі як Android, iOS і Raspberry Pi. Тому ви можете використовувати його різними способами для транскрипції відео, а також для домашньої техніки та чат-ботів.

2. 4 Висновок до розділу 2

Комп'ютер голосового керування на основі глосарію з алгоритмами розпізнавання мовлення — це програма з графічною оболонкою, особливо для людей з обмеженими можливостями, але завдяки зручному інтерфейсу та простоті це зручна програма.

У цьому розділі аналізуються існуючі методи розпізнавання людської мови та обговорюються основні типи завдань, які можуть існувати без систем розпізнавання. Також були визначені необхідні вимоги та опис математичного рішення програми.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЮ ГОЛОСОВОГО АСИСТЕНТА ТА ЙОГО ТЕСТУВАННЯ

3.1. Вибір програмних засобів та операційної системи

Всі органи управління розроблені вручну. Передача даних між Клієнтом і сервісом є складним організаційним процесом.

Але у цього методу є багато істотних переваг:

- легкість поділу на модулі та програмування кожного з них окремо один від одного;
- з наведених вище функцій впливає легкість процесу налагодження;
- дуже велика гнучкість і легкість внесення змін, як простих, так і тривимірних;
- незалежність виконання від встановленої операційної системи;
- Простота проектування програми під особисті потреби користувача.

3.2 Мова програмування

Для виконання завдання була обрана мова програмування Python.

Ідея Python виникла в 1989 році, коли її творець Гвідо ван Россум зіткнувся з недоліками мови ABC (включаючи розширюваність). Rossum вже почав розробляти нову мову, яка об'єднує всі корисні функції мови ABC, а також нові та ідеальні функції, такі як розширюваність та обробка винятків.

Комісія ПОД та ПЗ				ВСП КРФК НАУ 22 12 25 003 ПЗ			
Виконав	Ринденко В.А.			Програмна реалізація	Літ.	Арк.	Аркуші
Керівник	Дроздова А.М					23	40
Консульт.	Рубан Т.М.				3-087 121		
Консульт.	.						
Голова ЦК	Кравчатий А.В.						

Python 1.0 був випущений в 1994 році. Він бере на себе модульну систему від Modula-3, може взаємодіяти з операційною системою Amoeba і включає функціональні інструменти.

Програмування.

У 2000 році основна команда розробників Python переїхала в Beopen.com, а в жовтні 2000 року Python 2.0 випустила ряд імпровізацій, включаючи збір сміття і підтримку Unicode.

Python 3.0 був випущений в грудні 2008 року, відмовившись від зворотної сумісності і нових конструкцій, щоб уникнути дублювання конструкцій і модулів. Це все ще багато парадигмальна мова, яка надає розробникам параметри для об'єктно-орієнтованого, структурованого програмування та функціонального програмування. Сьогодні існує кілька реалізацій Python, в тому числі Jython, написаних Java для віртуальної машини Java; IronPython, написаний на C#, для інфраструктури спільної мови, і піпі-версія, написана в RPython і перекладена на C. Хоча ці реалізації працюють на рідній мові, вони можуть взаємодіяти з іншими мовами за допомогою модулів. Більшість з цих модулів засновані на моделях соціального розвитку і є відкритими і безкоштовними.

Тестовою реалізацією Python є інтерпретатор CPython, який підтримує більшість активних платформ. Він поширюється за безкоштовною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень у будь-якій програмі. Існує реалізація компілятора для JVM з можливістю компіляції, CLR, LLVM та інші незалежні реалізації. У проекті PyPy використовується компіляція JIT, що значно збільшує швидкість роботи програм Python.

Компанії, що займаються програмним забезпеченням, віддають перевагу Python через свою універсальність і менше програмного коду. Майже 14% програмістів використовують їх в операційних системах, таких як UNIX, Linux, Windows і Mac. Програмісти великих компаній

використовують Python, оскільки він зарекомендував себе в розробці програмного забезпечення з наступними унікальними характеристиками.:

- Інтерактивність
- Інтерпретація
- Модульність
- Динаміка
- Об'єктно-орієнтований стиль
- Портативність
- Високий рівень
- Розширення з C і C ++

Ця мова також надає великі стандартні бібліотеки. Найбільш часто використовувані завдання програмування вже написані в цих бібліотеках, що підвищує компактність коду.

Більше того, поріг входу низький, а код значною мірою лаконічний і зрозумілий навіть для тих, хто ніколи ним не користувався. Завдяки простоті коду подальше обслуговування програм, написаних на Python, стає легшим і приємнішим порівняно з Java або C++. З точки зору бізнесу перевагою є зниження витрат і підвищення продуктивності співробітників .

Окрім сотень тисяч індивідуальних розробників та невеликих компаній, Python підтримує таких ІТ-гігантів, як:

- Facebook
- Яндекс
- RedHat
- Google
- Dropbox
- Telegram
- Mozilla
- Amazon

Найголовніше, Google Python забезпечується дуже зручно API Клієнт.

Загалом, ми маємо на увазі Python - це мова програмування загального призначення високого рівня, орієнтована на підвищення продуктивності розробників та коду читання; Python Синтаксис ядра мінімалістичний; Це мова програмування, Призначений для написання високошвидкісних програм; Підтримує структуроване, об'єктно-орієнтоване, функціональне, імперативне та аспектно-орієнтоване програмування; Ключові архітектурні особливості — Динамічний введення тексту, автоматичне управління пам'яттю, повний самоаналіз, механізм обробки винятків, підтримка багатопотокових обчислень, передові структури даних; Підтримка поділу програм на модулі, які можна об'єднати в пакети.

3.3 Інтегроване середовище розвитку

Інтегроване середовище розробки - це програмне забезпечення, яке надає інтерфейс користувача для розробки, тестування та налагодження коду. Інтегроване середовище розробки надає різноманітні інструменти та функції для спрощення розробки та стандартизації мов програмування на основі коду, написаного розробниками.

За допомогою інтегрованого середовища розробки ви можете розробляти додатки або динамічні веб-додатки тощо. Воно включає редактор коду, компілятор або інтерпретатор і налагоджувач для доступу до графічного інтерфейсу користувачів, а також дозволяє користувачам писати та редагувати вихідний код у редакторі коду.

Основна мета використання полягає в тому, щоб дозволити вам швидко і якісно писати програмний код. Як згадувалося вище, середовище включає вбудований компілятор, який перетворює програми в машинний код або байт-код і економить багато часу. Ви також можете вибрати з декількох мов програмування на ваш вибір.

Хоча серед багатьох, але всі вони мають деякі спільні риси:

- Текстовий редактор
- Налаштовувач
- Компілятор або інтерпретатор
- Виконання написаного коду
- Підтримка принаймні однієї мови програмування та всіх її версій
- Різноманітність плагінів

Написання голосового помічника було здійснено в програмуванні JetBrains PyCharm Community Edition 2019. Середовище показано на рис. 3.1.

3.4 Розробка програми

Загалом розробка програми виконувалася в такому порядку:

- Позначення каталогу з проектом, створення додаткових каталогів, створення файлової структури;
- Створення моделі;
- створення алгоритму додавання нових команд до програми та їх виконання
- Створення та тестування двигуна;
- Програмування механізму обробки помилок двигуна;
- створення та програмування механізму роботи із запитами;
- програмування механізму обробки помилок при роботі з Глосарієм;
- Розробка основної форми. Основна форма розроблялася в кілька етапів:
 - Програмування основної форми;
 - Написання власних функцій для взаємодії з кнопками;
 - Розробка поточної форми;
 - Написання власних функцій для взаємодії з глосарієм;
 - Програмування механізму регулярних виразів;
 - Зручність обслуговування та налагодження коду.

Розробка програми завершується процесом тестування розробником і формуванням відповідного інсталяційного файлу для полегшення процесу встановлення програми.

Діаграма класів UML і блок-схема програми знаходяться у відповідних додатках В і Р.

3 . 5 кроків налаштування

Програма була апробована та відкоригована в кінці розробки кожної частини окремо.

Висока модульність програми дозволяє протестувати повністю завершену частину проекту відразу після його програмування.

Щоб перевірити програму, її протестували в реальній ситуації.

Суть налагодження полягає в тому, щоб користувач розробив тест-систему для тестування програми за всіма можливими схемами. Кожен тест має набір входів, які знають результат. Вони намагаються підбирати тести таким чином, щоб не тільки факти помилки були визначені, але і локалізовані, тобто виявлена і звужена частина програми, яка містить помилку.

3 . 6 Висновки в розділі 3

У цьому розділі описані можливі бібліотеки та мови програмування для вирішення проблеми. Виявилось, що цей проект краще виконувати на мові програмування Python 3.x, використовуючи фреймворк Vosk . Після внесення істотних змін у структуру програми необхідно перекомпілювати проект і бажано протестувати змінену програму. Описано етапи розробки як алгоритму програми . Також були розглянуті варіанти можливих програмних помилок та процесу налагодження для пошуку помилок.

ВИСНОВКИ

Дипломний проект містить ґрунтовну інформацію про роботу голосових помічників, їх майбутнє. Обробка природної мови зараз не використовується, за винятком консервативних галузей. Більшість технологічних рішень вже давно визнають розпізнавання та обробку «людських» мов: тому звичайний IVR з жорстко визначеними параметрами відповіді поступово зникає, чати починають спілкуватися більш адекватно без участі живого оператора.

Метою дипломного проекту «Розробка віртуальної системи голосового помічника для персональних комп'ютерів» є створення програми для повсякденного життя.

Програмне забезпечення включає в себе:

- Бібліотека з алгоритмами розпізнавання мовлення;
- Арифметика установки інструкцій;
- Алгоритми виконання команд;

Об'єктом автоматизації є виконання цих голосових команд.

Працюючи над цим продуктом, я вивчив середовище розробки QtDesigner і мову програмування Python. Програмний продукт можна покращити шляхом додавання нових функцій і класів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python[Електронний ресурс]. – Режим доступу:
<https://ru.wikipedia.org/wiki/Python>
2. Swaroop С.Н. «А Byte of Python»[Текст]
3. What is Apple Siri [Електронний ресурс] – Режим доступу:
<https://www.pocketlint.com/apps/news/apple/112346-what-is-siri-apple-s-personal-voice-assistant-explained>
4. What is Microsoft Cortana [Електронний ресурс] – Режим доступу:
<https://searchenterprisedesktop.techtarget.com/definition/Cortana>
5. What is Google Assistant [Електронний ресурс] – Режим доступу: <https://www.pocket-lint.com/apps/news/google/137722-what-is-google-assistant-howdoes-it-work-and-which-devices-offer-it>
6. Python benefits [Електронний ресурс] – Режим доступу:
<https://www.invensis.net/blog/it/benefits-of-python-over-other-programming-languages/>
7. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. сангл. – СПб.: Символ-Плюс, 2011. – 992 с
8. Библиотека для распознавания речи VOSK [Електронний ресурс] – Режим доступу: <https://alphacephei.com/vosk/index.ru>