**Професионална гимназия по електроника и енергетика, гр. Банско**

**България, Банско 2770, ул. "България" №23,**

**тел: 074988402, е-mail: pgeebansko@abv.bg**

|  |
| --- |
|  |

Дипломна работа

Тема: н

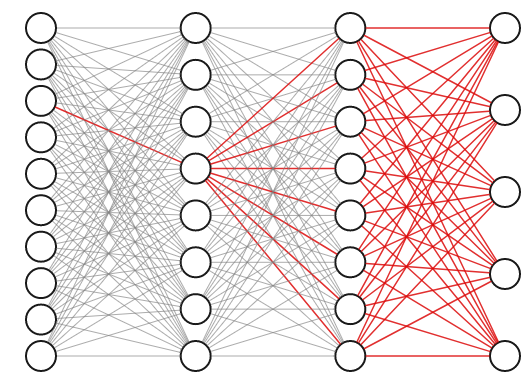
Програма на Python за решаване а квадратно уравнение

Банско 2023

Невронни мрежи

Представете си, че сте планински катерач на върха на планина и е паднала нощ. Трябва да стигнете до базовия си лагер в подножието на планината, но в тъмнината само с малкото си фенерче не можете да видите повече от няколко метра пред себе си. И така, как слизате? Една стратегия е да погледнете във всяка посока, за да видите в коя посока земята се спуска най-много, и след това да пристъпите напред в тази посока. Повторете този процес много пъти и постепенно ще се спускате все по-надолу. Понякога може да заседнете в малка падина или долина, в който случай можете да следвате инерцията си малко по-дълго, за да излезете от нея. Като оставим настрана предупрежденията, тази стратегия в крайна сметка ще ви отведе до дъното на планината. Този сценарий може да изглежда отделен от невронните мрежи, но се оказва добра аналогия за начина, по който са обучени. Всъщност толкова добра, че основната техника за това, градиентно спускане, звучи много като това, което току-що описахме. Спомнете си, че обучението се отнася до определяне на най-добрия набор от тегла за максимизиране на точността на невронната мрежа. Невронните мрежи могат да се използват, без да се знае как точно работи обучението, точно както човек може да работи с фенерче, без да знае как работи електрониката вътре в него. Повечето съвременни библиотеки за машинно обучение са автоматизирали значително процеса на обучение. Поради тези неща и тази тема, която е по-строга от математическа гледна точка, може да се изкушите да я оставите настрана и да се втурнете към приложения на невронни мрежи. Освен това способността за обучение на големи невронни мрежи ни убягваше в продължение на много години и едва наскоро стана осъществима. Невронна мрежа със скрити слоеве са силно взаимозависими, за да видите защо, помислете за подчертаната връзка в първия слой на трислойната мрежа по-долу. Ако променим леко теглото на тази връзка, това ще повлияе не само на неврона, към който се разпространява директно, но и на всички неврони в следващите два слоя, и по този начин ще повлияе на всички резултати.

фиг. 1



И поради сложността за обучение на невронни мрежи ние ще използваме готови обучени невронни мрежи ще разгледаме кои са най-популярните за разпознаване на реч.

**Как работи разпознаването на реч – Общ преглед**

Първият компонент на разпознаването на реч е, разбира се, речта. Речта трябва да се преобразува от физически звук в електрически сигнал с микрофон и след това в цифрови данни с аналогово-цифров преобразувател. Веднъж дигитализирани, няколко модела могат да се използват за транскрибиране на аудиото в текст.

За да се декодира речта в текст, групи от вектори се съпоставят с една или повече фонеми - основна единица на речта. Това изчисление изисква обучение, тъй като звукът на една фонема варира от говорещ до говорещ и дори варира от едно изказване на друго от един и същи говорещ. След това се прилага специален алгоритъм за определяне на най-вероятната дума (или думи), които произвеждат дадената последователност от фонеми. За да се декодира речта в текст, групи от вектори се съпоставят с една или повече фонеми - основна единица на речта. Това изчисление изисква обучение, тъй като звукът на една фонема варира от говорещ до говорещ и дори варира от едно изказване на друго от един и същи говорещ. След това се прилага специален алгоритъм за определяне на най-вероятната дума (или думи), които произвеждат дадената последователност от фонеми.

За щастие, като програмист на Python, не е нужно да се тревожите за нищо от това. Редица услуги за разпознаване на реч са достъпни за използване онлайн чрез API и много от тези услуги предлагат Python SDK .

Избор на библиотеки за разпознаване на реч в Python

В PyPI съществуват няколко библиотеки за разпознаване на реч. Някои от тях включват:

* apiai
* assemblyai
* google-cloud-speech
* pocketsphinx
* SpeechRecognition
* watson-developer-cloud
* wit

Някои от тези библиотеки като wit и apiai – предлагат вградени функции, като обработка на естествен език за идентифициране на намерението на говорещия, които надхвърлят основното разпознаване на реч. Други, като google-cloud-speech, се фокусират единствено върху преобразуването на реч в текст. Има една библиотека, който се откроява по отношение на лекотата на използване която е SpeechRecognition. Разпознаването на реч изисква въвеждане на аудио, а SpeechRecognition прави извличането на този вход наистина лесно. Вместо да се налага да създавате скриптове за достъп до микрофони и обработка на аудио файлове от нулата, SpeechRecognition ще ви накара да стартирате само за няколко минути. Библиотеката SpeechRecognition действа като обвивка за няколко популярни API за реч и по този начин е изключително гъвкава. Един от тях – Google Web Speech API – поддържа API ключ по подразбиране, който е твърдо кодиран в библиотеката SpeechRecognition. Това означава, че можете да станете без да се налага да се регистрирате за услуга. Гъвкавостта и лекотата на използване на пакета SpeechRecognition го правят отличен избор за всеки проект на Python. Въпреки това поддръжката за всяка функция на всеки API, който обвива. И така, сега, след като сте убедени, че трябва да изпробвате SpeechRecognition, следващата стъпка е да го инсталирате във вашата среда.

**Разпознаване на реч**

В днешната ера на бързо напредващи технологии, разпознаването на реч се откроява като нещо обичайно във всеки аспект от живота ни. Разпознаването на реч е лингвистика в компютрите, която позволява на системите да разпознават и превеждат човешка реч в текст. Включва областите на електротехниката, компютърните науки и лингвистиката. Нивото на автоматизация и логика обаче, което виждаме в днешните електронни устройства не би било възможно без концепцията за изкуствен интелект. Използват се гласови асистенти техники за разпознаване на глас, синтез на реч и обработка на естествен език (NLP). Картографирайте гласовите входове към командните изходи. Тези техники са подполе на изкуствените интелект, който позволява на компютърните системи да имат интелект, близък до човешкия (в този случай, по отношение на разбирането на нашата реч). Разпознаването на реч включва няколко стъпки, които включват анализ на звука вълнова форма, разделяне на изказвания чрез мълчание, разпознаване на думите във всяко изказване и произвеждайки крайния резултат под формата на текст (оттук и терминът реч към текст (STT)). За всяко изказване характеристиките се извличат като вектор на характеристиките. Аудио моделът се нарича Hidden Моделът на Марков (HMM) описва всеки последователен процес като речта. В речта разпознаване, акустичният модел, езиковият модел и фонетичният речник са трите използвани неща. В днешния свят разпознаването на реч и гласовите асистенти са навсякъде. от персонализирани гласови записи в кол център, които водят обаждащите се през опциите до бизнеса използвайки имейли и документи, генерирани от говор, тази концепция нарасна много в миналото десетилетие. Гласовите асистенти могат да предоставят голямо разнообразие от услуги и по-специално тези от Amazon Alexa и Google Assistant растат с всеки изминал ден. Те включват предоставяне на информация извлечени от интернет, възпроизвеждане на музика или видеоклипове и заместване на хора в областта на обслужване на клиенти и разговорна търговия. Те също са внедрени в интелигентен гласов асистент в автомобилният сектор за управление на функции и настройки от вътрешността на автомобила чрез гласови команди. Технологичните компании работят за създаването на все по-сложни технология, която ще автоматизира повечето процеси, които извършваме за един денЗа повечето от нас върховният лукс би бил асистент, който винаги ни изслушва вашето обаждане, предвижда всяка ваша нужда и предприема действия, когато е необходимо. Този лукс е сега достъпно благодарение на асистентите с изкуствен интелект, известни още като гласови асистенти. Те влизат донякъде малки пакети и може да извършва различни действия, след като чуе дума за събуждане или команда. Те могат да включват светлини, да отговарят на въпроси, да пускат музика, да правят онлайн поръчки, и т.н. Те също така откриха нарастваща съвместимост с IoT (Интернет на нещата), свързан устройства. Гласовите асистенти не трябва да се бъркат с виртуалните асистенти, които са хора които работят дистанционно и следователно могат да се справят с всякакви задачи. По-скоро гласови асистенти са базирани на технологии. Тъй като гласовите асистенти стават по-здрави, тяхната полезност както в личните и бизнес области също ще растат. Проектът, озаглавен „гласов асистент“, е самостоятелно приложение, което предоставя уникален потребителски интерфейс за изпълнение на ежедневни задачи в системата. Този софтуер може да помогне на потребителите да постигнат задачи, които обикновено биха направили сами. Причината за това е, че изговарянето на командата ще бъде по-лесно и по-бързо от въвеждане или щракване. То предлага потребителски интерфейс на естествен език за клиентите. Приложението може да приеме въвеждане в Интелигентен гласов асистент. Това може да се вземе от микрофона, който повечето устройства вече са вградени в тях. Получава се съответната команда и изпълнен. След като командата приключи изпълнението, се осигурява подходяща обратна връзка продължаване на процеса на разпознаване. Приложението е комбинация от Python и Електронни среди.

Инсталиране на SpeechRecognition

SpeechRecognition е съвместим с Python 2.6, 2.7 и 3.3+, но изисква някои допълнителни стъпки за инсталиране за Python 2. В нашият случай ще го инсталираме на Python 3.11.1

Можем да инсталираме SpeechRecognition от терминал с pip:

pip install SpeechRecognition

Преди да продължите, ще трябва да изтеглите аудио файл. Този, който използвах, за да започна, „harvard.wav.  Уверете се, че сте го запазили в същата директория, в която се изпълнява вашата Python проект

SpeechRecognition прави работата с аудио файлове лесна благодарение на своя удобен AudioFileклас. Този клас може да се инициализира с пътя към аудио файл и предоставя интерфейс на контекстния мениджър за четене и работа със съдържанието на файла.

В момента SpeechRecognition поддържа следните файлови формати:

* WAV: трябва да бъде във формат PCM/LPCM
* AIFF
* AIFF-C
* FLAC: трябва да е оригинален FLAC формат; OGG-FLAC не се поддържа

Цел на проекта

Този проект има за цел да предостави услугите на гласово управление. Приложението може да манипулира системата изпълнява основни задачи чрез разпознаване на реч. В днешния свят на нарастващо търсене за лични асистенти този проект предлага решение, което опростява функционалността на днешни асистенти. Стабилният растеж на тази технология в различни области си доказа своята компетентност в сравнение с автоматизацията, базирана на текст и изображения. Този проект е самостоятелно приложение, което може да се използва от всеки, който не го използва познават работата на системата. Може да изпълнява задачи като отваряне на приложение (като настройки или калкулатор), отваряне на часовника или аларми, контролиране на силата на звука или яркостта и т.н. Тези операции могат да се извършват от произнасяне на командите на глас. Може да анализира и синтезира реч и тук, поддържаният език ще бъде български. Гласовият асистент позволява на потребителите да имат интерфейс със свободни ръце към своите устройства. За потребители, които не знаят как да работят със система, това може да се окаже много полезно. Може да показва уебсайтове, отваряйте други. То ще чуе изречените думи, разпознават думите, преобразуват изречената фраза в текст и след това обработват текущата задача. Машинно изпълнимите команди ще взаимодействат с други приложения в системата и предоставя подходяща обратна връзка на потребителя, който ще бъде уведомен, че задачата е изпълнена успешно. Този проект може да бъде разширен, за да бъде приложен върху различни приложения. Може да се използва на персонални компютри, работещи с всяка операционна система.

Характеристика на проекта

Приложението е базирано на технологията за обработка на естествен говор. Той ще осигури интерфейс със свободни ръце между потребителите и системата. Разпознаването на реч включва превод на изговорени думи или фрази в машинно четим формат чрез анализиране на говорния сигнал. Акустичните етикети са комбинирани с езикови модели за създаване на текстова версия на изговорените команди. Обработката на естествен език се използва за наблюдение и контрол на взаимодействието между хората и компютрите. Той анализира данните от естествения език и помага за трансформирането за въвеждане на реч в текст или обратно. Този проект ще позволи на потребителите да имат достъп и да използват функциите на гласов асистент, като Alexa и Siri. Той ще слуша изговорените команди и ще идентифицира задачите ще бъде изпълнено. Той ще изпълнява операции в системата като отваряне на приложение, търсене онлайн, регулиране на звука, настройване на аларми или отговаряне на запитвания. Дори човек, който не знае как да работи със системата, може лесно и ефективно да получи достъп с помощта на такова приложение. В този проект автоматизацията се извършва с помощта на Python. Този език за програмиране позволява взаимодействие със системата чрез клавишни комбинации, използвайки „pyautogui“ пакет. Чрез програмиране на Python ние също имаме достъп до сървъра на Google, запис, аудио от системния микрофон. Има два различни Pythonфайлове, включени в този проект, един за аудио разпознаване и един за изпълняваща системи задачи. Файловете на Python се изпълняват като дъщерни процеси (процеси на рендеринг) заедно с основния електронен процес, Тези моменти са, когато потребителят взаимодейства със системата или таймерът се изключва.

ГЛАСОВО РАЗПОЗНАВАНЕ

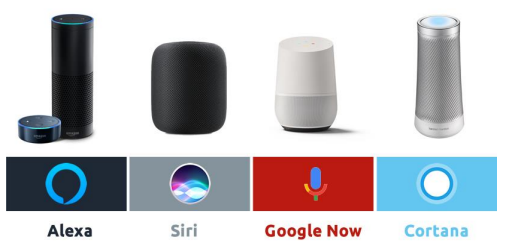
Разпознаването на реч е способността на машината да разпознава думи, изречени в някой език и превежда думите в машинно четим формат. Асистентът е софтуер, който изпълнява задачи на потребителя. Тези асистенти могат да работят с текст, реч или изображения. Гласовите асистенти извършват операции на устройства, базирани наустна команда.

Най-широко използваните гласови асистенти са Siri на Apple, GoogleAssistant, Alexa на Amazon и Cortana на Microsoft. До голяма степен инсталирани са в смартфони, настолни компютри и самостоятелни устройства, тези софтуерни агенти обикновено се интегрират с операционната система и предоставят средство за подобрена достъпност при изпълнение на всяка задача .В началото на 60-те години на миналия век Уилям К. Дерш от IBM създава „Кутия за обувки“, първата в историята реч машина за разпознаване, която можеше да разбере общо шестнадесет думи и да изпълнява математически изчисления, базирани на изречената командна фраза. Предшественикът на днешните системи с гласово управление, имаше хардуер, който се състоеше от микрофон, вкойто потребителят ще произнесе командата. Машината не беше като последните иновации и изискваше всяка дума да бъде ясно, ясно и бавно, с паузи между тях. Въпреки това, следващите 57 години донесе много подобрения в технологиите с навлизането на интернет и облачните изчисления, което доведе до разрастването на виртуалните асистенти, каквито ги познаваме днес. IBMShoebox започна революция със своята технология.

**Гласови асистенти**

С навлизането на интернет и изкуствения интелект, гласовите асистенти са едно цяло от най-често използваните софтуери днес. Иновации като Siri, Alexa и Cortana направят нашия личен и професионален живот по-лесен и по-ефективен. Над деветдесет милиони души само в САЩ използват гласови асистенти поне веднъж месечно. Според изследвания, около 20% от търсенията в Google се правят чрез гласова среда, като най предпочитан начин за комуникация между хора и устройства. Те са дигитализирали джаджи до такава степен, че да могат да намират упътвания, провеждат обаждания, резервират срещи и дори поръчват храна чрез проста гласова команда. Тези асистенти имат достъп до голяма колекция от онлайн и сървърни данни, до които се осъществява достъп бързо да интерпретира и изпълнява желаните задачи, според избора на потребителя. Проучване установи, че около 600 милиона души по света използват гласови приложения поне веднъж седмично под формата на Google Assistant, Amazon Alexa, Apple Siri или друг софтуер с гласово управление. С появата на интелигентните високоговорители и домакински уреди, гласовите асистенти се използват все повече. В началото търсачките се нуждаеха от изключително точни и кратки данни, за да извършване на основни търсения. С възхода на IoT и smart приложения, такива асистенти могат да взаимодействат не само с едно устройство, но и да автоматизират много уреди в къщата.

Фиг. 2 популярни гласови асистенти



Някои от функциите, които гласовият асистент може да изпълнява, включват отключване на устройства, отваряне на приложения, провеждане на повиквания, изпращане на съобщения, извличане на последните новини, заснемане изображения, възпроизвеждане на музика, задаване на напомняния или аларми, изпращане на имейл до колеги, изпълнение онлайн търсения, резервиране на билети и предлагане на препоръки за храна, развлечения и още. Основната причина за преминаването от традиционни системи към гласови потребителски интерфейси е поради променящите се потребителски изисквания. Постоянното подобряване и оптимизиране на бързината, точността, ефективността и удобството доведоха до необходимостта от гласово управление на системи. Друг основен фактор е растежът на изкуствения интелект във всяка фаза от живота ни с нарастващия брой IoT устройства като интелигентни хладилници, термостати, телевизори, високоговорители и микровълнови печки, животът на потребителите става все повече свързани и гласови асистенти помагат да се създаде тази връзка между всички уреди. В банковия сектор гласовата технология позволи на клиентите да проверяват своите балансират и плащат сметките си с помощта на гласовия асистент. В рекламата, човешките ресурси и маркетинга, гласовите асистенти ще формират интерфейса между клиенти и компании, позволявайки по-бързо, по-ефективен анализ на данни и грижа за повтарящите се и транзакционни фази на индустрията. Дори в търговията на дребно гласовите асистенти дадоха на клиентите възможността да поръчат всеки продукт онлайн само чрез няколко прости команди. Това се отнася и за обществения транспортен сектор, където потребителите могат да запазят такси или да направят запитване за разписанията на полетите с помощта на гласовите асистенти. Всички индустрии вече са наситени с множество продукти които са се опитали да интегрират гласова технология, включително развлекателни сектори. Има много технически предизвикателства, пред които са изправени разработчиците, когато опит за създаване на гласов асистент. Първо, има отделни процеси за всяка стъпка от входа до изхода, вместо рационализиран, интегриран подход, който позволява на машините действително да разбират командите. Това означава, че въвеждането, разпознаването, обучението и декодирането фазите са отделни единици. По този начин машината всъщност не разбира команди, но просто преминава от една фаза към следващата, нанасяйки всеки вход на изход. Поради това интелигентността на машината е ограничена до определено ниво и не се изравнява с човешкото поведение. На второ място, има липса на съобразени с контекста отговори от машината. Машината не взема под внимание гласа на потребителите, интонациите, възприеманите емоции, ситуациите на околната среда или други контекстуални фактори това понякога може да доведе до отговори, които не отговарят на очакванията на потребителя. И накрая, има трудности при постигането на 100% точност поради безкрайните вариации в реч. Има безброй акценти и вариации в самият език, с речников запас достигайки до милиони. Тъй като клиентите очакват гласовите асистенти да разбират добре няколко езика, става трудно да се постигне 100% точност, поради сумата от данни и безкрайните възможности по отношение на вариации или произношения. Има няколко ключови фактора, които влизат в действие при обмислянето на създаването на интелигентен личен асистент. Те включват индивидуални и обширни разработки които обхващат следните полета: реч към текст, текст към реч, шум управление, компресиране на реч, гласова биометрия и гласов потребителски интерфейс. Всеки от тези областите представляват важна част от развитието на гласовия асистент и трябва да бъдат подготвени ефективно, като се отдава значение на тяхната точност и бързина. В бъдеще гласовите асистенти ще стават все по-сложни. Някои от подобренията, които очакваме, ще бъдат персонализирани и съобразени с контекста. Тъй като пазарът за това технологията непрекъснато се разраства, това затруднява разработчиците да създават продукти, които се открояват от съществуващите системи. Има няколко платформи за разработчици с отворен код които са пуснати от много компании, които позволяват на програмистите да създават и персонализират своя собствен виртуален асистент. Тези независими услуги съдържат модулите за разпознаване на реч, библиотеките за гласови проби, алгоритмите за машина обучение и средствата за преобразуване и картографиране на текст в изпълними команди.

**Предимства от гласовите асистенти**

* По-бързо е от писане или кликване.
* Повишава производителността.
* Може да бъде полезно за хора с увреждания.
* Намалява забавянията и социалните пристрастия по отношение на сектора за обслужване на клиенти.

**Недостатъци от гласовите асистенти**

* Гласовите асистенти наскоро бяха обект на опасения за поверителността поради интерфейс за слушане, който има достъп до лична информация.
* Машините отнемат време, за да се настроят и научат.
* По-малко точен е в случай на непрекъснат поток от думи или фонов шум.
* Не винаги е рентабилно.
* Склонен е към хакери.

**Защо избирам Python за разработка на гласовия асистент**

Python е интерпретиран език за програмиране на високо ниво, проектиран от Guido van Росъм. Той е написан основно, за да осигури език, който има прост синтаксис и е четлив. Поради по-кратките кодове и лекотата на писане, програмистите започнаха все повече придържайте се към Python за програмиране. Освен това има много вградени функции и може да работи като обектно ориентирано, функционално или процедурно програмиране. Освен това е независим от платформата. Той е безплатен и с отворен код и също така има огромна поддръжка от библиотеки, може да се използва за извършване на огромни разнообразие от действия и програмистите го намират за по-лесно за научаване и прилагане в сравнение с други езици. Освен това има обработка на изключения и вградено управление на паметта. Тъй като е динамично въведен, няма декларации, което го прави компактно и сбито. Най-важната част от Python е отстъпът, тъй като той определя поток от изявления. Python също така съдържа изкуствен интелект и обработка на естествен език библиотеки, което го прави полезен в тези области. Използва се и в информационната сигурност и като основен език за Raspberry Pi. Въпреки това, в сравнение с C/C++, Python е малко по-бавен и не поддържа браузъри и мобилни устройства. Python се инсталира в много компютри с Linux и Mac OS. Ако вашата система няма Python, инсталационната програма може да бъде изтеглена от страницата за изтегляне на уебсайт python.org. Стартиране на инсталатора и избиране на опцията за добавяне към променлива PATH ще настрои Python файловете. Има две версии на Python: 2.x и 3.x. Малко са незначителни разлики между тях като функция за печат, обработка на грешки и оператор за разделяне. Има също много начини за изпълнение на код в Python. Един от начините е да напишете „python“. командния ред под инсталираната папка, която ще отвори обвивката на Python. поредица от твърдения. Друг начин е да използвате IDE (интегриран софтуер развитие) като PyCharms. В Python има няколко важни библиотеки като: • Библиотеката “Bokeh” за визуализация на данни, която показва информативни диаграми

• Пакетът “Numpy” за обработка на масиви за научни изчисления на многомерни масиви

• Библиотеката “keras”, която е пакет за невронни мрежи с отворен код

• “Pandas DataFrame”, която е двуизмерна таблична структура от данни с две оси. Освен това има множество методи за внедряване на GUI (графичен потребителски интерфейс). от които „tkinter“ е един от най-популярните. Също така има и модули за математически функции, календар, време, автоматизация и обработка на URL адреси. Следват някои от предимствата на Python.

• Наличие на модули на трети страни

• Лесен за научаване

• Динамично въведен

**Анализ на изискванията**

Изискванията, които определят какви услуги една система може да предостави на крайния потребител се наричат функционални изисквания. Функционалните изисквания са тясно свързани с изискванията на потребителските спецификации. Това може да включва изчисления, обработка на данни, технически операции и друга такава функционалност, която има за цел да изпълни целите на приложението. Всички операции по проследяване, законови изисквания, подробности за интерфейса, нива на оторизация, актуализации на транзакции и административни функции отговарят на функционални изисквания. Техническата архитектура на системата е определени от тези изисквания. Функционалните изисквания на този проект са:

• Трябва да показва приятен потребителски интерфейс, който клиентът да разбира системни процеси.

• То трябва да слуша изречените команди и да разпознава чутите думи. • Трябва да предоставя подходящи указания на потребителя за лесна употреба.

• Трябва да дава признание за процеса на разпознаване чрез някои анимация и комуникации.

• Трябва да може да изпълнява задачите, които потребителят изисква чрез Python за автоматизация.

• Трябва да се опреснява или презарежда след всяка команда и да изчиства всеки допълнителен кеш използвана памет.

• Трябва да поддържа подходящи времеви ограничения за изпълнение, докато записва аудио или предоставяне на резултати.

• Трябва да доставя съобщения за грешка винаги, когато е необходимо.

Ефективността се оценява чрез следните спецификации:

• Време за реакция: Това е времето, необходимо на системата да приеме въвеждане от потребителя и да отговори към него чрез показване на някакъв резултат. Обикновено съобщенията за обратна връзка се показват в рамките интелигентен гласов асистент. Максимум 10 секунди на диалог прозорец гарантира, че потребителят няма да загуби интерес или ход на мисълта. Отговорът времето също трябва да е последователно и да не варира в зависимост от броя на едновременните сесии.

• Работно натоварване: Това е количеството стрес или работа, което системата може да издържи наведнъж. Това може да бъде по отношение на паралелни сесии, брой активни потребители или брой транзакции с бази данни. Работното натоварване обикновено се описва като сценарии, които потребителите най-вероятно ще срещнат. Специални случаи като сценарии за грешка, архивиране и исканията за управление трябва да се вземат предвид при уточняването на натовареност.

**Архитектура на системата**

Процесът на проектиране на системната архитектура се фокусира върху разбиването на система в различни компоненти и техните взаимодействия, които удовлетворяват функционални и нефункционални изисквания. Входовете за проектиране на софтуерна архитектура са изискванията и хардуерната архитектура. Този проект няма периферия хардуерни устройства, така че софтуерните компоненти взаимодействат само със системата микрофон и дисплей.

Диаграма на употреба

Въвеждане

Слушане на изречени команди

*микрофон*

Разпознаване на език и превръщане в текст

Валидиране на потребителя

*Изпълняване на задачата*

*монитор*

*Потребител*

Диаграмата на системната последователност показва събитията, които са генерирани от външни потребители в последователен ред, за конкретен сценарий на случай на употреба. Те са начин за визуално обобщаване на това, което се случва в един единствен случай на употреба.

