Dokumentation KOMMI

1. Text-Summarization

Die Text-Summarization findet in der KI-Engine des Voicebots statt. Es wurde die Python-Bibliothek „sumy“ verwendet.
Wenn der KI-Engine klassifiziert, dass die Intention des Users „service-BW Preamble ist, dann wird der Antwort-Text zusammengefasst. Dafür wird die Funktion *sumy_summary(antwort)* verwendet (siehe Abb1).

```
# Sumy Pkg
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
from sumy.summarizers.lex_rank import LexRankSummarizer

# Sumy
def sumy_summary(antwort):
    parser = PlaintextParser.from_string(antwort,Tokenizer("german"))
    lex_summarizer = LexRankSummarizer()
    summary = lex_summarizer(parser.document,2)
    summary_list = [str(sentence) for sentence in summary]
    result = ' '.join(summary_list)
    return result
```

Abb 1.: Funktion sumy_summary

2. „Text-to-Speech“ und „Speech-to-Text“

Die Spracherkennung (Speech-to-Text) wie auch die Sprachsynthese (Text-to-Speech) wurde mit dem Web Speech API realisiert.

Die Web Speech API ist eine Spezifikation der Speech API Community Group innerhalb des W3C, um die Nutzung von Funktionen zur Sprachsynthese und Spracherkennung mittels JavaScript in Webbrowsern zu ermöglichen.

Weitere Information und Dokumentation zum Web Speech API kann man hier finden.

Spracherkennung

Bei KOMMI wird die Funktion *startrecorder()* aktiviert wenn der User auf dem Mikrofon-Button klickt (siehe Abb 3).

Zunächst werden die Konstanten *SpeechRecognition* und *recorder* erstellt, um eine neue Instanz des SpeechRecognition-Objekts zu erzeugen (siehe Abb 2).

Als nächstes wird die Sprache (‚de-DE') der Spracherkennung definiert (siehe Abb 2).

```
const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
const recorder = new SpeechRecognition();
recorder.lang = 'de-DE';
```

Abb 2.: SpeechRecognition-Objekt

Bei der Funktion startrecorder() wird für die vorher erzeugte Instanz des SpeechRecognition-Objekts die start()-Methode der Web Speech API aufgerufen (siehe Abb 3), um den Spracherkennungsdienst zu starten.
Abschließend wird das Ereignis onresult der Web Speech API ausgelöst (siehe Abb 3), wenn der Spracherkennungsdienst ein Ergebnis zurückgibt, bzw. ein Wort oder eine Phrase positiv erkannt wurde. Dies wird an KOMMI zurückgemeldet und in Message Text geschrieben.

```
    function startrecorder (){

        recorder.start();
        console.log('Ready to receive the voice.');
    }

    recorder.onresult = function (event) {
  for (var i = event.resultIndex; i < event.results.length; ++i) {
    if (event.results[i].isFinal) {
        var str = event.results[i][0].transcript;
var newstr = str.replace(/Conny/i, "Kommi");
      Message.value += newstr;
    }

  }
}
```

Abb 3.: Funktion startrecorder

Sprachsynthese

Wenn der User auf den Button klickt, um seine Frage in KOMMI abzusenden, wird die Funktion *Reply()* aktiviert (siehe Abb 4). Diese Funktion kommuniziert mit der KI-Engine des Voicebots. Sie sendet die Frage zur KI-Engine und bekommt die Antwort auf die Frage von ihr. Die Antwort wird im Chatfenster mit der Funktion *drawMessage(res)* dargestellt.

```
function Reply() {
var msg = msgText.val()
    if (msg == '') {
        toastr.error("Message can not be empty")
        return;
    }

    var data = {
        "msg": msg, "userId":$("#userId").val(), "startcontext": ""
    }
    var id= $("#userId").val()
    set_id(id)
    set_frage(msg)

    $.ajax({
        url: '/kommi/kommi',
        data: JSON.stringify(data),
        method: 'POST',
        success: function (res) {
            drawMessage(res)

        },
        error: function (err) {
            if (err.desc == null)
                err.desc = "Something went wrong"
            toastr.error(err.desc)
        }
    })
```

Abb 4.: Funktion Reply

Von der Funktion *drawMessage(res)* wird die Funktion *drawAnswer(res)* ausgelöst und von der Funktion *drawAnswer(res)* wird die Funktion *botVoice(message)* ausgelöst (siehe Abb 5).

```
var msgContent = $('#MessageContent');

function drawMessage(res) {
    msgContent.append("<div class='col-md-6 owner-message'><p>"+res.ques+"</p></div>");

    scrollDown();
    drawAnswer(res)
}
function drawAnswer(res) {

    var q= jQuery("<div></div>")
    q.addClass("col-md-6").addClass("guest-message")

    if (Array.isArray(res.res)) {
        q.append("<p>"+paint(res.res[0])+"</p>")
    } else q.append("<p>"+paint(res.res)+"</p>");



     msgText.val('')

    q.append('<div class="feedback"><ul class="feedback__buttons"><li><button class="feedback__button feedback__button--positive gut" aria-lab



    q.appendTo(msgContent)
    scrollDown();
    console.log('res.res[0]', res.res[1])
    if (Array.isArray(res.res)) {
        botVoice(res.res[1])
    } else botVoice(res.res);


}
```

Abb 5.: Funktionen drawMessage und drawAnswer

Für die Realisierung der Sprachsynthese wird die Funktion *botVoice(message)* verwendet (siehe Abb 6).

Erst wird die Konstante *speech* erstellt, um eine neue Instanz des SpeechSynthesis-Objekts zu erzeugen (siehe Abb 6).

Zunächst wird für die vorher erzeugte Instanz des SpeechSynthesis-Objekt die *speak()*-Methode der SpeechSynthesis-Schnittstelle aufgerufen, um den Antwort-Text vorzulesen (siehe Abb 6).

```
function botVoice(message) {
    const speech = new SpeechSynthesisUtterance();
    console.log(message)
    speech.text = message
    set_voiceaktuell(message)
    speech.volume = 1;
    speech.rate = 1;
    speech.pitch = 1;
    window.speechSynthesis.speak(speech);

}
```

Abb 6.: Funktion botVoice

Beim „Klicken“ des Buttons „Play“ wird die Funktion *play()* ausgelöst und somit wird der aktueller Text nochmal vorgelesen (siehe Abb 7).

Beim „Klicken“ des Buttons „Stop“ wird die Funktion *stop()* ausgelöst (siehe Abb 7). Somit wird die *cancel()*-Methode der SpeechSynthesis-Schnittstelle aufgerufen(siehe Abb 7), um das Vorlesen des Antwort-Textes zu stoppen.

```
function play(){

console.log('voiceaktuell', voiceaktuell)
        const speech = new SpeechSynthesisUtterance();

        speech.text = voiceaktuell

        speech.volume = 1;
        speech.rate = 1;
        speech.pitch = 1;
        window.speechSynthesis.speak(speech);

}

function stop(){

        speechSynthesis.cancel();

};
```

Abb 7.: Funktionen play und stop