

Aktueller Stand

Der Bot war ständig nicht erreichbar

Der Bot läuft via docker-compose in einer VM. Immer wenn der Bot nicht erreichbar war, startete er neu sobald sich jemand in die VM einloggte und war dann auch wieder erreichbar. Der Grund dafür war, dass docker-compose so konfiguriert war, dass die Container zwar neu starten sollten, aber nicht sofort wenn sie abstürzten (sondern in diesem Fall dann eben beim Login durch einen docker-user).

Der Lösung bestand entsprechend in der Änderung der Configuration nach „restart-always“:
<https://github.com/beuthbot/beuthbot/pull/3>

Gateway funktionierte nicht ohne Telegram-ID

Bei ersten Experimenten ist aufgefallen, dass wenn eine Nachricht an das Gateway geschickt wird und diese keine valide Telegram-ID enthielt, wurde die Nachricht ignoriert. Dieses war entgegen der Dokumentation, welche die Telegram-ID als optional definierte. Da dies für Testzwecke sehr hinderlich ist und im Projektverlauf zwei weitere Messenger (Discord und eigene Webseite) hinzugefügt werden sollen, galt dieses als eines der ersten Probleme die behoben werden sollten.

Durch eine Anpassung des Gateways bei der User-Abfrage wird eine valide Telegram-ID nicht vorausgesetzt: <https://github.com/beuthbot/gateway/pull/2>

Continous Deployment

Zuvor musste man um eine neue Version des BeuthBots zu deployen, musste manuell die neuste Version von GitHub gepulkt werden und der Docker-compos-Befehl ausgeführt werden. Mithilfe eines Selfhosted-Runners welcher auf dem BeuthBot-Server installiert wurde, ist es jetzt möglich diesen Prozess zu automatisieren. Sobald ein Git-Commit mit einem Versions-Tag gepusht wird, wird dies vom Runner erkannt und der Deploy-Prozess wird angestoßen.

<https://github.com/beuthbot/beuthbot/pull/4>

Text 2 Speech

- Say.js
 - <https://www.npmjs.com/package/say>
 - <https://github.com/marak/say.js/>
- 2. Web Speech API
 - https://wiki.selfhtml.org/wiki/JavaScript/Web_Speech
- 3. Text2Speech
 - <https://www.npmjs.com/package/text-to-speech-file>

Speech 2 Text

Mozilla Voice STT (DeepSpeech)

<https://github.com/mozilla/DeepSpeech> <https://github.com/AASHISHAG/deepspeech-german>

- Opensource
- Offline nutzbar
- Viel Dokumentation
- Deutsches Modell
- WER: 15%
- Zukunft ungewiss

Kaldi

<https://github.com/kaldi-asr/kaldi> <http://kaldi-asr.org/doc/about.html>

- Opensource
- Offline nutzbar
- Deutsche Modelle
- WER: 8,44%

Wav2Letter

<https://github.com/facebookresearch/wav2letter>

- Opensource
- Offline nutzbar
- Deutsche Modelle
- WER: 4%

Espresso

<https://github.com/freewym/espresso>

- Opensource
- Offline nutzbar
- Kein deutsches Modell

Nvidea OpenSeq2Seq

<https://github.com/NVIDIA/OpenSeq2Seq>

- Opensource
- Offline nutzbar
- Kein Deutsches Modell

WER Vergleich 2017

- Google (8%)
- Microsoft (5.9%)

- IBM (5.5%)
- Apple (5%)
- Baidu (16%)
- Hound (5%)

Quelle:

<https://askwonder.com/research/current-voice-recognition-word-error-rates-google-amazon-microsoft-ibm-apple-5b88trj0t>

Nutzungshinweis: Auf dieses vorliegende Schulungs- oder Beratungsdokument (ggf.) erlangt der Mandant vertragsgemäß ein nicht ausschließliches, dauerhaftes, unbeschränktes, unwiderrufliches und nicht übertragbares Nutzungsrecht. Eine hierüber hinausgehende, nicht zuvor durch *datenschutz-maximum* bewilligte Nutzung ist verboten und wird urheberrechtlich verfolgt.