



Open Audio Search

Endbericht | Call 15 | Projekt ID 5198

Lizenz CC-BY-SA

Inhalt

1	Einleitung.....	3
2	Projektbeschreibung	3
3	Verlauf der Arbeitspakete	4
	3.1 Arbeitspaket 1 - Detailplanung und Formales am Projektstart	4
	3.2 Arbeitspaket 2 - NLP.....	4
	3.3 Arbeitspaket 3 - Trainingspipeline/Evaluationspipeline	5
	3.4 Arbeitspaket 4 - Hardware	6
	3.5 Arbeitspaket 5 - ASR-Pipeline an OAS-Core.....	7
	3.6 Arbeitspaket 6 - Preprocessing & Analyse.....	7
	3.7 Arbeitspaket 7 - Dokumentation und Formales während des Projekts und am Projektende	8
4	Umsetzung Förderauflagen	9
5	Liste Projektergebnisse	9
6	Verwertung der Projektergebnisse in der Praxis	9
7	Öffentlichkeitsarbeit/ Vernetzung.....	11
8	Eigene Projektwebsite.....	11
9	Geplante Aktivitäten nach netidee-Projektende.....	11
10	Anregungen für Weiterentwicklungen durch Dritte.....	11

1 Einleitung

Was für ein Jahr! Nach unzähligen Online-Calls und gemeinsamen Programmiersessions konnten wir unser Projekt erfolgreich beenden. Open Audio Search ist lauffähig und die Speech to Text Engine liefert brauchbare Ergebnisse, um Audioinhalte durchsuchbar zu machen. Unter <https://demo.openaudiosearch.org> kann unsere Software ausprobiert und erkundet werden. Wir sind gespannt, wohin uns die weitere Reise mit OAS führen wird...

2 Projektbeschreibung

Im Internet gibt es große und schnell wachsende Bestände an interessanten Radio-Features und Podcasts, die unter freien Lizenzen veröffentlicht sind. Ein gutes Beispiel ist unsere eigene Plattform, das Cultural Broadcasting Archive (cba.media) oder auch freie-radios.net. Auf diesen Plattformen ist eine Vielzahl an relevanten Audio-Inhalten archiviert, aber die einzelnen Beiträge und Features sind oft nur schwer wiederaufzufinden. Insbesondere für ehrenamtlich getragene Projekte bleibt oft wenig Zeit, um Beschreibungstexte, Metadaten oder gar Transkripte einzupflegen, nachdem die eigentliche Produktion abgeschlossen ist – das gilt umso mehr für Inhalte, die vor zwanzig Jahren produziert wurden und nun im Rahmen von Digitalisierungsprojekten online gestellt werden.

Hier wollen wir mit Open Audio Search Abhilfe schaffen, einer intelligenten Suchmaschine für Audiodateien. Über automatische Spracherkennung wird gesprochene Sprache in Text umgewandelt, der dann in einer Volltextsuchmaschine indiziert wird. Die Engine kann RSS-Feeds abonnieren, um Inhalte von verschiedenen Quellen zu importieren. Über eine webbasierte Oberfläche können Nutzer*innen die generierten Transkripte von Radiosendungen und Podcasts durchsuchen und direkt von der Zeitmarke der Suchergebnisse abspielen. Open Audio Search wird als kollaboratives Open-Source-Projekt entwickelt und kann leicht auf einem beliebigen Server installiert werden.

Mit Open Audio Search wollen wir dazu beitragen, eine zukunftssichere und gut wartbare Open-Source-Plattform für community media discovery herzustellen. Wir hoffen, dass wir damit Community-Media-Produzent*innen mit mehr Reichweite und einem breiteren Publikum versorgen können. Denn wenn Inhalte nicht so schnell nach ihrer Produktion wieder verschwinden, sondern stattdessen Teil eines lebendigen und leicht durchforstbaren Archivs werden, dann gehen diese Inhalte auch nicht so schnell wieder verloren, wie es derzeit oft der Fall ist.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - *Detailplanung und Formales am Projektstart*

Kurzbeschreibung der Haupttätigkeiten

- Vertrag unterschrieben, Detailprojektplan (Arbeitsblatt Arbeitspakete) erstellt und abgenommen,
- detaillierte Liste Projektergebnisse mit Lizenz und Ort der öffentlichen Bereitstellung erstellt und abgenommen (Arbeitsblatt Projektergebnisse)
- Projekt-Website in Betrieb & erster Blogeintrag erstellt
- erste Förderrate beantragt; ggf. nach Feedback ergänzt
- Förderrate1 genehmigt

Erkenntnisse zur Vorgangsweise

- Keine besonderen Erkenntnisse

Kurzbeschreibung der erreichten Ergebnisse

- Siehe Projektplanung

Besondere Erfolge/ Probleme

- Keine besonderen Erfolge und Probleme

Gab es große Abweichungen zum Plan? Warum?

- Nein

3.2 Arbeitspaket 2 - *NLP*

Kurzbeschreibung der Haupttätigkeiten

- NER Modul, das ASR Results nimmt und Entities erkennt.
- Punctuation Modul, das Transkripte nimmt und Interpunktion wiederherstellt.
- Erstellung eigener Knowledgebase (CBA Keywpords, Wikidata)
- Abgleich und Korrektur der Resultate mit Knowledgebase

Erkenntnisse zur Vorgangsweise

- Zu den Ergebnissen der Named Entity Recognition haben wir einen Blogbeitrag verfasst. Insgesamt war die erste Umsetzung ohne größere Probleme zu bewerkstelligen. Unsere Experimente das frei-verfügbare spaCy Modell anzupassen waren erfolgreich, jedoch hatten wir nicht ausreichend annotierte Trainingsdaten, um ein sinnvoll einsetzbares Modell zu trainieren.

Jedoch ist der Wechsel des Modells sehr einfach und die nötigen Softwarekomponenten sind integriert.

- Die Punctuation Reconstruction konnten wir mit einem Modul des von uns eingesetzten Vosk-Toolkits einfach umgesetzt werden.
- Abgleich und Korrektur der Resultate mit Knowledgebase findet statt jedoch nicht auf der eigenen Knowledgebase sondern mit der Wikibase API.

Kurzbeschreibung der erreichten Ergebnisse

- Automatische Tagging von Eigennamen von Personen und Orten.
- Verlinkung der extrahierten Eigennamen auf Einträge der Wikipedia und Wikidata.
- Punctuation der Transkripte wird automatisch wieder hergestellt.
- Abgleich und Korrektur der Resultate mit Knowledgebase findet statt, jedoch nicht auf der eigenen Knowledgebase sondern mit der Wikibase API.

Besondere Erfolge/ Probleme

- keine

3.3 Arbeitspaket 3 - *Trainingspipeline/Evaluationspipeline*

Kurzbeschreibung der Haupttätigkeiten

- Docker container, der die gesamte Trainingspipeline einschließt.
- Pfade für Daten bzw. Modelldateien etc. werden in dem Container gemounted.
- Grundsätzlich sollen existierende Kaldi-Skripte zum Trainieren (Bash, Python, ausführbare Dateien) wiederverwendet werden.
- Datapreparation
- Skripte zum automatischen Alignieren von Transkripten und Audiodaten
- Skripte zum Import von Daten und Konvertieren in ein Format, das für Training verwendet werden kann.
- Evaluationspipeline ist entwickelt und implementiert.

Erkenntnisse zur Vorgangsweise

- Die Trainingspipeline trainiert sogenannte “chain” Modelle, das sind akustische Modelle, die unter Verwendung von Kaldi mit lattice-free maximum mutual information (LF-MMI) Zielfunktion trainiert werden. Für diese Art von Modellen ist es erforderlich, sogenannte Alignments für die Trainingsdaten zu erzeugen. Um dies zu bewerkstelligen, werden kleinere ASR-Modelle auf Basis der Daten trainiert, die im Anschluss zur Erzeugung der Alignments verwendet werden.

Kurzbeschreibung der erreichten Ergebnisse

- Eine Trainingspipeline auf Basis der derzeit modernsten Kaldi-Rezepte (“chain2”) für das Trainieren von akustischen Modellen wurde entwickelt.

- Eine Pipeline zum Alignieren und Bereinigen von transkribierten Daten erstellt. Diese Pipeline ist in der Lage, Transkripte, die nur ungefähr mit den Audiodaten übereinstimmen, zu verarbeiten, um für das Trainieren von Modellen verwendbare Trainingsdaten zu erzeugen.
- Programme wurden erstellt, um größere Mengen neuer Daten für das Trainieren von ASR Modellen aus dem Internet herunterzuladen (zu “scrapen”).
- Programme wurden erstellt, um existierende Korpora sowie große Mengen an neuen Daten für Training sowie Evaluierung vorzubereiten. Dies inkludiert Konvertieren der Dateien sowie Formatieren in das Kaldi-Datafolder Format.
- Evaluation von Wortfehlerrate ist derzeit als Teil der Trainingspipeline implementiert.

Besondere Erfolge/ Probleme

- Modelle basierend auf rel. geringen Datenmengen wurden trainiert, um die Funktionstüchtigkeit der Trainingspipeline zu Verifizieren.
- Das Trainieren von Modellen auf größeren Datenmengen, die eine höhere Performanz aufweisen, ist derzeit nicht möglich, da dafür vorgesehen Grafikkarten noch nicht gekauft werden konnten aufgrund der Preissteigerungen.
- Derzeit ist die Trainingspipeline noch nicht in Docker container enkapsuliert.

Gab es große Abweichungen zum Plan? Warum?

- Bislang keine größeren Abweichungen

3.4 Arbeitspaket 4 - Hardware

Kurzbeschreibung der Haupttätigkeiten

- Hardware-Infrastruktur für Trainings- und Evaluationspipeline ist eingerichtet.
- Erkenntnisse zur Vorgangsweise
- Kurzbeschreibung der erreichten Ergebnisse
- Server wurde gekauft, zusammengebaut und eingerichtet
- Setup für Demo-Instanz wurde eingerichtet
- Demo-Instanz läuft unter <https://demo.openaudiosearch.org>
- Setup für Trainingspipeline ist vorbereitet

Besondere Erfolge/ Probleme

- Erfolge: Lauffähige Demo-Instanz (s.o.)
- Probleme: Bislang keine, abseits von üblichen kleineren Problemen im Setup-Prozess, die sich alle schnell lösen ließen

Gab es große Abweichungen zum Plan? Warum?

- Bislang wurden die Grafikkarten noch nicht gekauft aufgrund der aktuellen hohen Preissteigerungen
- Da Franz Heinzmann in Freiburg lebt, war für den Ankauf, den Zusammenbau und die Einrichtung die Mitarbeit von Thomas Diesenreiter am Serverstandort vor Ort in Linz notwendig.

3.5 Arbeitspaket 5 - ASR-Pipeline an OAS-Core

Kurzbeschreibung der Haupttätigkeiten

- ASR-Pipeline ist entwickelt und in Core implementiert.

Erkenntnisse zur Vorgangsweise

- Der Umstieg auf Rust als Programmiersprache im Core hat sich als sehr sinnvoll erwiesen
- Die Arbeit mit dem Vosk-Toolkit als High-Level-Abstraction über Kaldi ist bislang zielführend

Kurzbeschreibung der erreichten Ergebnisse

- Inference-Engine via Vosk läuft im Core
- Anbindung an Modelle aus eigenem Training ist vorbereitet

Besondere Erfolge/ Probleme

- Implementierung in Rust war ein Erfolg
- Etablierung einer effizienten und leicht zu managenden Job-Queue zwischen Rust und Python erweist sich als Herausforderung, die noch nicht gänzlich zufriedenstellend gelöst ist
- Besondere Probleme gab es bislang keine

Gab es große Abweichungen zum Plan? Warum?

- Etwas mehr Zeitbedarf als ursprünglich eingeplant, da teilweise Restrukturierungen am Core vorgenommen wurden, um die entwickelten Features gut unterstützen zu können.

3.6 Arbeitspaket 6 - Preprocessing & Analyse

Kurzbeschreibung der Haupttätigkeiten

- Entwicklung einer Conversion-Pipeline auf Grundlage von ffmpeg.
- Entwicklung einer Lösung zur Diarisierung von Audiofiles, d.h. zum Chunking nach zugeordneter Sprecher:innen-Identität. Evaluation der vorhandenen Engines und Modelle, Auswahl und Integration der Engine pydiar.

- Entwicklung einer Lösung zur Erkennung von gesprochener Sprache und Unterscheidung von Musik oder Noise. Evaluation der vorhandenen Engines und Modelle, Auswahl und Integration der Engine silero-vad.
- Anpassung der ASR-Tasks zur effizienten Unterstützung von Preprocessing und Analyse-Tasks.
- Test und Tuning der verwendeten Komponenten sowie Vorbereitung einer Evaluation der verwendeten Engines
- Entwicklung eines User Interfaces zur Darstellung der Analyse-Ergebnisse (Sprecher*innen-Unterscheidung sowie Unterscheidung Sprache/Nicht-Sprache)

Erkenntnisse zur Vorgangsweise

- In wenigen Fällen erkennt die Engine silero-vad gesprochene Sprache als Musik oder Lärm (v.a. bei umfangreichen Hintergrundgeräuschen). Evtl müssen hier noch die Modelle verbessert oder angepasst werden.
- Die Sprecher*innen-Unterscheidung funktioniert nach ersten Tests relativ gut, kann aber auch noch optimiert werden

Kurzbeschreibung der erreichten Ergebnisse

- Diarisierung und Segmentierung in Sprache/Nicht-Sprache läuft vollautomatisch vor bzw. mit der Audiotranskription
- Transkripte werden im User Interface jetzt in Absätze nach zugeordneter*m Sprecher*in dargestellt. Wenn Teile nicht als Sprache erkannt werden, werden diese ausgegraut.

Besondere Erfolge/ Probleme

- Die Lesbarkeit der erzeugten Transkripte hat sich durch die Unterscheidung in Sprecher*innen-Chunks außerordentlich verbessert.

Gab es große Abweichungen zum Plan? Warum?

Keine größeren Abweichungen

3.7 Arbeitspaket 7 - Dokumentation und Formales

während des Projekts und am Projektende

Kurzbeschreibung der Haupttätigkeiten

- Die geplanten Projektergebnisse (siehe Arbeitsblatt "Projektergebnisse") sind erstellt/ funktionsfähig und ausreichend dokumentiert;
- Projekt-Website wurde ein letztes Mal aktualisiert: Projektergebnisse sind unter Angabe der open source bzw. creative commons Lizenz der Öffentlichkeit zur Verfügung gestellt
- Projektendbericht und Endabrechnung sind abgenommen; abschließender Projektblogeintrag; letzte Förderrate beantragt

4 Umsetzung Förderauflagen

Es wurden keine besonderen Förderauflagen erteilt.

5 Liste Projektergebnisse

1	Projektwischenbericht	CC-BY SA 3.0 AT	https://netidee.at/open-audiosearch https://www.netidee.at/sites/default/files/2021-12/prj5198_call15_ZB_V0_2_0.pdf
2	Projektendbericht	CC-BY SA 3.0 AT	https://netidee.at/open-audiosearch https://www.netidee.at/sites/default/files/2022-08/Endbericht_fin_2.pdf
3	<p>Entwickler_innen-DOKUMENTATION des Projektergebnisses für andere Entwickler_innen ("Dritte"), die das Projektergebnis nach Projektende nutzen/weiterentwickeln wollen</p> <p><u>Für Entwickler_innen (Systemkonzept, ggf. Grobspezifikationen):</u></p> <p>a. WAS IST ES b. FÜR WEN IST ES /WEM HILFT ES WODURCH c. WIE FUNKTIONIERT ES (für Entwickler_innen: Übersicht und detailliertes Systemkonzept, SW-Struktur)</p>	CC-BY SA 3.0 AT	https://netidee.at/open-audiosearch https://www.netidee.at/sites/default/files/2022-08/Developer%20Documentation_0.pdf
4	<p>Anwender_innen-DOKUMENTATION des Projektergebnisses für Anwender_innen, die das Projektergebnis nach Projektende nutzen wollen</p> <p><u>Für Anwender_innen ("Bedienungsanleitung"):</u></p> <p>a. WAS IST ES b. FÜR WEN IST ES /WEM HILFT ES WODURCH c. WIE FUNKTIONIERT ES</p>	CC-BY SA 3.0 AT	https://netidee.at/open-audiosearch https://www.netidee.at/sites/default/files/2022-08/User%20Guide.pdf
5	<p>Veröffentlichungsfähiger Einseiter</p> <ul style="list-style-type: none"> * Kurzfassung WAS FÜR WEN WIE * Liste Projektergebnisse - also diese Liste, ggf. kompromiert * mit Angabe Open Source Lizenz/Webadresse * wo finden Dritte die Projektergebnisse (inkl. Dokumentation Anwender_innen bzw. Entwickler_innen) * mögliche Weiterentwicklungen/ weitere Einsatz-/ Nutzungsmöglichkeiten 	CC-BY SA 3.0 AT	https://netidee.at/open-audiosearch https://www.netidee.at/sites/default/files/2022-08/Endbericht_Kurzfassung_0.pdf
6	<p>Dokumentation Externkommunikation zur Erreichung Sichtbarkeit /Nachhaltigkeit (Teil des Endberichtes)</p> <ul style="list-style-type: none"> * Welche Maßnahmen wurden in welchem Umfang gesetzt * Jeweils Bewertung Aufwand / Nutzen * Lessons Learned / Empfehlungen für andere Projekte 	CC-BY SA 3.0 AT	https://netidee.at/open-audiosearch https://www.netidee.at/sites/default/files/2022-08/Endbericht_fin_2.pdf

7	<i>Development von OAS geht Open Source</i>	AGPL-3.0	https://github.com/openaudiosearch https://netidee.at/open-audiosearch
8	<i>Automatisches Tagging der Keyword-Extraction</i> NLP liefert automatisches Tagging der Keyword-Extraction mit Abgleich zu eigener und externer Knowledge-Base.	AGPL-3.0	https://github.com/openaudiosearch https://netidee.at/open-audiosearch
9	<i>Beginn des Testbetriebs</i> Hardware für Trainings- und Evaluationspipeline ist eingerichtet, ASR-Pipeline ist an den OAS-Core angebunden.	AGPL-3.0	https://demo.openaudiosearch.org https://netidee.at/open-audiosearch
10	<i>Evaluation von ASR-Modell auf eigenen Testdaten</i> Testdaten und Trainingsamples für ASR sind aufbereitet. Evaluationspipeline läuft in OAS integriert.	AGPL-3.0	https://github.com/openaudiosearch https://netidee.at/open-audiosearch
11	<i>Software-Client</i> Python-Client für die OAS-API ist entwickelt, Javascript-Client für die OAS-API ist entwickelt und in OAS-UI (Prototype-Fund) integriert	AGPL-3.0	https://github.com/openaudiosearch https://netidee.at/open-audiosearch
12	<i>Software-Server</i> ASR-Pipelines über OAS Core (AP5) mit Elasticsearch-Integration (Prototype-Fund) in ein Software-Paket integriert, Packaging als Docker-Paket.	AGPL-3.0	https://github.com/openaudiosearch https://netidee.at/open-audiosearch
13	<i>Daten-Projektergebnisse als Open Data</i>	CC-BY SA 3.0 AT	https://github.com/openaudiosearch https://netidee.at/open-audiosearch

6 Verwertung der Projektergebnisse in der Praxis

Die Demo unter <https://demo.openaudiosearch.org> läuft. Der Übergang von Demo- auf Beta-Betrieb ist für das kommende Frühjahr geplant. Im Rahmen eines Projekts mit der European Cultural Foundation ist die Ausweitung von OAS zur Replication-Plattform in Planung. Dazu findet im Mai ein Hackathon in Linz statt, veranstaltet von cba und arso.

7 Öffentlichkeitsarbeit/ Vernetzung

- Projektwebsite: <https://openaudiosearch.org>
- Demo: <https://demo.openaudiosearch.org>

Eine eigene Projektwebseite wurde eingerichtet, ebenso ist eine Demoversion online. Die Veröffentlichung dieser Demoversion wurde über die jeweiligen Social-Media-Kanäle von cba und arso angekündigt. Ebenso wurden die Programmkoordinator*innen der freien Radios Österreichs per Mail-Aussendung informiert. Durch eine Teilnahme an einem NLP-Meetup konnte das Interesse von Entwickler*innen, die in diesem Bereich tätig sind, geweckt werden. Über den auf der Projektseite verlinkten Discord-Server wurden wir auch schon von Interessierten kontaktiert.

Ebenso konnte in persönlichen Gesprächen mit Menschen, die in den Bereichen Podasting, Radio und Softwareentwicklung tätig sind, großes Interesse für Open Audio Search geweckt werden.

Außerdem haben wir Open Audio Search in einen Vortrag und Workshop auf der “Zukunftswerkstatt Community Media” im Herbst 2021 vorgestellt (Link), die Diskussion mit den Teilnehmer*innen - insbesondere Redakteur:innen von Freien Radios - war fruchtbar und gab diverse Impulse für die Weiterentwicklung.

8 Eigene Projektwebsite

- <https://openaudiosearch.org>

9 Geplante Aktivitäten nach netidee-Projektende

- Beta-Betrieb starten
- Ausweitung auf weitere Projektpartner und Entwicklung einer Replication-Engine bis Ende 2022
- Training eines Modells für Österreichisch ist in Planung bzw. Fördermittelakquise
- Ausweitung auf weitere Sprachen ist in Planung
- In den kommenden Wochen wird es ein Release-Event für Interessierte geben, bei dem OAS im Rahmen eines Live-Streams vorgestellt wird.

10 Anregungen für Weiterentwicklungen durch Dritte

Wir sind offen für alle Formen der Zusammenarbeit am Open-Source-Code sowie für darüberhinausgehende Projektzusammenarbeit. Mögliche Themen können sein:

- Anbindung weiterer Content-Plattformen
- Entwicklung anderer Frontends oder Apps