

Pushing transcription work to the next level: Using ASR and LaBB-CAT for linguistic studies

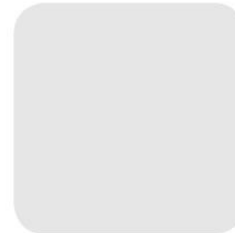


Table of Contents

01 ASR

intro & overview

→ IBM Watson STT via API

→ IBM Watson STT via BAS Web Services

02 LaBB-CAT

intro & overview

→ preparing and uploading files into LaBB-CAT

→ searching, annotating and exporting via LaBB-CAT

03 Discussion

advantages and limitations

04 References

bibliography & summary of useful resources

All resources (including slides) can be retrieved from:



<https://andreas-weilinghoff.com/ASR.html>

01 ASR

What is ASR?

“Automatic speech recognition (ASR) is the process and the related technology for converting the speech signal into its corresponding sequence of words or other linguistic entities by means of algorithms implemented in a device, a computer, or computer clusters.”

(Deng and O’Shaughnessy 2003; Huang et al. 2001 cited in Li et al. 2016)



pushing transcription work to the
next level

- research field for roughly 70 years

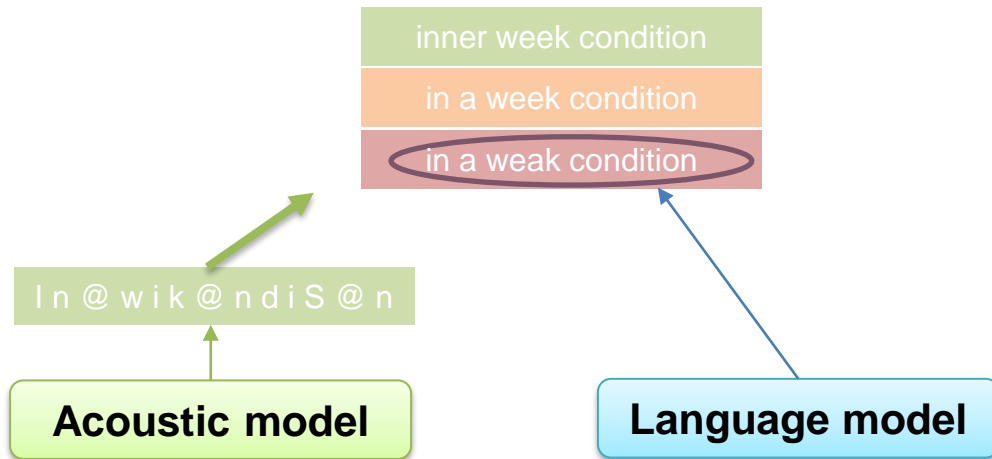
What is ASR?

- great advancements in recent years due to
 - exponential growth of data
 - drastic increase of computing power
 - successful implementation of neural networks

applications: voice search, personal digital assistance systems (PDA), automated captioning, gaming etc.

transcription work ?!

How does ASR work?



Overview of ASR systems by Microsoft Research:



<https://www.youtube.com/watch?v=q67z7PTGRi8>

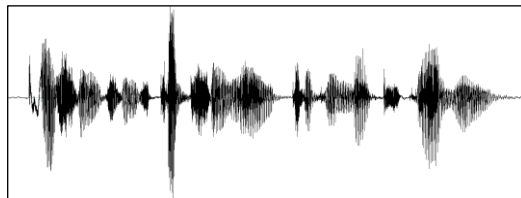
Previous presentation on ASR:



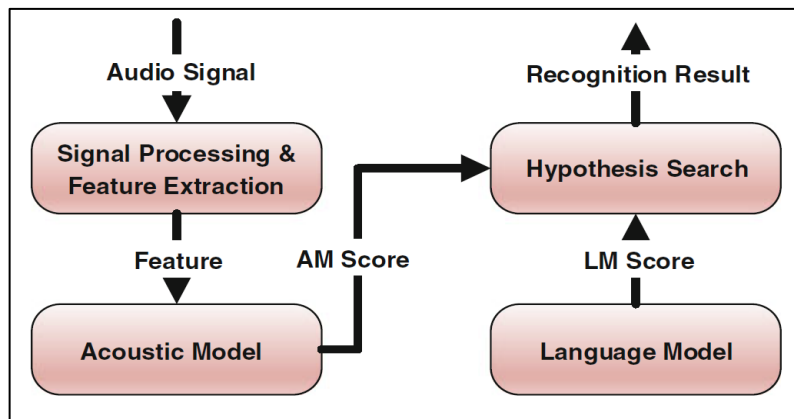
<https://andreas-weilinghoff.com/docs/Seeking%20patterns%20in%20language.pdf>

(Yu and Deng 2015: 4)

How does ASR work?



pushing transcription work to the next level



Overview of ASR systems by Microsoft Research:



<https://www.youtube.com/watch?v=q67z7PTGRi8>

Previous presentation on ASR:



<https://andreas-weilinghoff.com/docs/Seeking%20patterns%20in%20language.pdf>

(Yu and Deng 2015: 4)

ASR Services

Commercial:

→ IBM Watson STT



→ Microsoft Azure STT



→ Google Cloud STT



→ Amazon AWS STT



...

ASR Services

Non-commercial:

- HTK toolkit (University of Cambridge)
- CMU Sphinx toolkit
- Kaldi toolkit



Further platforms/libraries: Common Voice (Mozilla), Tensorflow (Google)

IBM Watson STT

- state of the art ASR service
- long soundfile transcription possible
- includes timestamps and word alignment
- robustness & different language models (e.g AusE, BrE, AmE)
- user-friendly, adaptable and well documented
- free of charge (500 minutes per month for Lite users)

IBM Cloud Watson STT



<https://www.ibm.com/cloud/watson-speech-to-text>

→ Accessible for academic users via WebMAUS interface



IBM Watson STT via API

Prerequisites: You need an [IBM Cloud Account](#) and have [Python](#) installed on your machine. You also need to install the [textgrids](#) library by Tommi Nieminen.

IBM Cloud Watson STT



<https://www.ibm.com/cloud/watson-speech-to-text>

1. Get [Testfile](#) from <https://andreas-weilinghoff.com/ASR.html> and store it on your machine (make sure you can find the directory)
2. Set-up Watson Speech to Text service and get personal API & URL
3. Copy script [Watson_STT_To_Textgrid](#) from <https://andreas-weilinghoff.com/index.html#code> and adapt file directory, API & URL

IBM Watson STT via BAS Web Services

1. Get Testfile from <https://andreas-weilinghoff.com/ASR.html> and store it on your machine (make sure you can find the directory)
2. Go to BAS Web Services and select “ASR“. Log-in with your institutional account or BAS user license and upload Testfile.wav
3. Set up the service options, accept the license agreement
4. Hit “Run Web Service“



Service options	
User email notification	<input type="text"/>
Language	<input type="text" value="English (Great Britain) (emlgoogle/st/watson/webasr)"/>
ASR service	<input type="text" value="IBM Watson ASR"/>
Output format	<input type="text" value="Praat (TextGrid)"/>
Diarization	<input type="text" value="true"/>
Number of speakers	<input type="text" value="0"/>
Speaker label mapping	<input type="text"/>
Exceed quota code	<input type="text"/>

IBM Watson STT via BAS Web Services

Another option for easier correction:

- Transform word level transcription into utterance level transcription with [word_to_utterance_2.0](#) python script from <https://andreas-weilinghoff.com/#code>

02 LaBB-CAT

What is LaBB-CAT?

LaBB-CAT is a **browser-based linguistics research tool**

developed by Robert Fromont and Jen Hay at the New Zealand Institute of Language, Brain and Behaviour (Fromont and Hay 2012)

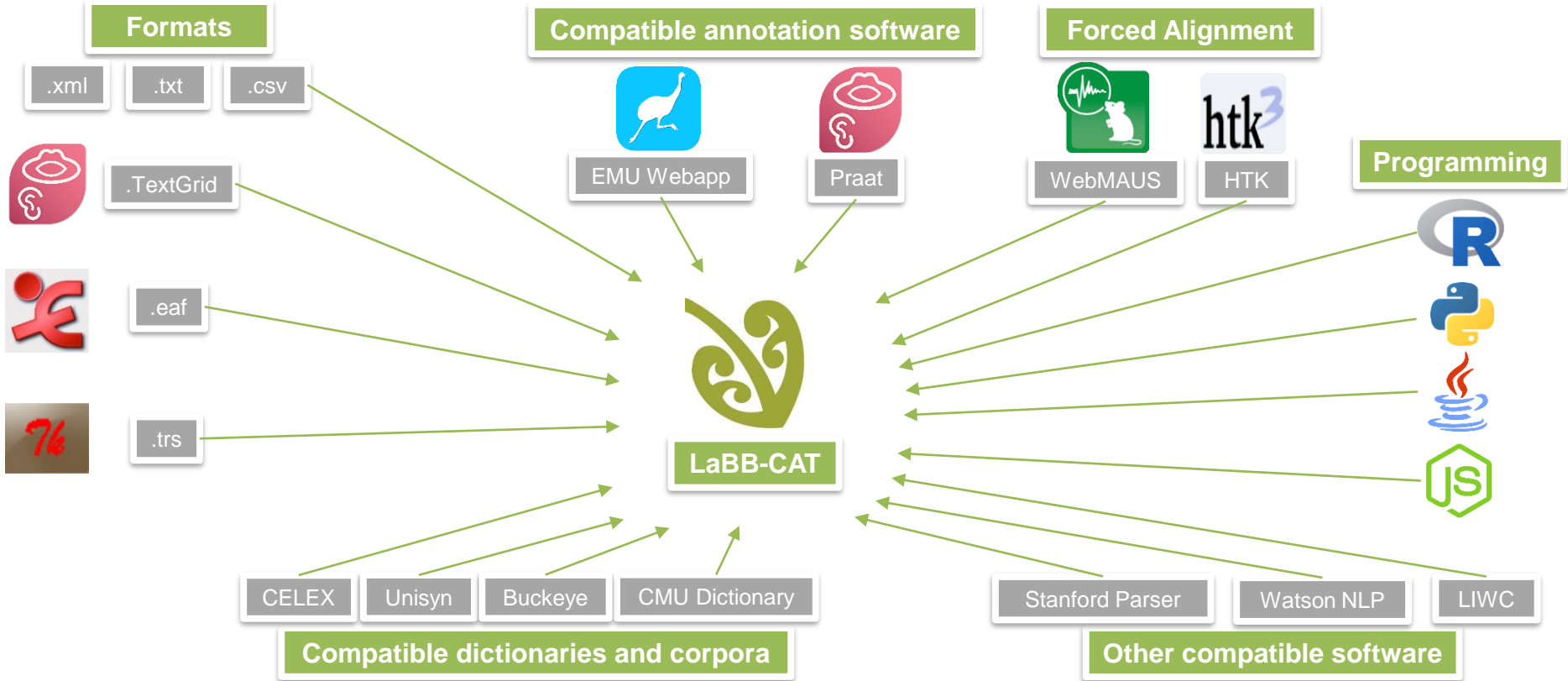
- stores audio/video recordings, transcripts and other annotations of different formats as well as participant background data
- enables automatic and manual annotations on different layers
- includes search function with regular expressions
- compatible with other (linguistic) software

LaBB-CAT



<https://labbcatsystem.canterbury.ac.nz/system/>





LaBB-CAT: Preparing and uploading files

1. Check utterance transcription of [Testfile](#)
Optional: Rename file and tier names for better overview
2. Upload file to LaBB-CAT
3. Adapt participants settings
4. Check out CELEX layer managers

LaBB-CAT



<https://labbcat.canterbury.ac.nz/system/>

LaBB-CAT: Searching, annotating and exporting

1. Check out search function
2. Layer annotation
3. Export functions

LaBB-CAT



<https://labbcatsystem.canterbury.ac.nz/system/>

03 DISCUSSION

ASR: Advantages

- automatic pre-processing of data (great assistance for transcription work)
- time saving
- less tedious workflow
- less “random“ errors → more consistent data preparation

ASR: Limitations

- depending on the research project, manual checking and correcting remains necessary
- Data protection issues for different services
- limited amount of language models
- performance depends on input

ASR Limitations

... the higher the audio quality

... the more structured the speech

... the less regional the speech

... the less speakers involved?

... the better

LaBB-CAT: Advantages

- user-friendly interface
- well documented and great support by authors
- compatibility with different file formats and programs
- search function (→ efficient data preparation and analysis)

LaBB-CAT: Limitations

→ some add-ons (e.g. HTK) are difficult to set up (at least on Windows)

04 REFERENCES

References

Boersma, Paul & Weenink, David (2021). *Praat: doing phonetics by computer* [Computer program]. Version 6.1.42, retrieved 15 April 2021 from <http://www.praat.org/>

Deng, L., O'Shaughnessy, D. (2003). *Speech Processing – A Dynamic and Optimization-Oriented Approach*. CRC Press.

ELAN (Version 6.0) [Computer software]. (2020). Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. Retrieved from <https://archive.mpi.nl/tla/elan>

Fromont, Robert, & Hay, Jennifer. (2012). LaBB-CAT: An annotation store. Proceedings of Australasian Language Technology Association Workshop: 113-117.

HTK (Version 3.4.1) [Software]. (2009). University of Cambridge. Retrieved from <http://htk.eng.cam.ac.uk>

Huang, X., Acero, A., Hon, H.W. (2001). *Spoken Language Processing: A guide to theory, algorithm, and system development*. Prentice hall PTR.

IBM Watson STT [Software]. (2021). IBM Cloud. Retrieved from <https://www.ibm.com/cloud/watson-speech-to-text>

Kisler, T., Reichel U. D. & Schiel, F. (2017): Multilingual processing of speech via web services, *Computer Speech & Language*, 45, 326–347.

References

LaBB-CAT (Version 20210601.1528) [Software]. (2021). NZILBB, University of Canterbury, New Zealand. Retrieved from <http://labbcats.sourceforge.net/>

Ladefoged, P., Johnson. K. (2015). *A Course in Phonetics*. Stamford: Cengage Publishing.

Lamere, P., Kwok, P., Gouv, E. B., Singh, R., Walker, W., Wolf, P. (2003). *The CMU SPHINX-4 speech recognition system*, IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong, 1, pp. 2–5.

Li, J., Deng, L., Haeb-Umbach, R., Gong, Y. (2016). *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Elsevier Publishing.

Microsoft Research. (2017). *Automatic Speech Recognition: An Overview*. [Video]. Youtube. <https://www.youtube.com/watch?v=q67z7PTGRi8>

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K. (2011). *The Kaldi Speech Recognition Toolkit*. IEE Signal Processing Society.

Yu, D., Deng, L. (2015). *Automatic Speech Recognition: A Deep Learning Approach*. London: Springer Publishing.

Summary of useful resources

Speech processing / forced alignment:

BAS Webservice (WebMAUS)

<https://clarin.phonetik.uni-muenchen.de/BASWebServices/>

DARLA

<http://darla.dartmouth.edu/cave>

Montreal Forced Aligner

<https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner>

FAVE Aligner

<https://github.com/JoFrhwld/FAVE>

Summary of useful resources

Toolkits:

HTK

<https://htk.eng.cam.ac.uk>

CMUSphinx

<https://cmusphinx.github.io/>

Kaldi ASR

<https://kaldi-asr.org/>

Summary of useful resources

IBM Watson:

IBM Watson STT

<https://www.ibm.com/cloud/watson-speech-to-text>

Nicolas Renotte

<https://www.nicholasrenotte.com/>

<https://github.com/nicknochnack>

<https://www.youtube.com/channel/UCHXa4OpASJEwrHrLelzw7Yg>