spontaneus speech recognition. Even small amount of training data (6 hrs) can be sufficient to train an experimental ASR model to AID with transcribing the speech materials if texts are missing. Even if the ASR output is not accurate, it saves time compared to if one has to transcribe everything from scratch. Also you can use a spell checker for semi-automatically correcting the output (such as provided by Divvun: https://divvun.no/korrektur/speller-demo.html).

- ***Lacking tools for transcribing, processing data (ASR, force-alignment etc.)*** → These tools are used to aid in automatizing corpus processing for TTS and ASR, by for example automatically finding timestamps from audio matching the texts, thus aligning them with each other. Many open source technologies allow for building the tools with your own language; such as Montreal Forced Aligner (MFA; https://montreal-forced-aligner.readthedocs.io/en/latest/), which can be trained with 1-3 hours of transcribed speech. Also look for possibilities to use tools for related languages: in the case of North Sámi, we have used the Finnish WebMAUS (https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/WebMAUSBasic) model/tool to force-align the materials and automatically find sentence boundaries for our data. It will not work well for finding segment boundaries but for sentence boundaries it is ok. Also look for other useful tools on their website!

- ***Finding voice talents for TTS*** → We recommend to work WITH the language community that the speech technology tools are aimed for! One cannot successfully build speech/language technology for a language in isolation, if we don't know the needs of the community itself. We found voice talents via our native speaker member of our project team. So it's better to have a team member who is part of the language community him/herself OR at least to have good contacts with the community. You MUST give back to the language community and prepare to explain thoroughly what is the purpose and aim of your project.

- ***Poor quality of TTS recordings*** → There are some possibilities to enhance/restore audio quality, one option is VoiceFixer (https://github.com/haoheliu/voicefixer), or to use noise reduction/filtering techniques, for example in Audacity (https://audacityteam.org/home/download/index.html). Be aware of potential artifacts caused by filtering. There are fancier audio plugins to remove echo etc. but these are not free.

For the future:

- By addressing these issues now in a sustainable and smart way, we alleviate these in the future!

- It is important is to stick to open-source technologies – by making our work (models, tools, corpora) openly available for others, and by referring to existing works that we found useful in our project, also sharing these tips hopefully helps!

HANDOUT – our tips for challenges on the way

BUILDING OPEN-SOURCE SPEECH TECHNOLOGY FOR LOW-RESOURCE MINORITY LANGUAGES WITH SÁMI AS AN EXAMPLE – TOOLS, METHODS AND EXPERIMENTS

Katri Hiovain-Asikainen & Sjur Nørstebø Moshagen
UiT The Arctic University of Norway

Some challenges we have faced in our task so far and how did we address them:

- ***Underspecifications/any issues with the orthography*** → convert texts to IPA (many options for doing this), for example. TTS systems can be trained using IPA instead of orthography for better phoneme accuracy but there are some downsides with this...

- ***Finding enough texts for recording the TTS corpus*** → Look from many (public) domains: news, educational, bureaucracy, bible, advertisements... anything you can find. Approximately 75 000 words could suffice for enough speech recordings and thus a TTS voice.

- ***Low amount of audio data for TTS*** → Find out about possibilities to use existing data such as audio books (you can try to ask for permission for copyrighted materials) or even archive materials if the material is good enough. HOWEVER, a 10-hour TTS corpus is doable in less than 1 week, and the quality requirement is nowadays so high that most likely a speech corpus has to be carefully recorded from scratch specifically for this task. You can also look for possibilities to use (Tacotron2) transfer learning between voices or between neighboring languages: https://github.com/NVIDIA/tacotron2/issues/321

- ***Low amount of data for ASR*** → Look for possibilities to use archive materials from national language banks, universities etc., also massive online campaigns such as "Donate your speech" in Finland (https://www.helsinki.fi/en/news/economics/donate-speech-help-artificial-intelligence-understand-dialects) where anyone can donate their speech by speaking to their phone about a topic via an app. Keep in mind that also TTS material can be used to train ASR models – almost any spoken material is usable as long as it is multi-speaker and preferably multi-dialectal. Currently we have collected approx. 34 hours from different sources mentioned above, reaching promising results even for spontaneus speech recognition. Even small amount of training data (6 hrs) can be sufficient to train an experimental ASR model to AID with

transcribing the speech materials if texts are missing. Even if the ASR output is not accurate, it saves time compared to if one has to transcribe everything from scratch. Also you can use a spell checker for semi-automatically correcting the output (such as provided by Divvun: https://divvun.no/korrektur/speller-demo.html).

- ***Lacking tools for transcribing, processing data (ASR, force-alignment etc.)*** → These tools are used to aid in automatizing corpus processing for TTS and ASR, by for example automatically finding timestamps from audio matching the texts, thus aligning them with each other. Many open source technologies allow for building the tools with your own language; such as Montreal Forced Aligner (MFA; https://montreal-forced-aligner.readthedocs.io/en/latest/), which can be trained with 1-3 hours of transcribed speech. Also look for possibilities to use tools for related languages: in the case of North Sámi, we have used the Finnish WebMAUS (https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/WebMAUSBasic) model/tool to force-align the materials and automatically find sentence boundaries for our data. It will not work well for finding segment boundaries but for sentence boundaries it is ok. Also look for other useful tools on their website!

- ***Finding voice talents for TTS*** → We recommend to work WITH the language community that the speech technology tools are aimed for! One cannot successfully build speech/language technology for a language in isolation, if we don't know the needs of the community itself. We found voice talents via our native speaker member of our project team. So it's better to have a team member who is part of the language community him/herself OR at least to have good contacts with the community. You MUST give back to the language community and prepare to explain thoroughly what is the purpose and aim of your project.

- ***Poor quality of TTS recordings*** → There are some possibilities to enhance/restore audio quality, one option is VoiceFixer (https://github.com/haoheliu/voicefixer), or to use noise reduction/filtering techniques, for example in Audacity (https://audacityteam.com/home/download/index.html). Be aware of potential artifacts caused by filtering. There are fancier audio plugins to remove echo etc. but these are not free.

For the future:

- By addressing these issues now in a sustainable and smart way, we alleviate these in the future!

- It is important is to stick to open-source technologies – by making our work (models, tools, corpora) openly available for others, and by referring to existing works that we found useful in our project, also sharing these tips hopefully helps!

HANDOUT – our tips for challenges on the way

BUILDING OPEN-SOURCE SPEECH TECHNOLOGY FOR LOW-RESOURCE MINORITY LANGUAGES WITH SÁMI AS AN EXAMPLE – TOOLS, METHODS AND EXPERIMENTS

Katri Hiovain-Asikainen & Sjur Nørstebø Moshagen
UiT The Arctic University of Norway

Some challenges we have faced in our task so far and how did we address them:

- ***Underspecifications/any issues with the orthography*** → convert texts to IPA (many options for doing this), for example. TTS systems can be trained using IPA instead of orthography for better phoneme accuracy but there are some downsides with this...

- ***Finding enough texts for recording the TTS corpus*** → Look from many (public) domains: news, educational, bureaucracy, bible, advertisements... anything you can find. Approximately 75 000 words could suffice for enough speech recordings and thus a TTS voice.

- ***Low amount of audio data for TTS*** → Find out about possibilities to use existing data such as audio books (you can try to ask for permission for copyrighted materials) or even archive materials if the material is good enough. HOWEVER, a 10-hour TTS corpus is doable in less than 1 week, and the quality requirement is nowadays so high that most likely a speech corpus has to be carefully recorded from scratch specifically for this task. You can also look for possibilities to use (Tacotron2) transfer learning between voices or between neighboring languages: https://github.com/NVIDIA/tacotron2/issues/321

- ***Low amount of data for ASR*** → Look for possibilities to use archive materials from national language banks, universities etc., also massive online campaigns such as "Donate your speech" in Finland (https://www.helsinki.fi/en/news/economics/donate-speech-help-artificial-intelligence-understand-dialects) where anyone can donate their speech by speaking to their phone about a topic via an app. Keep in mind that also TTS material can be used to train ASR models – almost any spoken material is usable as long as it is multi-speaker and preferably multi-dialectal. Currently we have collected approx. 34 hours from different sources mentioned above, reaching promising results even for