Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020), pages 194–201
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Macsen: A Voice Assistant for Speakers of a Lesser Resourced Language

Dewi Bryn Jones
Language Technologies Unit
Bangor University, Wales
d.b.jones@bangor.ac.uk

Abstract

This paper reports on the development of a voice assistant mobile app for speakers of a lesser resourced language – Welsh. An assistant with a smaller set of effective but useful skills is both desirable and urgent for the wider Welsh speaking community. Descriptions of the app's skills, architecture, design decisions and user interface is provided before elaborating on the most recent research and activities in open source speech technology for Welsh. The paper reports on the progress to date on crowdsourcing Welsh speech data in Mozilla Common Voice and of its suitability for training Mozilla's DeepSpeech speech recognition for a voice assistant application according to conventional and transfer learning methods. We demonstrate that with smaller datasets of speech data, transfer learning and a domain specific language model, acceptable speech recognition is achievable that facilitates, as confirmed by beta users, a practical and useful voice assistant for Welsh speakers. We hope that this work informs and serves as a model to researchers and developers in other lesser-resourced linguistic communities and helps bring into being voice assistant apps for their languages.

Keywords: Welsh, voice assistants, Mozilla Common Voice, Mozilla DeepSpeech, speech recognition, transfer learning

1. Introduction

Research and development of language technologies for lesser-spoken languages is characterised by not only a lack of data and human resources but also a lack of useful end user applications that address the technological needs and expectations of speakers as well as better mitigate linguistic digital extinction. In recent years the increasing popularity and capabilities of voice assistants such as Google Assistant, Alexa, Siri and Cortana for larger languages such as English have increased the urgency for language technologies and similar applications to serve speakers of other languages, in particular lesser resourced languages. (Evans, 2018)

This paper reports on the ongoing work for developing a useful voice assistant for Welsh speakers. This work builds on a series of previous short term projects which had crowdsourced Welsh speech corpora (Cooper et. al., 2019) (Prys et. al., 2018(1)) and developed a prototype voice assistant (Jones et. al., 2016) that could run on Raspberry Pis. Our initial prototype assistant was capable of responding to a very limited collection of questions regarding time, weather, news and a small set of article titles from the Welsh language Wikipedia. These 'skills' however were not fully implemented, and therefore not very useful due to the need to devote time and priorities towards speech recognition capabilities, to the neglect of other constituent components such as intent parsing, third party API service integration and generation of natural language answers. Despite attempts to make the Welsh voice assistant's implementation accessible[1] to an audience of developers not expert in language technologies, factors such as the complexity of speech recognition development kits, lack of simple documentation, hardware limitations and licensing restrictions collectively undermined stimulating wider development of voice based applications and services for the Welsh speaking community by other developers.

Fortunately, progress in open source speech technology, machine learning and software development tools has accelerated in recent months and years to provide new opportunities for empowering lesser resourced language communities to develop their own improved voice assistants and other applications. (Jones, 2019)

2. A Welsh Voice Assistant Application

We have developed our own Welsh language voice assistant mobile application and have named it Macsen.

Some existing open source digital assistants' projects and products, such as MyCroft[2], support localization, allowing its software and supported skills to be translated into Welsh. A functionally complete and fully localized assistant is not feasible however if the underlying language technology components, in particular speech recognition and text-to-speech, are not yet as capable as English counterparts. MyCroft also requires specialised hardware, including Raspberry Pi, as a prerequisite for end users to use the software. We had observed from our previous projects that there was limited knowledge and usage of such equipment by the Welsh speaking community at large. Installations of any localized versions would thus be very limited.

Mobile devices however are very prevalent. Developing an assistant for such devices is a very obvious choice of platform for providing voice assistant functionality as easily and as wide as possible to the wider Welsh speaking community.

[1] Our website at https://projectmacsen.github.io/ provided easy to follow information on how to setup and create your own (limited) Welsh voice assistant on Raspberry Pi equipment.

[2] MyCroft – an open source voice assistant: https://mycroft.ai

We designed the Macsen app with the following objectives in mind:

- It must be able to run on Android and iOS phones (and possibly other platforms in the future)
- It must provide complete and useful skills that users will want to use
- Communication should be in as natural as possible language
- Users should be able to easily ascertain what skills and questions the assistant supports
- If the app is not able to recognise the user's speech, then it must
  - still be usable as a text-based (chatbot) assistant
  - provide opportunities for users to contribute to improve its abilities
- It must be easy to add new skills with as few updates as possible
- It must provide freedom to the user and respect privacy
- The entire solution should be open source, be easy to integrate in part or as a whole into other solutions and permissively licensed.
- It must be helpful to the research of language technologies for lesser-resourced languages.

We have succeeded in developing the Macsen app for the two mobile platforms from more or less a single code base by using Flutter[3] by Google, an open source toolkit for building mobile, desktop and web applications. Platform specific code that access underlining OS services or integrate third-party SDK libraries is very limited but include:

- geolocation detection for the weather skill
- creating scheduled notifications for the alarm skill
- Spotify Android/iOS SDK integration[4]

The source code can be found on GitHub[5]. The app is available on Apple AppStore[6] and on Google Play[7].

The skills we anticipated being most useful and popular with end users were:

tywydd (*weather*) – provides the latest weather forecast for today and tomorrow at device's geo-location. Conditions are retrieved from the OpenWeatherMap API.

newyddion (*news*) – reads frontpage or category news headlines from RSS feeds provided by the Golwg360 Welsh language news website.

amser (*time*) – ask the app what time it is. TimeZoneDB API was used so as to get the time for the app's geo coordinates.

larwm (*alarm*) – ask the app to ring an alarm at a particular time later in the next 24 hours.

spotify – ask the app to play music by a particular Welsh musical artist on the device's Spotify app. Most popular artists were selected and whose names are challenging for English language assistants.

wicipedia – the assistant reads the first two sentences from the requested subject's Welsh language Wikipedia article.

For ease of implementation, components providing speech recognition, intent parsing, natural language generation and text-to-speech functionality are provided externally and are accessed by the app over the internet via specially crafted APIs. Separation of the key language technology components to hosted servers provides a more modular and flexible architecture. Such an architecture however might alarm users concerned with privacy. The app therefore provides reassurance with a page that states the privacy policy, explains what information the server uses and assures that no data is retained.

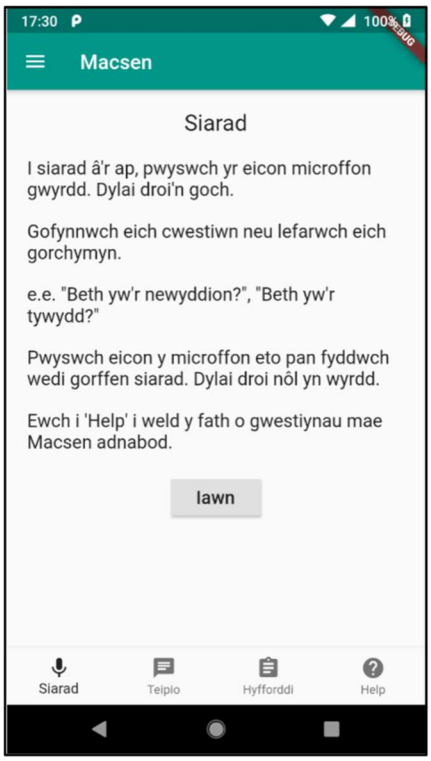

Figure 1 - Screenshot of Macsen upon opening

Figure 1 shows the first screen presented to the user upon opening the app. Four tabs at the bottom provides the user with four ways to interact with the assistant. 'Siarad' *(Speak),* the first and primary tab, allows the user to speak to the app. The 'Teipio' (*Typing*) tab allows users to type in their command or question in case the user's speech has perhaps not been recognized. With the use of predictive text keyboards, frequently typed questions can be quickly learnt and gradually become less cumbersome. The 'Hyfforddiant' (*Training*) tab meanwhile allows the user to contribute recordings for aiding in improving the

[3] Flutter: https://flutter.dev/
[4] Spotify for Developers: https://developer.spotify.com
[5] Source code for the app can be found on GitHub at: https://github.com/techiaith/macsen-flutter
[6] Macsen on Apple AppStore : https://apps.apple.com/gb/app/macsen/id1489915663
[7] Macsen on Google Play : https://play.google.com/store/apps/details?id=cymru.techiaith.flutter.macsen

assistant's speech recognition. Finally the 'Help' tab provides the user with information on the skills and questions the app can respond to. The app also contains a burger menu on the top left, where further information can be found about Macsen, Privacy, server configuration and the Mozilla Common Voice project.

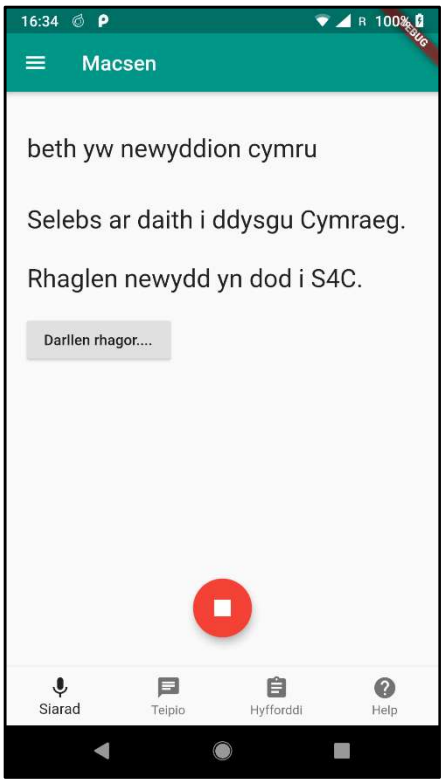

Figure 2- Example result from asking : "What's the news in Wales?"

Similar to other mobile based assistants, such as Siri and Alexa, speech interaction with Macsen is initiated by pressing a microphone button to start listening and then again to stop listening. It does not rely on a wake word which aids ease of implementation and improves privacy. The first opening screen shown in Figure 1 explains the operation of the microphone button and also recommends some initial questions for users to try (e.g. "*What is the news?*"). First impressions are important and so these recommended questions are easier for the speech recognition to recognize and provide very useful and dynamic information in their responses.

Figure 2 shows a screenshot of the app having responded to a more detailed question "What's the news in Wales?" This response is typical of a question and answering type skills (such as time, Wicipedia and weather). The recognised question is displayed first and above the text currently being uttered by the text-to-speech. A button may also be provided for each utterance that activates any hyperlinks to further related content. For example to the associated full news article or Wicipedia page. Displaying the uttered text is useful and probably necessary for the user to comprehend the speech produced from a simple MaryTTS based Welsh voice that may not yet be every time sufficiently naturally sounding and intelligible.

Further example usages are showcased in videos online[8].

As mentioned, the assistant's language technology components are hosted externally to the app and are accessible via cloud based APIs.

Text to speech APIs from previous work in a project on providing online Welsh language text-to-speech services (Lleisiwr, 2019) was available and thus did not require further effort. Speech recognition required newly trained models and are described in subsequent sections.

Particular effort was required to implement intent parsing and the generation of natural language responses. We decided that utilising an existing library for intent parsing would limit development time and increase reusability in other projects and products. A number of chatbot platforms incorporate intent parsing, but we opted for the padatious[9] library developed by the MyCroft project.

Padatious is a very simple and fast neural network intent parser that is able to recognise the intent in a sentence and therefore identify the encompassing skill. Each intent is trained from example sentences generated from templates decorated with lists of associated entities. The entire collection of generated sentences[10] serves other purposes in the construction of our voice assistant, including as will be elaborated later in this paper, language modelling and transcripts for recording.

For example, the intent to trigger playing music by certain artist can be trained by the following template sentences:

```
Chwaraea gerddoriaeth {artist}
Chwaraea fiwsig gan {artist}
...
```

and an associated entities file containing all supported `{artist}` names (one per line):

```
Bryn Fôn
Lleuwen
Y Cyrff
Anhrefn
Sibrydion
Bwncath
Alffa
. . .
```

Especially crafted code in still required to handle each recognized intent and to construct natural language responses from further templates sentences decorated with the results from consuming third party APIs. Some API

[8] https://vimeo.com/showcase/6772051
[9] https://github.com/MycroftAI/padatious

[10] All generated sentences can be obtained from the API at: https://api.techiaith.org/assistant/get_all_skills_intents_sentences

services do not provide their results in Welsh, therefore extra effort was required to translate data items. For example, our weather skill handler uses our localization of the weather conditions[11] such as 'windy', 'raining', 'sunny' returned from the OpenWeatherMap API.

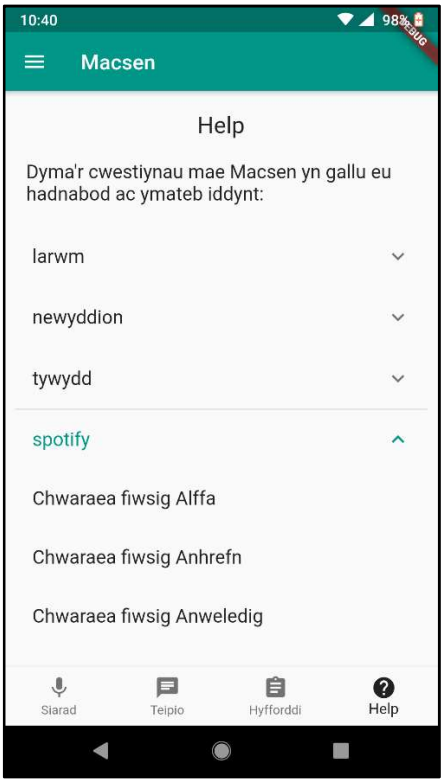

Figure 3- Help tab displaying the sentences Macsen recognizes for each skill

Figure 3 shows the app's 'Help' tab which consists of a collapsible tree like structure populated with the padatious intent parser with the skills and their generated sentences. This helps inform the user of Macsen's skills and of the sentences they can speak.

Finally, figure 4 shows the app's 'Hyfforddiant' (*Training*) tab which consists of a very simple interface for crowdsourcing recordings of Macsen sentences from users. A random sentence is selected from all the sentences generated by the intent parser and is presented for recording. The user uses the same microphone button operation to start and stop recording. Upon stopping the microphone, the recording is uploaded whilst the app presents the next random sentence. Before recording their first sentence, users have been presented with a disclaimer explaining the purpose of recording with assurances that no personal information is collected. Users have to confirm they consent to the sharing their recordings according to open and permissive licensing before recording can begin. Thus users are contributing their voices to an in-domain collection of speech data that, as can be seen in the next section, serves as the test set for evaluating the voice assistant's speech recognition component.

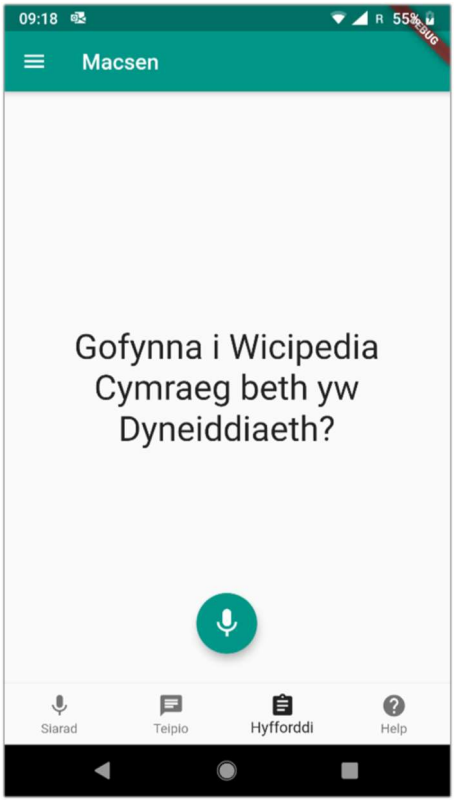

Figure 4- Simple interface for crowdsourcing recordings of Macsen domain sentences from users. "Ask Welsh Wikipedia what is humanism?"

3. Data Collection and Analysis

The data we used for training our assistant's speech recognition engine was sourced from a number of open speech and textual resources.

The primary speech data resource was the Welsh language data from Mozilla's Common Voice multilingual speech corpus (Ardila et. al., 2019). Having made previous attempts at crowdsourcing speech corpora (Cooper S. et. al., 2019), the Welsh community, consisting of our university research unit[12] and members of Welsh open source community[13] were ideally placed to enact Welsh as one of the first languages in the multilingual expansion of Common Voice in June 2018. (Henretty M., 2018).

Table 2 shows how Welsh has progressed through each Common Voice release since June 2018.

In comparison to previous efforts at crowdsourcing Welsh speech data, and relative to some larger languages in Common Voice, 77 hours of speech data for Welsh is a significant achievement and is evidence of the hard work done by Welsh open source volunteers in successfully promoting and attracting the wider Welsh language

[11] Our translations for OpenWeatherMap API weather status can be found at: https://github.com/techiaith/macsen-sgwrsfot/blob/master/online-api/assistant/skills/tywydd/owm/status.cy

[13] http://www.meddal.com

speaking community to contribute their voices to the Mozilla Common Voice project.

	Published	Recordings	Duration (hrs)
CV1	Feb 2, 2019	19403	21
CV2	June 11, 2019	38001	47
CV3	June 24, 2019	38628	48
CV4	Dec 10, 2019	61235	77

Table 1 - Welsh data in Common Voice releases

Each Common Voice language specific release publishes its crowdsourced recordings with transcripts in six datasets. Tables 3 and 4 shows the details of each subset for Welsh in each Common Voice release to date.

	validated (hrs)	invalidated (hrs)	other (hrs)
CV1	21	0.7	0
CV2	40	2	5
CV3	42	2	4
CV4	59	3	15

Table 2 - Datasets contained within Welsh Common Voice data

While the number of validated speech (i.e. volunteers who have listened to a recording and have confirmed it is a valid recording of the transcript) has increased, so too has the number of hours of recordings not yet reviewed. This is an indication of the Welsh speaking community's enthusiasm for recording their voices. Hopefully by the fifth release, after a period of appealing for help in validating recordings, the number of 'other' hours can be brought down.

	train (mins)	dev (mins)	test (mins)
CV1	34	35	37
CV2	37	37	40
CV3	37	36	41
CV4	66	55	59

Table 3- Mozilla prescribed datasets within each Welsh Common Voice release

Table 4 shows the other three datasets. These datasets are recommended by Mozilla as the most suitable as training, development and training sets for creating Mozilla DeepSpeech models. They are very small in comparison to the entire Welsh data size, and have not increased much during 2019. Mozilla's explanation and justification[14] is that DeepSpeech models may exhibit bias towards recognizing sentences that have been recorded multiple times and therefore would not be as optimal as a general purpose speech recognition engine.

Figure 5 shows that the Welsh data in Common Voice contains just over 2000 sentence and therefore a significant proportion have been recorded many tens of times (one sentence having been recorded 69 times) by multiple speakers. During releases CV1, CV2 and CV3 all sentences were being recorded 10, then 20, then 30 times on average. The number of sentences available for recording increased by version CV4, thanks to efforts that began in August 2019 to add significant amounts of new sentences via the Mozilla Sentence Collector website[15]. Sentences were sourced from various public domain collections (Gutenburg, OCRed out of copyright books), donations of portions of copyrighted works and other corpora resources at the disposal of our research unit (Prys et. al., 2016). Since each sentence has to follow in the Mozilla Sentence Collector website a review process to confirm its suitability for reading and training speech recognition models, progress on adding to Common Voice has been slow. We aim to continue our co-ordinated efforts to submit and review sentences at a rate that is at least up with or ahead of the rate of new recordings.

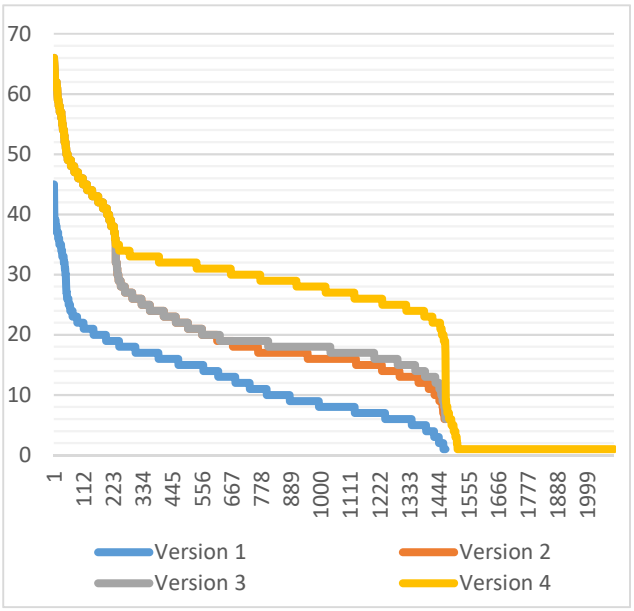

Figure 5- Repeat recordings in each Welsh Common Voice release (validated recordings)
(x – no. of sentences, y – no of recordings)

In the meantime, despite containing repeat recordings, the number of hours provided in Common Voice's validated (and 'other') datasets represent a high number of hours of speech data for a lesser resourced language such as Welsh, which cannot be dismissed and must be evaluated for training a speech recognition engine for a simple voice assistant.

In our speech recognition experiments we formed the following training sets:

SET_1: Mozilla's recommended training set

SET_2: the validated set

SET_3: the validated and other sets combined.

SET_4: Mozilla's recommended training, dev and test set.

The 'Hyfforddiant' (*Training*) from our Macsen voice assistant app, as described in section 2, has provided an

[14] "Why train.tsv includes few files (just 3% of validated set)?": https://discourse.mozilla.org/t/why-train-tsv-includes-a-few-files-just-3-of-validated-set/36471

[15] Mozilla Sentence Collector : https://common-voice.github.io/sentence-collector/#/

additional speech dataset[16] which to date contains 433 recordings, totalling approximately 20 minutes, from 19 beta testers. Each recording has been validated internally and can be used as an in-domain test set. In the presentation of experiment results in section 4, this dataset is denoted by the label 'MACSEN'.

The entire collection of sentences generated for training intent parser training also serves as a text corpus for training a domain specific language model. Consisting of 1033 sentences with 570 unique tokens/words, the language model is denoted in the experiment results in section 4 with the label LM_MACSEN.

We have also trained and used in our experiments a general purpose language model using Welsh texts sourced from the OSCAR multilingual corpus (Suárez et. al., 2019), which is derived from the CommonCrawl[17] corpus. Segments containing any illegal characters such as numbers were excluded and so reduced the training corpus word count from 37 million to 11 million. In the results in section 4, the usage of the OSCAR based language model is denoted by the label LM_OSCAR.

4. Speech Recognition Effectiveness

We decided to evaluate and use Mozilla DeepSpeech (Mozilla) for our assistant's speech recognition engine. Other speech recognition kits, such as Kaldi (Povey, 2011), may also perform sufficiently. Mozilla DeepSpeech however has the advantage of being much easier to use, is very developer friendly and easy to integrate into projects implemented in a wide range of programming languages and technologies.

Mozilla DeepSpeech is based on Tensorflow and is end-to-end neural network architecture which maps audio features directly to characters in words. While the general underlying architecture of DeepSpeech is language independent, the alphabet of possible output characters may be language specific, as is the case with Welsh. The following shows the alphabet used in our experiments:

<space>,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,r,s,t,u,v,w,y,z,á,â,ä,é,ê,ë,î,ï,ô,ö,ô,û,ŵ,ŷ,'

In our experiments we have evaluated the effectiveness of speech recognition for our assistant with two versions of DeepSpeech, each supporting two approaches of machine learning. First the conventional learning approach with the latest release of DeepSpeech (0.6.1[18]). Secondly, a branched work from DeepSpeech 0.5.1 that implements transfer learning[19].

Transfer learning provides a mechanism to exploit models trained on much larger collections of data from a larger language in the training of a new model for new and lesser resourced language. Typically, the bottom layers of a model trained with English language data are kept while a number of top layers are replaced by training with data from a lesser resourced language such as Welsh. Through initial trials and experimentation we found that the optimal number of top layers to drop (our value for the `--drop_source_layers` flag) was 2. The English model provided by Mozilla for DeepSpeech 0.5.1 was used as the source model.

ID	Training	Dev	Test
Training Method : Transfer Learning Language Model : LM_OSCAR			
TL 1	CV1_SET_1	CV1_DEV	CV1_TEST
TL 2	CV3_SET_1	CV3_DEV	CV3_TEST
TL 3	CV4_SET_1	CV4_DEV	CV4_TEST
TL 4	CV4_SET_1	10 epochs	MACSEN
TL 5	CV4_SET_2	10 epochs	MACSEN
TL 6	CV4_SET_3	10 epochs	MACSEN
Training Method : Transfer Learning Language Model : LM_MACSEN			
TL 7	CV4_SET_1	10 epochs	MACSEN
TL 8	CV4_SET_2	10 epochs	MACSEN
TL 9	CV4_SET_3	10 epochs	MACSEN
Training Method : Transfer Learning (with kfold devset) Language Model : LM_MACSEN			
TL 10	CV4_SET_3	MACSEN	MACSEN
TL 11	CV4_SET_4	MACSEN	MACSEN
Training Method : Conventional (no transfer learning) Language Model : LM_MACSEN			
N 1	CV4_SET_1	10 epochs	MACSEN
N 2	CV4_SET_2	10 epochs	MACSEN
N 3	CV4_SET_3	10 epochs	MACSEN
Training Method : Conventional (with kfold devset) Language Model : LM_MACSEN			
N 4	CV4_SET_3	MACSEN	MACSEN
N 5	CV4_SET_4	MACSEN	MACSEN

Table 4 - Experiments for Speech Recognition Evaluation

ID	WER	CER
TL 1	60.54	35.13
TL 2	92.17	68.45
TL 3	62.28	34.23
TL 4	69.16	34.74
TL 5	41.78	17.43
TL 6	41.83	17.84
TL 7	26.01	16.11
TL 8	17.27	9.27
TL 9	15.97	8.42
TL 10 (kfold)	**18.23**	**10.24**
TL 11 (kfold)	25.75	16.71
N 1	100	100
N 2	30.03	15.32
N 3	25.21	13.37
N 4 (kfold)	**25.50**	**13.91**
N 5 (kfold)	96.37	93.47

Table 5 - Speech recognition evaluation results

[16] Macsen test set data is available at http://techiaith.cymru/deepspeech/macsen/datasets/

[17] https://commoncrawl.org

[18] Mozilla DeepSpeech 0.6.1 release : https://github.com/mozilla/DeepSpeech/releases/tag/v0.6.1

[19] Mozilla DeepSpeech transfer learning forked branch : https://github.com/mozilla/DeepSpeech/tree/transfer-learning2

Default values for flags affecting learning rate and dimensions were used in all experiments. The results are shown in Table 6. After initial experimenting with Mozilla's recommended train, dev and train (first three rows in Table 6) we observed that 10 epochs would suffice for experiments where a development set was not possible or appropriate.

Our training setup consisted of a single workstation containing two NVIDIA GTX 1080Ti graphics cards operated by our own crafted scripts[20] that made full use of Dockerfiles provided in each Mozilla DeepSpeech release. Training times ranged from a couple of minutes for SET_1 based experiments to approximately an hour for SET_3.

In Table 6, the k-fold cross validation (k=10) evaluation method (shown in bold), a commonly used resampling procedure to evaluate with a limited data set, was used as a means for reliably confirming the optimal model training configuration for each approach. Thus, the best WER scores are achieved by utilising as much speech data as possible (even if not yet validated by Common Voice volunteers) with transfer learning machine learning approach and a domain specific language model.

We learn from contrasting experiment N1 (WER 100%) with its corresponding transfer learning experiment – TL7 (WER 26.01%) in table 6 that transfer learning with only one hour of training speech data (CV4_SET_1) can provide an immediate and drastic reduction in WER. When we utilise all of Mozilla's recommended data sets of non-repeated recordings (CV4_SET_4, approximately 3 hours) and contrast results between experiments N5 and TL11, we observe there is no significant reduction in WER from the transfer learning method – 0.26% - compared to a 3.63% reduction achieved in the WER for conventional machine learning approach. Additional hours of training data brings down the conventional learning method's WER to a best score of 25.21% in N3. The same amount of data however brings the transfer learning method in TL9 to its lowest WER at 15.97%.

We can also observe also from these results that a domain specific language model aids significantly in reducing the WER. Experiment TL6 shows that running a DeepSpeech decoder, trained with transfer learning and with an OSCAR based language model only achieves a WER of 41.83% for the Macsen voice assistant domain. When using instead a domain specific language model, as in TL9, a WER of 15.97% is achieved.

These results reinforce that for a lesser resourced language, transfer learning and domain specific language models provide the best and only feasible means at present to achieve effective speech recognition capability for voice assistants.

5. Conclusion

This paper has reported on a project to develop a Welsh language voice assistant app for Apple iOS and Android mobile devices as well present how Mozilla's Common Voice and DeepSpeech projects were exploited to provide an effective speech recognition engine component.

Our speech recognition engine still has a WER above 10 whereas a score of below 10 – as is regularly reported for engines for larger languages – may be considered as a prerequisite. We believe however that our speech recognition engine is sufficiently practical and effective in a voice assistant application setting. As the output from speech recognition in a voice assistant provides the input to the intent parser, the intent parser's tolerance and flexibility on sentence variations may mitigate any issues arising from less than perfect recognition results.

Initial user feedback has informed us that the Macsen voice assistant app is able to recognize and respond to nearly all of their questions and commands. The Spotify skill is particularly popular. Users also commented that they routinely ask for and obtain the latest news and weather from Macsen.

We aim to continue analysing and improving the Welsh data in Mozilla Common Voice in collaboration with Mozilla and the Welsh open source volunteer community. Analysis in this paper has highlighted the scale of repeated recordings and so an immediate task for further training of Welsh speech recognition is to understand and mitigate any bias.

We also aim to continue working on Macsen by adding more skills and making it more useful as a resource to the developer community. A particular ambition is for Macsen to be useful for researchers and developers in other lesser resourced language communities to bootstrap voice assistants for their languages. In this regard, the app's user interface is easily localizable with Flutter's i18n support. Crucially however, this evaluation of Mozilla's DeepSpeech suggests that, even with as little as an hour of speech data from the Mozilla Common Voice project, utilising a transfer learning approach to training, along with possibly a reduced number of skills, and therefore a smaller language model and possibly a smaller alphabet, an effective and reliable speech recognition component for a voice assistant is achievable and can enable the more immediate availability of voice assistant apps for other lesser resourced languages.

9. Acknowledgements

We thank all the volunteers who have recorded and contributed their voices to the Welsh Common Voice effort. We in particular thank Rhoslyn Prys who has given a great deal of his time and energy in a volunteer capacity to initially translate the Common Voice website to Welsh and subsequently to promoting the project relentlessly. Recruitment campaigns have been supported by the Welsh Government, Welsh Language Commissioner, large public organisations in Wales as well as coverage by local media.

We thank also Mozilla for the opportunities it has provided via its Common Voice and DeepSpeech projects, as open and decentralized speech technologies, to empower lesser

[20] Our scripts and information for reproducing our experiments can be found on GitHub at https://github.com/techiaith/docker-deepspeech-cy

resourced language communities to develop their own applications such as voice assistants.

The work on developing the Macsen app was made possible with financial support from the Welsh government.

6. Bibliographical References

Cooper,S. Jones, D.B. and Prys, D. (2019) Crowdsourcing the Paldaruo Speech Corpus of Welsh for Speech Technology. *Information,* 10(8), p.247. Available at: http://dx.doi.org/10.3390/info10080247

Evans, J. (2018) Report on language equality in the digital age. *European Parliament Report 2018/2028 INI*. Available at http://www.europarl.europa.eu/doceo/document/A-8-2018-0228_EN.pdf

Henretty, M. (2018) More Common Voices. https://medium.com/mozilla-open-innovation/more-common-voices-24a80c879944 [Accessed Feb 5, 2020]

Jones, A. (2019) Voice recognition project offers big opportunity to Welsh language. https://nation.cymru/news/voice-recognition-project-offers-big-opportunity-to-welsh-language/ [Accessed Feb 5, 2020]

Jones,D.B. Cooper,S. (2016) Building Intelligent Digitial Assistants for Speakers of a Lesser-Resourced Language. *Proceedings of LREC 2016 Workshop "CCURL 2016 – Towards an Alliance for Digital Language Diversity" p.74.* Claudia Soria et. al. (eds). Available at: http://www.lrec-conf.org/proceedings/lrec2016/workshops/LREC2016 Workshop-CCURL2016_Proceedings.pdf

Lleisiwr (2019) Synthetic voices for patients that are about to loose their ability to speak as a result of diseases such as Motor Neurone Disease or throat cancer. Available at: https://lleisiwr.techiaith.cymru/?lang=en [Accessed: Feb 5, 2020]

Mozilla (n.d.). A TensorFlow implementation of Baidu's DeepSpeech architecture. Available at https://github.com/mozilla/DeepSpeech [Accessed: Feb 5, 2020].

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glem, Qian,Y., Schwarz, P., Silovský, J., Stemmer, G., and Veselý,K. (2011). The Kaldi speech recognition toolkit. *In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. Waikoloa, HI, USA

Prys, D., Prys, G., Jones, D.B. (2016). Cysill Ar-lein Corpus: A corpus of written contemporary Welsh compiled from an online spelling and grammar checker. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). Protoroz, Slovenia*

Prys,D. Jones, D.B. (2018(1)). Gathering Data for Speech Technology in the Welsh Language: A Case Study. *Proceedings of the LREC 2018 Workshop "CCURL 2018 – Sustaining Knowledge Diversity in the Digital Age",* p.*56.* Claudia Soria, Laurent Besacier and Laurette Pretorius (eds.). Available at: http://lrec-conf.org/workshops/lrec2018/W26/pdf/book_of_proceedings.pdf

Prys, D., Jones D.B. (2018(2)) National Language Technologies Portals for LRLs: A Case Study. *In: Vetulani Z., Mariani J., Kubis M. (eds) Human Language Technology. Challenges for Computer Science and Linguistics*. LTC 2015. Lecture Notes in Computer Science, vol 10930. Springer, Cham

7. Language Resource References

Ardila R., Branson M., Davis K., Henretty M., Kohler M., Meyer J., Morais R., Saunders L., Tyers F., Weber G. (2019) Common Voice: A Massively-Multilingual Speech Corpus. *arXiv:1912.06670v1 [cs.CL]*

Suárez, P.J.O., Sagot,B., Romary, L. (2019) Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7),* Jul 2019, Cardiff, United Kingdom. (hal-02148693)