Building an ASR System for a Low-research Language Through the Adaptation of a High-resource Language ASR System: Preliminary Results

Odette Scharenborg [1], Francesco Ciannella [2], Shruti Palaskar [2], Alan Black [2], Florian Metze [2], Lucas Ondel [3], Mark Hasegawa-Johnson [4]

[1] Centre for Language Studies, Radboud University, Nijmegen, The Netherlands
[2] Carnegie Mellon University, Pittsburgh, PA, USA
[3] Brno University of Technology, Brno, Czech Republic
[4] University of Illinois at Urbana-Champaign, Champaign, IL, USA
o.scharenborg@let.ru.nl

Abstract

For many languages in the world, not enough (annotated) speech data is available to train an ASR system. We here propose a new three-step method to build an ASR system for such a low-resource language, and test four measures to improve the system's success. In the first step, we build a phone recognition system on a high-resource language. In the second step, missing low-resource language acoustic units are created through extrapolation from acoustic units present in the high-resource language. In the third step, iteratively, the adapted model is used to create a phone transcription of the low-resource language, after which the model is retrained using the resulting self-labelled phone sequences to improve the acoustic phone units of the low-resource language. Four measures are investigated to determine which self-labelled transcriptions are 'good enough' to retrain the adaptation model, and improve the quality of the phone speech tokens and subsequent phone transcriptions: TTS and decoding accuracy to capture acoustic information, a translation retrieval task to capture semantic information, and a combination of these three. The results showed that in order to train acoustic units using self-labelled data, training utterances are preferably needed that capture multiple aspects of the speech signal.

Index Terms: Low-resource language, Automatic speech recognition, Adaptation, Linguistic knowledge

1. Introduction

Automatic speech recognition technologies require a large amount of annotated data for a system to work reasonably well. However, for many languages in the world, not enough speech data is available, or these lack the annotations needed to train an ASR system. In fact, it is estimated that for only about 1% of the world languages the minimum amount of data that is needed to train an ASR is available [1]. In order to build an ASR system for such a low-resource language, one cannot simply use a system trained for a different, even if related, language, as cross-language ASR typically performs quite poorly [2]. Different languages have different phone inventories, and even phones transcribed with the same IPA symbol are produced slightly differently in different languages [3].

Recently, different approaches have been proposed to build ASR systems for such low-resource languages. One strand of research focuses on discovering the linguistic units of the low-resource language from the raw speech data, while assuming no other information about the language is available, and using these to build ASR systems (the Zero-resource approach) [4]-[15]. Another strand of research focuses on building ASR systems using speech data from multiple languages, thus trying to create universal or cross-linguistic ASR systems [16]-[19].

However, most of the world's languages have been investigated by field linguists, meaning that some information about the language typically is available. We here propose a method to adapt an ASR system for a high-resource language using linguistic information of the low-resource language to build an ASR system for that low-resource language. In addition to some unlabelled speech data (in line with the Zero-resource approach), we assume that a 'description' of the phone(me) inventory of the language is available, e.g., obtained from a field linguist. A second assumption is that enough annotated speech material of a related high-resource language is available to build an ASR system for that related high-resource language. Note, however, that the here-proposed system does not rely on having a high-resource related language; in principle, the approach presented here could work for any language pair. Experiments in cross-language ASR adaptation tend to report that adaptation between related languages is more successful than adaptation among unrelated languages [17], though many other factors seem to be equally important, including similarity of the speaker voices and recording conditions of the two speech corpora [18].

Because different languages have different phone inventories, whichever high-resource language we choose, some of the phones from the low-resource language will not be present in the high-resource language. For instance, when comparing Dutch and English, English has, e.g., the /æ/ (as in *f**a**nt**a**stic*) and /θ/ (as in ***th**ree*) which are lacking from Dutch. So, in order to build an ASR system for a low-resource language, first the acoustic phone tokens of the low-resource language need to be discovered. We propose a three-step method: (1) Build a phone recognition system on a high-resource language, in our case Dutch. (2) The phone inventory of the ASR system trained on the high-resource language is remapped or 'transferred' to the phone inventory of the low-resource language. For those phones from the low-resource language that are not present in the high-resource language, acoustic units need to be created in the high-resource language ASR system. A 'baseline' or starting point for the missing acoustic unit of the low-resource language is then created by extrapolating between acoustic units that are present in the high-resource language. In our case, we chose to use English as a low-resource language for reasons which will become clearer below. (3) The adapted model will iteratively be used to create a phone transcription of the low-resource language, after which the model will be retrained using the resulting self-labelled phone sequences in order to improve the acoustic phone units of the low-resource language.

The phone transcriptions created by the adapted models will contain errors which will have repercussions on the quality of the acoustic phone units. The main question for this paper is therefore whether selecting only those self-labelled transcriptions that are ‘good enough’ to retrain the adaptation model will improve the quality of the acoustic phone units and subsequent phone transcriptions. These acoustic phone units should on the one hand capture the acoustic information of that phone correctly, and on the other hand, they should be able to capture semantic information correctly. Four criteria were investigated: ASR score, text-to-speech (TTS) synthesis score, translation text retrieval score, and a fusion of the three.

2. Methodology

A baseline system was trained on Dutch, adapted to English, and then applied for the transcription of English utterances. The baseline was then compared to self-trained systems created using the four different criteria for determining which utterances to use in self-training. The different criteria capture different attributes of the speech, therefore we expect them to be complementary. ASR confidence scores measure the degree to which the transcription is a good match to the audio signal (relative to the model); in a sense, this is a measure of the phonetic quality of the transcription or the degree to which the transcription captures linguistically salient attributes of speech. TTS also measures phonetic quality, but with different models. TTS attempts to measure the adequacy of the transcription to capture all information that a human listener would hear. Translated text retrieval measures the degree to which the transcription is sufficient to communicate the meaning of the sentence. The experiments were run at the Pittsburgh Supercomputing Center (PSC; [20],[21]).

2.1. Speech materials

The Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN, [22]) is a corpus of almost 9M words of Dutch spoken in the Netherlands and in Flanders (Belgium), in over 14 different speech styles, ranging from formal to informal. For the experiments reported here, we only used the read speech material from the Netherlands, which amounts to 551,624 words for a total duration of approximately 64 hours of speech.

The English data came from the FlickR_8K corpus [23],[6] which contains 5 different natural language text captions for each of 8000 images captured from the FlickR photo sharing website which were read aloud by crowdsource workers from Amazon Mechanical Turk. Additionally, within the context of the Frederick Jelinek Speech and Language Technology (JSALT) workshop 2017, tokenised translations into Japanese were obtained for each of the 40,000 captions. Moreover, forced alignments for FlickR_8K were created using a DNN/HMM hybrid system using 8,000 CD states and logMELs as acoustic input features trained on data as described in [24].

To mimic a low-resource language we randomly selected 3660 utterances from the FlickR_8K training set. The duration of these utterances corresponded to approximately 4 hours of speech (which corresponds to the number of hours of speech material for an actual low-resource language, Mboshi [1]).

2.2. Proposed systems: Baseline and self-trained

Figure 1 shows an overview of the proposed adaptation system. First, a *Baseline* DNN is trained on the Dutch CGN. Next, the soft-max layer of the DNN is adapted from the Dutch to the English phone set (see Section 2.3): the *Adapted* model. Subsequently, the adapted soft-max layer is used to decode the English speech material using a free phone recognition pass. The projection and soft-max layers are then retrained with (1) all self-labelled utterances, or (2) only with those self-labelled utterances that have the best scores according to the four selection criteria. Two decoding and retraining iterations are carried out, yielding different *Self-trained* models.

All models are tested on our train-test set of 3660 utterances from FlickR_8K. The accuracy of the output phone sequences of the different models is evaluated by comparing them to the gold standard as created by the forced alignment (see Section 2.1) by calculating the edit-distance, and is reported as percentage Token Error Rate (%TER) where “tokens” here are phones, phone-like units, and acoustically discovered units.

2.2.1. Baseline model

The baseline model used for the experiments is trained using Connectionist Temporal Classification (CTC; [25]), implemented using Eesen [26]. The CTC paradigm uses a Recurrent Neural Network (RNN), trained using an error metric that compares the reference and hypothesis symbol sequences with no regard to the time alignment of symbols. The CTC-RNN models the mapping between the speech signal and the output labels without the need for an explicit segmentation of the speech signal into output labels (typically obtained using a forced-alignment), and models all aspects of the sequence within a single network architecture by interpreting the network outputs as a probability distribution over all possible label sequences, conditioned on a given input sequence.

The baseline RNN uses a six layer bidirectional LSTM Recurrent Neural Network. Each LSTM layer has 140 LSTM cells, and LSTM layers are connected using 80-dimensional projection layers. There is also an 80-dimensional projection layer at the input of the LSTM, which reduces the dimensionality of the input features, which consists of 3 stacked frames (at 10ms distance) of 40-dimensional FBank features. The network step size is 30 ms. The final LSTM outputs are connected to another 80-dimensional projection layer which is connected to the phone soft-max layer. The size of the soft-max layer depends on the phone set of the language; see Section 2.3. The network has been trained with Stochastic Gradient Descent for 20 epochs, using phones as targets.

We apply greedy decoding and thus take the output of the CTC network (consisting of a probability distribution over the phone set) and at every frame select the phone with the highest probability. Sequences of adjacent outputs with the same value are clustered into the same phone, and blanks (used by CTC to fill the distance between phonetic detections) are discarded. Note that all phones have an equal prior probability.

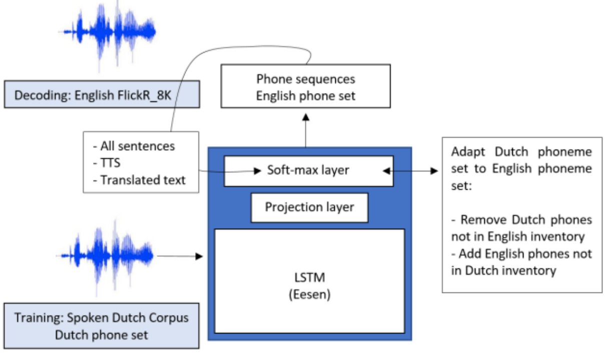
Decoding: English FlickR_8K
Phone sequences English phone set
- All sentences
- TTS
- Translated text
Soft-max layer
Projection layer
LSTM (Eesen)
Adapt Dutch phoneme set to English phoneme set:
- Remove Dutch phones not in English inventory
- Add English phones not in Dutch inventory
Training: Spoken Dutch Corpus Dutch phone set

Figure 1. *Overview of the proposed adaptation system.*

Table 1. *Mapping of the English (L2) phone not present in the Dutch phoneme inventory, with an example of the sound (indicated with bold) in an English word.*

Missing L2 phone	Example	Mapping		
		L1:1	L1:2	L1:3
æ	m**a**p	ɛ	a	ɛ
ʌ	c**u**t	ɛ	ɑ	a
ð	**th**ey	v	z	v
ɝ	b**ir**d	ø	o	ø
θ	**th**ree	f	s	f
ʊ	b**oo**k	ɪ	u	ɪ

2.2.2. Best scoring utterances according to ASR score

The best scoring ASR utterances are those phone sequences that have the lowest TER on the train-test set. The number of selected phone sequences was identical to the number of phone sequences obtained from the translation retrieval measure (see Section 2.2.4).

2.2.3. Best scoring utterances according to TTS score

The TTS system used is Clustergen [27]. TTS typically consists of four stages. First, text is converted to a graph of symbolic phonetic descriptors. This step is omitted in our case, as the output of the *Adaptation* model already consists of a sequence of phones. Second, the duration of each unit in the phonetic graph is predicted. Third, every frame in the training database is viewed as an independent exemplar of a mapping from discrete inputs to continuous outputs, and a machine learning algorithm (e.g., regression trees [27] and random forests [14]) is applied to learn the mapping. Discrete inputs include standard speech synthesis predictors such as the phone sequence and prosodic context, as well as variables uniquely available to Clustergen such as the timing of the predicted frame with respect to segment boundaries at every prosodic level. Continuous outputs include the excitation and pitch, the mel-cepstrum [29], and a representation of the dynamic trajectory of the mel-cepstrum (its local slope and curvature); mel-cepstrum of the continuous signal is then synthesized using trajectory overlap-and-add. Clustergen works well with small corpora because it treats each frame of the training corpus as a training example, rather than each segment. This makes it suitable for our low-resource scenario.

Synthesized speech can be compared to a reference speech signal using mean cepstral distortion (MCD, [27]). MCD measures the average distance between the log-spectra of the synthetic and natural utterances. MCD has been demonstrated to be an extremely sensitive measure of the perceived naturalness of speech utterances, e.g., an MCD difference between two synthesis algorithms of 0.3 (on the same test corpus) is usually perceptible by human listeners as a significant difference in perceived naturalness [27].

In the experiments reported here, MCD measured the difference between synthetic and reference speech signals. Low MCD suggests that the ASR generated a pretty reasonable transcription of the utterance. MCD of the re-synthesis was therefore used as the second of our selection criteria.

2.2.4. Best scoring utterances according to translation retrieval

The translated text retrieval system is trained to retrieve the ID of the Japanese translated text from a database of Japanese translated texts which corresponds to the English transcription of the spoken utterance that is presented at its input. The Japanese translated texts database consists of all 3660 training-test utterances. The retrieval system is based on the image retrieval system by [6] but instead of matching images, we are matching phone sequences to translated texts. The retrieval system is implemented in the xnmt sequence-to-sequence neural machine translation architecture [30] using the DyNet neural network library [31]. The source and target encoders both consist of an input layer, LSTM hidden layer, and an output layer, each containing 512 nodes. The embeddings output by the encoders are fed into the retriever which calculates the dot-product of the two encoders and takes the smallest as the best match. The source encoder takes as input the phone sequences output by the soft-max layer of the CTC-DNN (see Section 2.2.1). The target encoder takes as input the IDs of the translated text utterances. Adam-training with a learning rate of 0.001 is used to map the phone sequences to the IDs of the Japanese translations. Training runs for 100 epochs.

After training, the 3660 training utterances are run through the retrieval system again in order to retrieve the phone sequences that score best on the retrieval task. These are those utterances for which the correct ID of the translated text appears in the top N=10 of answers.

2.2.5. Fusion of scores

System combination, of systems with complementary error patterns, often yields a combination system whose TER is lower than the TER of any component system [32]. Since translation, TTS, and ASR all capture different aspects of the speech signal, we expect that the TER of the combination system should be lower than the TERs of any component system. We therefore retrain the *Adaptation* model with those phone sequences that capture the acoustic and the semantic information best, i.e., we select those N utterances that appear in at least two of the three best utterances lists (giving preference to the combination of translation retrieval + TTS or ASR), where N is equal to the number of utterances for the other measures.

2.3. Adaptation of the soft-max layer

The number of different Dutch phones in CGN is 42, while the English FlickR_8K has 45 different phones. There are three reasons for the difference between the phone sets. (1) Nine English phones are diphthongs or affricates which do not exist in Dutch, but which can easily be constructed from a sequence of two Dutch phones. These nine English phones are not represented in the soft-max layer but dealt with in a post-processing step. (2) Eleven Dutch phones are not present in English and these are removed from the soft-max layer. (3) Six English phones do not exist in Dutch (referred to as missing L2 phones) and need to be added to the soft-max layer. Vectors in the soft-max layer are created for these missing L2 phones on the basis of the trained Dutch (L1) phones; the created soft-max nodes are then adapted using the speech data selected according to the selection criteria described in Section 2.2.

The desired English-language phones are initialized by linearly extrapolating the missing L2 (English) node in the soft-max layer from existing vectors for the Dutch L1 phones using:

$$\vec{V}_{|\varphi|,L2} = \vec{V}_{|\varphi|,L1:1} + 0.5\,(\vec{V}_{|\varphi|,L1:2} - \vec{V}_{|\varphi|,L1:3}) \qquad (1)$$

where $\vec{V}_{|\varphi|,L2}$ is the vector of the missing L2 phone φ,L2 that needs to be created, $\vec{V}_{|\varphi|,L1:x}$ are the vectors of the Dutch

L1 phones φ,L1:x in the soft-max layer that are used to create the vector for the missing English phone φ,L2. Among the three Dutch phones, L1:1 refers to the phone which is used as the starting point from which to extrapolate the missing L2 phone, and L1:2 and L1:3 refer to the L1 phones whose displacement is used as an approximation of the displacement between the Dutch L1 vector and the L2 phone that should be created. Table 1 lists the six missing L2 phones, and the Dutch L1 phones that are used to create the vectors for the missing English L2 phones.

3. Results

The TER of the *Adaptation* model, i.e., the *Baseline* model for which the soft-max layer had been adapted to the English phone set but not yet retrained, is 72.59%. Table 2 shows the TER results for the *Self-trained* models, i.e., the models after retraining. Iteration 1 refers to the models for which the projection layer and soft-max layer have been retrained with the best scoring self-labelled phone sequences according to the ASR, TTS, translation retrieval system, or the combination of these. Iteration 2 refers to the models for which the projection layer and soft-max layer have been retrained with the best-scoring (according to the DNN, TTS, and retrieval task) self-labelled utterances of the corresponding models after Iteration 1. The number of phone sequences used for retraining was 2468 (=67.43% Recall@10 on the translation retrieval task) for iteration 1 and 2101 (=57.40% R@10) for iteration 2.

Formal statistical significance tests have not yet been performed for these data, but an overly conservative model can be defined: if we assume that token errors within a speech file are 100% correlated, and follow a Bernoulli model [33], then two ASR systems are significantly different if their TERs differ by at least 50%/sqrt(3660)=0.83%. By this overly conservative standard, 3 of the 5 systems at Iteration 1 and the fusion system at Iteration 2 are significantly better than the baseline (see bold numbers in Table 2), and there is no significant difference among these different methods of selecting self-labelled utterances. Except for the fusion system, the systems in the second iteration, however all performed worse than the Iteration 1 models, occasionally even worse than the baseline model.

4. Discussion and conclusions

We presented a three-step method to build an ASR system for a low-resource language through the adaptation of an ASR system of a high-resource language, using a combination of linguistic knowledge and semi-supervised learning. Crucially, phones that are present in the low-resource language but not in the high-resource language need to be created. This is done through a linear extrapolation between existing acoustic units in the soft-max layer after which the acoustic units are iteratively retrained using all utterances or only those utterances that have the best score according to four different criteria: ASR score, TTS score, translated text retrieval score, and their combination.

The baseline TER is comparable to the phone error rates of cross-language ASR systems (e.g., [2] reports PER ranging from 59.83% to 87.81% for 6 test languages). Re-training the system, using a self-labelling approach with confidence scoring, can significantly improve TER after the first iteration (see Table 2). The differences between the different approaches are however small, a more sensitive statistical test might demonstrate significance of some of the differences in Table 2. Retraining the systems for a second pass, however, decreased performance for all measures but the fusion system, even surpassing the *Baseline*'s performance for some measures.

Table 2. *Token error rates (TER%) on the 3660 FlickR_8K train-test utterances for the different self-trained models. Bold indicates significantly better performance than Baseline.*

Selection criterion	**Iteration 1** (2468 utts)	**Iteration 2** (2101 utts)
All sentences	71.80	72.56
ASR	**71.67**	72.42
TTS	**71.71**	72.52
Translation retrieval	71.83	72.88
Fusion	**71.76**	**71.72**

The projection and soft-max layers were retrained using only the best scoring phone sequences according to four different criteria. However, since neural networks are extremely data hungry, the (in principle) improved quality of the training utterances at Iteration 2 did not outweigh the substantial decrease in training data from Iteration 1 to Iteration 2. The system retrained on all sentences, on the other hand, might have suffered from the presence of a couple of bad transcriptions. Only the fusion model's performance did not decrease from Iteration 1 to Iteration 2. This is likely due to the training utterances of this system capturing *both* phonetic and semantic information well. Thus, in order to train acoustic units using self-labelled data, training utterances are needed that capture multiple aspects of the speech signal.

Instead of discarding the bad data, future work will investigate the use of data augmentation methods to increase the importance of the good data. [33] demonstrated that ASR could be improved by making "perturbed" copies of each of the input waveforms, thus increasing the size of the training dataset. Perturbations include pitch shifting, speeding up, slowing down, or adding certain types of noise at different SNRs. The best-scoring phone sequences would then receive a duplication factor that is larger than those phone sequences which have a lower score. Relatedly, this would allow us to refine the fusion method by not using the majority vote but rather use the intersection of the three measures. Moreover, the current retraining only updates the projection and soft-max layers, because of the fairly low amount of (re)training data that is available. However, future work will investigate the effect of retraining the whole LSTM, or also introduce projections in the temporal dimension, or update only specific LSTM layers.

Although the aim of the paper is to investigate the possibility to build an ASR system for a low-resource language through the adaptation of an ASR system build for a high-resource language, the low-resource language we used in the current work is not an actual low-resource language. We plan to extend this work to Mboshi [1], a Bantu language which is an actual low-resource language.

5. Acknowledgements

The work reported here was started at JSALT 2017 in CMU, Pittsburgh, and was supported by JHU and CMU via grants from Google, Microsoft, Amazon, Facebook, and Apple. This work used the Blacklight system (NSF award number ACI-1041726) of XSEDE (NSF award number ACI-1053575) at the Pittsburgh Supercomputing Center (PSC). OS was partially supported by a Vidi-grant from NWO (276-89-003). The authors would like to thank Markus Müller for providing the forced alignments of FlickR_8K, and Graham Neubig for providing the tokenised Japanese translations, and for implementing and providing the xnmt/dynet tools for the translated text retrieval task.

6. References

[1] Adda, G., Stüker, S., Adda-Decker, M., Ambouroue, O., Besacier, L., Blachon, D., Bonneau-Maynard, H., Godard, P., Hamlaoui, F., Idiatov, D., Kouarata, G.-N., Lamel, L., Makasso, E.-M., Rialland, A., Van de Velde, M., Yvon, F., Zerbian, S., "Breaking the unwritten language barrier: The BULB project", Proc. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages, 2016.

[2] Hasegawa-Johnson, M., Jyothi, P., McCloy, D., Mirbagheri, M., di Liberto, G., Das, A., Ekin, B., Liu, C., Manohar, V., Tang, H., Lalor, E.C., Chen, N., Hager, P., Kekona, T., Sloan, R., Lee, A.K.C., "ASR for under-resourced languages from probabilistic transcription," IEEE/ACM Trans. Audio, Speech and Language 25(1):46-59, 2017. doi:10.1109/TASLP.2016.2621659

[3] Huang, P.-S., Hasegawa-Johnson, M., "Cross-dialectal data transferring for Gaussian Mixture Model training in Arabic speech recognition," in Internat. Conf. Arabic Language Processing (CITALA) pp. 119-122, ISBN 978-9954-9135-0-5, Rabat, Morocco, 2012.

[4] Jansen, A., Dupoux, E., Goldwater, S., Johnson, M., Khudanpur, S., Church, K., Feldman, N., Hermansky, H., Metze, F., Rose, R., "A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition," Proc. ICASSP, 2013.

[5] Ondel, L., Burget, L., Cernocky, J., "Variational Inference for Acoustic Unit Discovery", Procedia Computer Science, 81, Elsevier Science, http://www.fit.vutbr.cz/research/view_pub.php?id=11224, 2016.

[6] Harwath, D., Glass, J., "Deep multimodal semantic embeddings for speech and images," IEEE Automatic Speech Recognition and Understanding Workshop, Scottsdale, Arizona, USA, 237-244, 2015.

[7] Badino, L., Canevari, C., Fadiga, L., & Metta, G., "An auto-encoder based approach to unsupervised learning of subword units", Proc. ICASSP, 2014.

[8] Huijbregts, M., McLaren, M., van Leeuwen, D., "Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection", Proc. ICASSP, 4436–4439, 2011.

[9] Lee, C., Glass, J., "A nonparametric Bayesian approach to acoustic model discovery", Proc. 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, 40-49, 2012.

[10] Varadarajan, B., Khudanpur, S., Dupoux, E., "Unsupervised learning of acoustic subword units", Proc. ACL-08: HLT, 165-168, 2008.

[11] Jansen, A., Van Durme, B., "Efficient spoken term discovery using randomized algorithms", Proc. Automatic Speech Recognition and Understanding (ASRU), 401-406, 2011.

[12] Park, A. S., Glass, J. R., "Unsupervised pattern discovery in speech", Proc. ICASSP, 16(1), 186-197, 2008.

[13] Zhang, Y., Glass, J. R., "Towards multi-speaker unsupervised speech pattern discovery", Proc. ICASSP, 4366-4369, 2010.

[14] Harwath, D., Torralba, A., Glass, J., "Unsupervised learning of spoken language with visual context", Advances in Neural Information Processing System, 1858-1866, 2016.

[15] Chrupała, G., Gelderloos, L., Alishahi, A., "Representations of language in a model of visually grounded speech signal", arXiv:1702.01991v3 [cs.CL] 30 Jun 2017.

[16] Schultz, T., Waibel, A., "Experiments on cross-language acoustic modelling," Proc. Interspeech, 2001.

[17] Lööf, J., Gollan, C., Ney, H., "Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system," Proc. Interspeech, 2009.

[18] Vesely, K., Karafiát, M., Grezl, F., Janda, M., Egorova, E., "The language-independent bottleneck features," in Proc. SLT, 2012.

[19] Xu, H., Do, V.H., Xiao, X., Chng, E.S., "A comparative study of BNF and DNN multilingual training on cross-lingual low-resource speech recognition," Proc. Interspeech, 2132-2136, 2015.

[20] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G.D., Roskies, R., Scott, J.R. and Wilkens-Diehr, N., "XSEDE: Accelerating scientific discovery", Computing in Science & Engineering. 16(5):62-74, 2014.

[21] Nystrom, N., Welling, J., Blood, P. and Goh, E.L., "Blacklight: Coherent shared memory for enabling science", In *Contemporary High Performance Computing: From Petascale Toward Exascale*; Vetter J., Ed.; CRC Computational Science Series, Taylor & Francis:Boca Raton, 431-450, 2013.

[22] Oostdijk, N.H.J., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M., Baayen, H., "Experiences from the Spoken Dutch Corpus project", Proc. LREC – Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, 340-347, 2002.

[23] Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J., "Collecting image annotations using Amazon's Mechanical Turk", Proc. NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, 2010.

[24] Nguyen, T.-S., Müller, M., Sperber, M., Zenkel, T., Kilgour, K., Stüker, S., Waibel, A., "The 2016 KIT IWSLT speech-to-text systems for English and German", Proc. 13th International Workshop on Spoken Language Translation (IWSLT), Seattle, USA, December 8-9, 2016

[25] Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," Proc. 23rd international conference on Machine learning (ACM), 369-376, 2006.

[26] Miao, Y., Gowayyed, M., Metze, F., "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," arXiv:1507.08240v3, 2015.

[27] Black, A.W., "CLUSTERGEN: A statistical parametric speech synthesizer using trajectory modeling", Proc. ICSLP, 1762-1765, 2006.

[28] Black, A.W., Kumar Muthukumar, P., "Random forests for statistical speech synthesis", Proceeding of Interspeech, 1211-1215, 2015.

[29] Fukada, T., Tokuda, K., Kobayashi, T., Imai, S., "An adaptive algorithm for mel-cepstral analysis of speech", Proc. ICASSP, 1992. doi="10.1109/ICASSP.1992.225953.

[30] https://github.com/neulab/xnmt

[31] Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulous, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., Duh, K., Faruqui, M., Gan, C., Garrette, D., Ji, Y., Kong, L., Kuncoro, A., Kumar, G., Malaviya, C., Michel, P., Oda, Y., Richardson, M., Saphra, N., Swayamdipta, S., Yin, P., "DyNet: The dynamic neural network toolkit", arXiv preprint arXiv:1701.03980, 2017.

[32] Fiscus, J., "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," IEEE Workshop ASRU pp. 347-354, 1997.

[33] Gillick, L., Cox, S.J., "Some statistical issues in the comparison of speech recognition algorithms," Proc. ICASSP 532-535, 1989.

[34] Jaitly, N., Hinton, G.E., "Vocal tract length perturbation (VTLP) improves speech recognition", Proc. ICML Workshop on Deep Learning for Audio, Speech and Language, 2013.