# Lightly Supervised Automatic Subtitling of Weather Forecasts

*Joris Driesen, Steve Renals*

Center for Speech Technology Research, University of Edinburgh, UK

{jdriesen,srenals}@inf.ed.ac.uk

## Abstract

Since subtitling television content is a costly process, there are large potential advantages to automating it, using automatic speech recognition (ASR). However, training the necessary acoustic models can be a challenge, since the available training data usually lacks verbatim orthographic transcriptions. If there are approximate transcriptions, this problem can be overcome using light supervision methods. In this paper, we perform speech recognition on broadcasts of Weatherview, BBC's daily weather report, as a first step towards automatic subtitling. For training, we use a large set of past broadcasts, using their manually created subtitles as approximate transcriptions. We discuss and and compare two different light supervision methods, applying them to this data. The best training set finally obtained with these methods is used to create a hybrid deep neural network-based recognition system, which yields high recognition accuracies on three separate Weatherview evaluation sets.
**Index Terms**: Light supervision, Transcription, Segmentation, Acoustic Model Training, Subtitling

## 1. Introduction

Subtitles provide a transcription of a television soundtrack played in synchronisation with the broadcast content, to provide better access for deaf and hard-of-hearing people. In the UK, broadcasters such as the BBC[1] aim to provide subtitles for 100% of broadcast content. Subtitle generation of live, unscripted programmes is usually carried out by "respeaking", whereby a trained operator re-speaks the broadcast speech into a commercial large vocabulary dictation system. This is a rather labour intensive process, especially when multiple live programmes need to be transcribed simultaneously as in the case of regional news or weather forecasts (in the UK, the national broadcaster can provide 15–20 separate versions of a weather report, each focusing on a different part of the country).

This work has been funded by the European Union as part of the Seventh Framework Programme, under grant agreement no. 287658 (EU-BRIDGE). Thanks to John McLoughlin, Matt Simpson, and Nicola Greaves of Red Bee Media.

[1] British Broadcasting Corporation: the UK's national radio and television corporation

In this paper we are concerned with the automatic subtitling of weather forecasts, which because of the regional content is particularly labour-intensive. A typical weather forecast has a duration of 3 minutes. It is a domain specific task, made up largely of planned speech. However the speech is somewhat expressive in style, and the speaking rate is very high (an average of 210 words/minute, with some broadcasts containing over 700 words in 3 minutes).

The wide availability of subtitles makes it possible to train speech recognition systems on in-domain data. However, subtitles do not provide a verbatim transcription, a prerequisite for training in-domain acoustic models. The purpose of subtitles is to convey understanding of the speech content briefly and efficiently to the viewers, not to transcribe it literally. Therefore, even the best subtitles can differ from the spoken content in a variety of ways, i.e., the order of the words can be different; different words may be used; and interjections, repetitions or hesitations can be omitted. In the case of live subtitles produced by respeaking, there may be additional speech recognition errors of 2–3% (or more). Furthermore, although the timing of the subtitles is supposed to match that of the corresponding speech exactly, this cannot always be wholly relied on — a problem which is again more acute in the case of live subtitling. Since there is no reliable segmentation of the audio, one has to be created in an unsupervised (or lightly supervised) way.

The main focus of this paper is how we can use in-domain broadcast acoustic data and the accompanying subtitles to develop a system for weather forecast transcription. There is a large amount of speech data available which comes with imperfect transcriptions, including audio books and lectures, as well as broadcast material, and there has been considerable interest in methods to deal with such imperfect transcripts for speech recognition training, e.g [1, 2].

There are basically two ways to approach this problem: one is to make a forced alignment of speech with imperfect transcripts, leaving the possibility for some limited corrections, i.e., insertions, deletions or substitutions [3, 4, 5]. The other approach is to make a LVCSR (Large Vocabulary Continuous Speech Recognition) alignment of the audio using a language model that is heavily biased towards the contents of that speech segment [2, 6]. Meth-

ods have also been proposed to combine both approaches [1, 5]. All of these methods require an automatic alignment of audio with text, which can be computationally expensive and error-prone if the audio segments are long. Therefore, most methods require an a priori unsupervised segmentation of the audio based on silence/speech detection. In this paper, we discuss in detail two approaches to light supervision. The first one, in Section 2, is the method from [4], which is based on finite skip networks. The other, discussed in Section 3, is based on ASR decoding using a biased language model estimated from the imperfect transcripts. It is similar to the work in [6], but foregoes the acoustics-based segmentation of the audio, required by other methods.

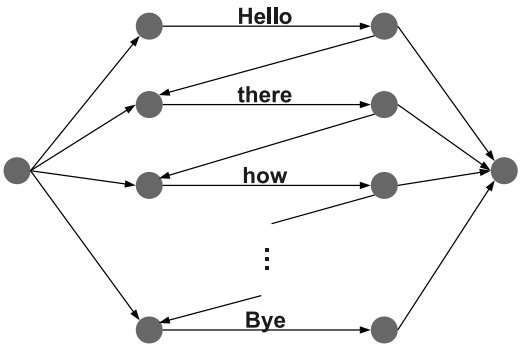
We then report on a set of experiments to test these approaches to light supervision using a training corpus of 1,446 three-minute weather forecasts, with results reported in terms of the amount of acoustic data that could be aligned and the resultant word error rate (WER) of an HMM/GMM-based speech recognition. Finally, using the best alignments generated in these experiments we train a hybrid deep neural network-based speech recognition system, reporting final results on three test sets.

## 2. Skip networks

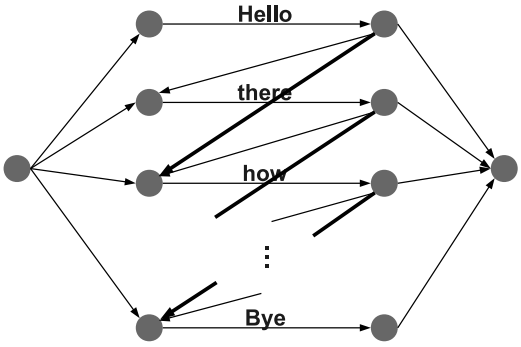
The first method for light supervision is based on a method which constructs finite state networks from the provided transcripts, and is designed to work well with weak acoustic models [4]. When presented with a short segment of audio data, this method searches through a text document for a piece of text that matches with the audio segment. Unlike biased language model approaches, the matched text is constrained to be a contiguous sequence of words. If found, this segment can then be added to the training set, using the corresponding text as the orthographic transcription.

The matching procedure works as follows: from the text document, two finite state networks are constructed as shown in Figure 1. Using the initial acoustic model, the audio segment is aligned with both of these networks. For the first one, the result is an exact word sequence occurring somewhere in the text document, with no skips. We will call this type of network a "sequence net". For the second network, the resulting alignment is also a fragment of the text document, but words may be missing from it.

There are then a number of criteria that must be met, for the audio segment to be accepted as valid training data. First, the two alignments must be identical, and their likelihood must also be equal. Failing this test indicates that there are acoustic deletions, i.e., words in the text that were not spoken in the audio. Secondly, the drop in likelihood over each word in the alignment must not exceed a pre-set threshold. If it does, there may have been acoustic substitutions or insertions, i.e., the word in question may be aligned with a different word in the audio, or multiple words. From a large batch of audio segments, the ones that are not rejected in this process are collected into a training set, with their alignments as transcription. A new acoustic model is then trained, and the process may be iterated. It is assumed a larger training set leads to a better acoustic model. In turn, a higher quality model yields a better alignment, and hence a larger training set for the next iteration. Thus, the training set obtained by this method grows with each iteration.




(a) Sequence net




(b) Skip net (additional arcs drawn in bold)

Figure 1: *An example of the finite state grammars constructed from a text. The arcs that don't have words are $\varepsilon$-arcs*

## 3. Greedily matching full alignments

The second method for light supervision is loosely based on the work in [6] and [2]. It hinges on the creation of a language model that is strongly biased towards the domain of the imperfectly transcribed training data. Using such a language model, audio from this training set can be aligned very accurately, even if the acoustic model is of lesser quality. These alignments are then compared with the available transcripts, and matching sequences assigned to the new training set. The major difference between our approach and these previous methods is that segments are not entirely rejected upon encountering differences between alignment and transcriptions. With dy-

namic programming, we detect the longest subsequences occurring in both, and discard only the parts that don't match. Subsequences below a certain minimum length are also rejected, since they are likely to match by mere coincidence. This is very different from the method of Section 2, since the amount of data recovered in this way is not influenced by the length of the segments, allowing us to consider segments of arbitrary length. Whereas before, a typically short piece of audio was matched with a small snippet of a larger text document, we can now consider longer audio segments, which roughly correspond to entire text documents or even encompass them. Note that this approach completely disregards the notion of sentences, since matching subsequences may freely cross sentence boundaries. This is not a problem, since our purpose is merely to obtain a good set of acoustic training data with matching transcriptions, whether the segments in this training data are complete syntactic sentences or not. This sets our method apart from e.g. [2], where the text is first split into sentences, and one sentence is assigned to each audio segment. As before, this procedure of aligning audio, matching alignments and retraining acoustic models, is repeated several times, yielding a larger training set and an improved acoustic model with every new iteration.

The method described above may require the automatic alignment of long audio segments, which can be computationally demanding. This problem can be solved by splitting audio files blindly into overlapping segments, aligning them, and concatenating the resulting alignments using a form of dynamic programming. Concretely, consider two word sequences $S_1 = \{w_1, w_2, \ldots, w_m\}$ and $S_2 = \{v_1, v_2, \ldots, v_n\}$ resulting from the alignments of overlapping audio chunks. Because of the overlap, it is reasonable to assume the last part of $S_1$ is approximately equal to the first part of $S_2$. In other words, we can safely assume that there exist indices $a \in [1, m]$ and $b \in [1, n]$, such that $\{w_a, w_b, \ldots, w_m\} \approx \{v_1, v_2, \ldots, v_b\}$. This equality is only approximate, since there may be insertions, deletions or substitutions in one or both of the alignments. Finding the overlapping part of the two word sequences is therefore not trivial. One way to do it, is by performing a dynamic alignment. We define a cost $c_{i,j}$ with $i \in [0, m]$ and $j \in [0, n]$, define $c_{0,0} = 0$, and update it locally as follows:

$$c_{i,j} = \min \begin{cases} c_{i-1,j} & \text{if } j == 0 \\ c_{i,j-1} & \text{if } i == m \\ 1 + c_{i-1,j} & \text{if } j > 0 \\ 1 + c_{i,j-1} & \text{if } i < m \\ -1 + c_{i-1,j-1} & \text{if } w_i == v_j \text{ and they overlap} \\ 1 + c_{i-1,j-1} & \text{if } w_i \neq v_j \text{ or they do not overlap} \end{cases}$$

As can be seen from this, for two words $v_i$ and $w_j$ to match, they not only have to match orthographically, but they must also roughly occur at the same time. By recording in each step the previous position, the best path from $(0, 0)$ to $(m, n)$ is determined. From this path, a unified word alignment can be made.

This procedure is reminiscent of ROVER, a technique to merge several different transcriptions of the same audio [7]. In each position $(i, j)$ of the backtrace, either word $w_i$ from sequence $S_1$, or $v_j$ from sequence $S_2$ is appended to the merged alignment. To decide between these two, a simple heuristic is applied. Since the acoustic segments were split blindly, there may be partial words at their boundaries, which are likely to cause errors and misalignments. We therefore insert the word that is the furthest away from its segment's "ragged end". The result is a reliable alignment of audio segments of arbitrary length. Although it is still possible the blind splitting of audio segments introduces some errors, an unsupervised segmentation based on acoustic cues is not guaranteed to do better. Usually such algorithms require a substantial amount of parameter tuning, and can perform poorly on data they were not optimised for. The method presented here works equally well on any data, at the cost of an increased computational complexity, due to the overlap between audio segments.

## 4. Experiments

We have performed experiments using a corpus of BBC television weather forecast programmes, named *Weatherview*. This is a daily programme, scheduled to be 3 minutes in duration, consisting of a discussion of the UK's weather on the day of the broadcast itself, and a forecast for the following few days. It adheres to a typical weather forecast format, with only one possibly regionally accented English speaker, the weather forecaster, standing in front of a map, directing his speech directly to the viewer. A screenshot of this can be seen in Figure 2. The presenters are highly skilled in timing and pacing their presentation, such that it fits exactly in the allotted time slot. As a result, deviations from the scheduled duration are usually very small, typically less than a few seconds. Since there is a tendency to convey a maximum of information in the given time, the rate of speech in these weather reports is high. Most reports contain between 600 and 700 words, i.e., a sustained rate of 3-4 words per second. For comparison, the standard recommendation for audiobooks is only 150-160 words per minute [8].

### 4.1. Weatherview corpus

The Weatherview corpus used in this paper contains 1,446 broadcasts from the period between 2008 and 2012. In many of the supplied recordings, the actual weather report is preceded and succeeded by a substantial amount of silence and out-of-domain (OOD) acoustic data, in most cases programme trailers. Including this OOD data the supplied training set has a total duration of 116.4 hours. The audio data is in stereo, has a sample rate of 48 kHz and is encoded in MP3 format. Verbatim tran-


Figure 2: *A screenshot of a Weatherview report.*

| | train | dev1 | dev2 | test |
|---|---|---|---|---|
| total | 1243622 | 6064 | 5966 | 11802 |
| unique | 10307 | 894 | 891 | 1134 |

Table 1: *The total number of words, and the number of unique words in each set*

scriptions for this training data are not available, however pre-recorded subtitles are provided. Each subtitle file matches an audio file, including its surrounding OOD data. In addition to the training set, the Weatherview corpus contains two development sets and an evaluation set. Both development sets, henceforth referred to as 'dev1' and 'dev2', contain 9 broadcasts each. The evaluation set, 'test', is twice as large, containing 18 broadcasts. Unlike the training set, the recordings in these sets are not preceded, nor followed by OOD data, and come with manually segmented verbatim transcriptions, allowing accurate ASR evaluation. Table 1 shows the number of word types and word tokens in the training and test sets.

Pre-processing on the audio data was performed as follows: first, it was converted from MP3 to WAV format, the two stereo channels mixed together, and downsampled to 16 kHz. Perceptual Linear Prediction (PLP) coefficients [9] (13 dimensions, including C0) were then computed using a 25-ms window, with a frameshift of 10 ms. Cepstral Mean Normalisation (CMN) was then applied and $\Delta$ and $\Delta\Delta$ features added. A 9-frame context window was used, resulting in 351-dimension feature vectors, which were then transformed back to 39 dimensions by means of a Maximum Likelihood Linear Transformation (MLLT) [10]. All acoustic models we trained on this data, unless otherwise specified, share the same characteristics: they are speaker-independent GMM-HMM models trained with maximum likelihood. They contain 3,000 context-dependent states, with a total of 48,000 diagonal-covariance gaussians.

Both methods discussed above, in Sections 2 and 3, require an initial acoustic model. We trained such a model on a orthographically transcribed set containing 15.8 hours of BBC radio recordings, and 5 hours of audio from a fictional drama series [11]. There is some overlap in domain between this data and Weatherview, since the radio part contains a number of weather reports. There are far too few, however, for the resulting model to be considered as in-domain. The language model used in our experiments is based on one used for the automatic transcription of meetings [12]. It was first biased towards the 21 hours of BBC data on which our initial acoustic model was trained. Then, we biased the resulting model towards the Weatherview corpus using the SRILM toolkit [13, 14]. Concretely, a simple 3-gram model was constructed using all 1,446 subtitles from the Weatherview training set, with an additional 553 Weatherview subtitle files for which audio was not available. This model was then interpolated with the background model using an interpolation factor of 0.9.

The original lexicon was also the same as that used in [12]. However, the Weatherview training data contained a large number of OOV words which were added to this lexicon using the Sequitur grapheme-to-phoneme (g2p) conversion tool [15]. As may be expected from weather reports, many of the OOV words were place names and other given names. A minority was due to the usage of non-standard language and neologisms, e.g. "thundery", "guesstimation", "slowish", etc. The majority of OOVs, however, were caused by typographical errors in the Weatherview subtitles. Although it is possible to fix such errors manually, we have chosen not to do so, since our aim is to avoid such human supervision. Instead, we have left these words for the g2p conversion to deal with, however imperfectly.

### 4.2. Skip nets

The training method discussed in Section 2 takes short audio segments and matches them with fragments of a larger text document. This does not match the setup of the Weatherview training data, which contains audio files of minimally 3 minutes that correspond from beginning to end with their respective subtitle files. We have thus performed an unsupervised segmentation of the audio using 'adintool', a part of the open-source ASR toolkit Julius [16], which uses the energy and zero-crossing rate of an acoustic signal to split it into speech segments, skipping the silences. The resulting set of speech segments varied in duration from less than a second, to several minutes. Segments of less than 1 second were then discarded, since they are unlikely to contribute much to the final set of training data. Segments longer than 1 minute were also discarded, since their alignment with ASR can be expensive. The remaining segments have a total duration of 67.11 hours. We have then applied several iterations of lightly supervised training as explained in Section 2, the results of which can be seen in Table 2.

| | # hours | WER dev1 | WER dev2 | WER test |
|---|---|---|---|---|
| init | / | 20.1 | 23.7 | 19.4 |
| skip iter1 | 11.42 | 15.3 | 16.3 | 15.4 |
| skip iter2 | 15.80 | 14.6 | 15.8 | 15.8 |
| skip iter3 | 16.93 | 14.4 | 15.5 | 15.8 |
| match iter1 | 48.44 | 12.4 | 13.4 | 12.4 |
| match iter2 | 51.93 | 12.1 | 13.1 | 12.2 |
| match iter3 | 52.34 | 11.7 | 13.0 | 12.0 |
| DNN | | **8.7** | **9.7** | **9.0** |

Table 2: *All results of the light supervision methods throughout their iterations. The amount of collected training data is shown, as well as the WERs obtained with a simple ASR system trained on it*

### 4.3. Greedy matching with alignments

To apply the light supervision method of Section 3 on Weatherview, no further preparation of the data is needed. Alignments are made of all 1446 broadcasts in the training set. This was done using the splitting and merging technique explained above. Broadcasts were blindly split into 1 minute segments with an overlap of 40 seconds, aligned and then combined into long coherent word sequences. This method may seem unnecessary for the Weatherview broadcasts, since aligning 3 minute audio files in one block is quite feasible, though expensive. However, when the OOD data surrounding the weather report is included, these files can be much longer, in rare cases up to 15 minutes. The alignment of each audio file was then compared with its corresponding subtitle file and the longest possible word sequences occurring in both are collected in a greedy way. Matching word sequences must be at least 3 words long in order to be accepted into the updated training set. Using this method, we updated the training set and the acoustic models trained on them for three iterations. The results are shown in Table 2.

### 4.4. Results and Discussion

In the evaluation of these methods on Weatherview, we wish to measure not only the *quantity* of the ASR training data recovered, but also its *quality*. This is why in Table 2, we not only show the size of the collected training set in each iteration, but also ASR results obtained by using this set to train simple acoustic models. These are the same models, in fact, that were used to create the training set of the next iteration. From the results, we see immediately there is a very large difference in performance between the two methods. One may argue that the comparison is not entirely fair, since the first method only has 67 hours of raw data to extract a training set from, whereas the second uses the full 116 hours available. This reduction of usable data, however, is a direct consequence of constraining ASR inputs to durations of 60 seconds or less, a constraint that is equally enforced in both methods. Moreover, the difference in initial data sets fails to explain the gap in performance completely. With roughly 57% of the initial data, the first method yields training sets that are about three times smaller than those of the second method. A likely explanation can be found in the unsupervised segmentation, which in the case of Weatherview is an exceptionally difficult task. There are several reasons for this: firstly, due to the fast speaking rate, pauses and silences are extremely short and are often confused with plosive stops, resulting in a large number of boundaries placed within words. This results in partial words at the segment boundaries, which in turn leads to errors when aligning them with sequence or skip nets. Secondly, with such speaking rates, cross-word pronunciation effects take place, causing coarticulation, reduction and contraction of words. This is especially detrimental for shorter words, since the alignment with a skip net is likely to mark them as deletions. A single deletion like this suffices for the entire segment to be rejected, regardless of its length. In conclusion, the word net method, although very successfully applied in other tasks [4], has proven ill-suited for the alignment of Weatherview data. This is not so surprising, since it was in fact designed for situations where no prior knowledge is available: no initial acoustic model and no biased language model. The original paper even foregoes the usage of a known phone set and lexicon, relying on grapheme models instead.

The results of the greedy matching method are highly encouraging. Not only do the obtained ASR results improve with each iteration, the minimum seems to be far from reached at the point where we left off, after 3 iterations. Note that the size of the obtained training set does not grow accordingly. Between iteration 2 and 3, the relative increase of the corpus is no larger than 0.7%. This supports the notion that the amount of ASR training data and the eventual WERs are not as closely linked as is sometimes naively assumed.

## 5. Building a state-of-the-art ASR system

The recognition system that produced the ASR results in Table 2 is relatively simplistic, intended for evaluating the different training sets rather than producing a competitive ASR score. In this section, we consider the training set obtained at the end of iteration 3 of the greedy matching algorithm and use it to train a more sophisticated system. Concretely, we use this data to train a Deep Neural Network (DNN), which we will use in a hybrid setup, similar to the one in [17]. In a first step, a context window of 11 frames is slid over the 39-dimensional spectrograms from before, yielding input vectors of length 421 ($11 \cdot 39$). These vectors are the inputs for the DNN, which in our experiments consists of 6 hidden layers, each containing 2048 nodes. It was initialised by greedily training each layer as a Restricted Boltzman Machine (RBM) [18]. In

a following step, a GMM-HMM model is trained with speaker-adaptive training (SAT), using a single fMLLR transform per Weatherview broadcast. Using this acoustic model, an alignment of the training data is created, which functions as the training target for the DNN, allowing supervised training using backpropagation. The results of this setup on the different sets of Weatherview data are shown at the bottom of Table 2.

## 6. Conclusion and Future Work

We have applied a lightly supervised training technique in which the absolute minimum of possible training data is discarded, while still maintaining confidence that each training segment fully matches its corresponding transcription. We have demonstrated that with this technique, a highly accurate ASR system can be trained on imperfectly transcribed data. With error rates around 9%, this system approaches the point where its generated transcriptions can be considered for subtitling. To this end, future work will include a number of post-processing steps, such as the automatic insertion of punctuation marks, and the shortening of wordy alignments into a more concise form, better suited for subtitling.

## 7. References


[1] A. Katsamanis, M. Black, P. Georgiou, L. Goldstein, and S. Narayanan, "Sailalign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, January 2011.

[2] N. Braunschweiler, M. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *Proc. Interspeech*, September 2010, pp. 2222–2225.

[3] P. Moreno and C. Alberti, "A factor automaton approach for the forced alignment of long speech recordings," in *Proc. ICASSP 2009.*, 2009, pp. 4869–4872.

[4] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miami, Florida, USA, December 2012.

[5] P. Placeway and J. Lafferty, "Cheating with imperfect transcripts," in *Proc. ICSLP*, vol. 4, 1996, pp. 2115–2118.

[6] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, no. 1, pp. 115–129, 2002.

[7] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proc ASRU*, 1997, pp. 347–354.

[8] J. R. Williams, "Guidelines for the use of multimedia in instruction," in *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting*, 1998, pp. 1447–1451.

[9] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *Journal for the Acoustic Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[10] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[11] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[12] T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grézl, A. El Hannani, M. Huijbregts, M. Karaát, M. Lincoln, and V. Wan, "Transcribing meetings with the amida systems," *IEEE transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, February 2012.

[13] A. Stolcke, "SRILM an extensible language modeling toolkit," in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002.

[14] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: Update and outlook," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Waikoloa, Hawaii, December 2011.

[15] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[16] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine," in *Proc. Eurospeech*, Aalborg, Denmark, 2001, pp. 1691–1694.

[17] P. Swietojanski, G. A., and S. Renals, "Unsupervised cross-lingual knowledge transfer in dnn-based LVCSR," in *Proc. IEEE Workshop on Spoken Language Technology*, Miami, Florida, USA, December 2012.

[18] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, 2006.