Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2016) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

5th Workshop on Spoken Language Technology for Under-resourced Language, SLTU 2016, 9-12 May 2016, Yogyakarta, Indonesia

Refining sparse coding sub-word unit inventories with lattice-constrained Viterbi training

Wiehan Agenbag[1], Thomas Niesler[2]

Department of Electrical and Electronic Engineering, Stellenbosch University, Stellenbosch, South Africa

Abstract

We investigate the application of two novel lattice-constrained Viterbi training strategies to the task of improving sub-word unit (SWU) inventories that were discovered using an unsupervised sparse coding approach. The automatic determination of these SWUs remain a critical and unresolved obstacle to the development of ASR for under-resourced languages. The first lattice-constrained training strategy attempts to jointly learn a bigram SWU language model along with the evolving SWU inventory. We find that this substantially increases correspondence with expert-defined reference phonemes on the TIMIT dataset, but does little to improve pronunciation consistency. The second approach attempts to jointly infer an SWU pronunciation model for each word in the training vocabulary, and to constrain transcription using these models. We find that this lightly supervised approach again substantially increases correspondence with the reference phonemes, and in this case also improves pronunciation consistency.

Peer-review under responsibility of the Organizing Committee of SLTU 2016.

Keywords: segmentation, clustering, sparse coding, sub-word units, Viterbi training

1. Introduction

We investigate the application of novel lattice-constrained Viterbi training strategies to the task of improving sub-word unit (SWU) inventories that were discovered using an unsupervised sparse coding approach. The unsupervised acquisition of high-quality SWU inventories is critical for implementing automatic speech recognition (ASR) for under-resourced languages, since it would obviate the resource-intensive task of recruiting trained linguists to design the inventories and produce pronunciation lexicons.

1.1. Sub-word units discovered by sparse coding

Sparse coding attempts the reconstruction of an input signal using a linear combination (called the *code*) of the fewest number of basis functions taken from a finite set. In the context of speech, we may consider a typical utterance

[1] wagenbag@sun.ac.za
[2] trn@sun.ac.za

Peer-review under responsibility of the Organizing Committee of SLTU 2016.

to be our input signal, which we wish to code using a highly sparse selection of sub-word units (SWU's), which act as basis functions.

In [1], we investigated the use of a shift and scale invariant sparse coding framework with non-overlapping basis functions for the unsupervised discovery of SWU inventories. This approach led to reasonable SWU's, but the transcription of the training utterances in terms of these units was generally inconsistent at the word level and as a consequence not directly useful for ASR. The goal of this study was to improve the SWU inventory and to extract more consistent transcriptions.

2. Related work

Previous applications of sparse coding on speech have primarily focused on feature extraction [2,3,4,5]. Existing approaches to unsupervised SWU discovery generally rely on blind segmentation and clustering [6,7,8,9], or on clustering context-dependant graphemes and G2P [10,11,12]. Other approaches rely on hierarchical Bayesian models [13,14] that try to jointly learn SWU inventories and pronunciation dictionaries.

The work in [6] uses HMM's with unigram SWU emission probabilities for pronunciation modelling, but stops short of jointly learning these models while producing new SWU transcriptions. The work presented in [9] also employs Viterbi training to improve SWU inventories and transcriptions, however our word-level lattice-based constraints are, as far as we know, novel.

3. Lattice-constrained refinement

The sparse coding approach presented in [1] suffers from some deficiencies which could be addressed to improve the quality of the discovered SWU inventories:

1. Sparse coding basis functions can only warp linearly, while speech generally warps non-linearly;
2. silences and pauses are not explicitly modelled;
3. no attempt is made to discover an underlying linguistic pattern in the sequential use of the discovered SWU's to transcribe speech, which could be reinforced to create more consistent transcriptions; and
4. the approach can not be easily extended to take advantage of knowledge about the particular word sequence of the utterance under consideration.

The first of these points can be addressed by modelling each SWU as a three-state left-to-right HMM with GMM's governing each state's emission probabilities, as is common in ASR applications. There is a potential information loss incurred by the imposition of this, since the prototype basis functions used during sparse coding could capture many more frames of temporal information than the HMM's used in speech typically have states. However, some of the loss is compensated for because GMM's also capture the variance at each state.

The second point can be dealt with by explicitly adding a unit modelling silence to the SWU inventory. In order to train this unit, we can take advantage of the fact that there is a larger likelihood of silences occurring at the beginning and end of an utterance, as well as between words.

We now introduce two novel approaches to SWU inventory determination, which addresses the third and fourth points.

3.1. Bigram constrained Viterbi training

The third point can be dealt with by attempting to jointly learn an N-gram SWU language model along with the SWU inventory, by iteratively reestimating the language and acoustic models from the produced SWU transcriptions, and then using those language and acoustic models to produce new transcriptions. We hypothesise that this could reinforce the use of likely SWU sequences, while diminishing the use of unlikely sequences and in doing so result in more consistent transcriptions.

3.2. Word-level SWU pronunciation modeling

The last deficiency will be addressed in this paper by attempting to learn an SWU pronunciation model for each word in the training corpus, and then constraining the transcription of each utterance by a decoding lattice formed by chaining the word models of each utterance together. This would allow pronunciation knowledge to be aggregated from all instances of a word and encourage all SWU transcriptions of that word to become more consistent. Since this approach requires word transcriptions of the training data, it can be considered lightly supervised.

3.2.1. Word pronunciation model

We follow the approach proposed in [6], of modelling each word w_j in the vocabulary as a single-state HMM (shown in Figure 1) emitting SWU's according to a unigram word pronunciation model $p(u_i|w_j)$. The self-transition probability a_s and word-exit transition probability a_e can be thought of as governing the length of the word in terms of the number of SWU's used to pronounce it. These single-state HMM's can subsequently be chained according to the

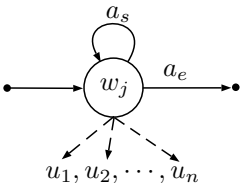


Fig. 1. Single-state word HMM.

word order of a particular utterance to form a composite HMM, which can be used to perform forced alignment or embedded reestimation.

3.2.2. Model initialisation

We choose uniform distributions for the initial $p(u_i|w_j)$. The transition probabilities of word w_j with character length n_j are set to

$$a_s(w_j) = \frac{n_j}{n_j + 1}, \quad a_e(w_j) = 1 - a_s(w_j). \tag{1}$$

This initialisation ensures that longer words have higher self-transition probabilities than shorter words.

3.2.3. Model reestimation

Given a set of observed SWU sequences, as well as some prior estimates for our word pronunciation models, we can produce updated model parameters by applying a number of iterations of embedded reestimation with the Baum-Welch EM algorithm.

One of the estimates that is produced as part of the Baum-Welch algorithm, is the expected number of times $\gamma_{i,j}$ that an SWU u_i is aligned with word w_j. This estimate can be used directly to produce the updated SWU emission models $p'(u_i|w_j)$:

$$p'(u_i|w_j) = \frac{\gamma_{i,j}}{\sum_i \gamma_{i,j}}. \tag{2}$$

However, many words in the vocabulary occur very infrequently, leading to very poor (and overly confident) estimates in those cases. To combat this, we apply add-one smoothing:

$$p'(u_i|w_j) = \frac{\gamma_{i,j} + 1}{\sum_i \gamma_{i,j} + N}. \tag{3}$$

This affects infrequent words disproportionately, since their expected counts will be smaller, effectively *backing off* to the uniform distribution, whereas the counts of frequent words are not be significantly affected.

3.2.4. Combined training procedure

Once we have obtained word pronunciation models, we can use this knowledge to produce refined SWU transcriptions of the training data. In order to present the word pronunciation models to the speech recogniser, we encode its parameters into a word pronunciation lattice as shown in Figure 2. The short pause and silence models at the end of each word lattice allows the acoustic decoder to insert a silence before transitioning to the next word. The word-level sub-lattices are then chained into utterance-level lattices and presented, along with the SWU acoustic models, to a speech recogniser to produce new SWU transcriptions.

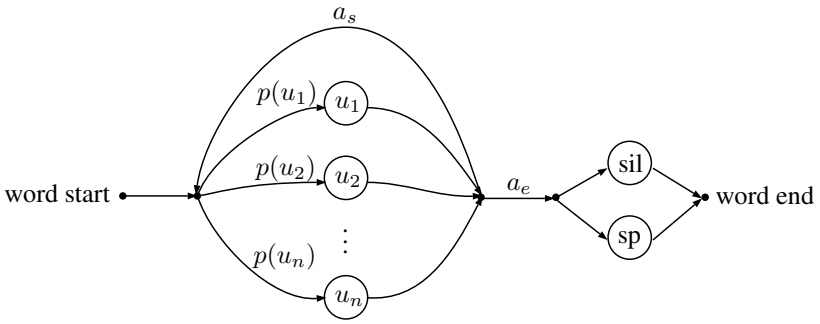


Fig. 2. Word pronunciation lattice model used to constrain SWU transcription.

With these steps in place, we can use Viterbi training to accomplish complete joint SWU and pronunciation model learning as follows:

1. With the SWU transcriptions fixed, produce updated pronunciation and SWU acoustic models.
2. Produce new SWU transcriptions with the updated pronunciation and acoustic models.
3. Repeat steps 1 and 2 until some termination criteria are met.

4. Experimental setup

The 1386 SI training utterances of the TIMIT corpus were used for experimental evaluation. These are phonetically diverse sentences each spoken only once. This choice is motivated by the desire to avoid repetition which could bias the development of sub-word units that favour very specific contexts. It is also the dataset used in [1], and therefore facilitates a comparison of results. The selected utterances were converted with HTK [15] to 39-dimensional feature vectors, consisting of twelve MFCCs, with the addition of log energy, and first and second order differential coefficients. In order to facilitate comparison with reference phonemes, the SWU rate-controlling parameters (i.e. sparse coding penalty and HVite insertion penalty) were chosen to produce units comparable in duration with phonemes, although this is not necessarily an optimal choice.

4.1. Coincidence with reference phonemes

The left-hand column of Figure 3 shows the coincidence between our SWU inventories and the TIMIT reference phonemes. These 2D coincidence histograms are computed by counting the number of times at least 50% of the span of one of our SWU's occurs within the boundaries of a reference phoneme in the TIMIT transcriptions. Further, each row is normalised to show the fraction of occurrences of a phoneme which are coded by a particular SWU.

These coincidence histograms illustrate the correspondence of our SWU inventories to those chosen by phonetic experts. However, it would be better if we could objectively quantify the correspondence in some way. To do this, we turn to two figures of merit: the entropic coding efficiency of our SWU's and the mutual information between the SWU's and the reference phonemes.

4.2. Weighted mean entropic coding efficiency

The weighted mean coding entropic coding efficiency is calculated by taking a weighted mean of the entropic efficiency of each reference phoneme's coincidence distribution with our set of SWU's:

$$\overline{\eta}_w = \sum_j p(\phi_j)\eta_j, \tag{4}$$

where η_j is the entropic coding efficiency of the j'th phoneme

$$\eta_j = \frac{H(u|\phi_j)}{H_{\max}} \in [0, 1]. \tag{5}$$

The values of η_j for the experiments in this study are shown in the right-hand column of Figure 3. The term $H(u|\phi_j)$ refers to the conditional entropy of the distribution $p(u|\phi_j)$:

$$H(u|\phi_j) = -\sum_i p(u_i|\phi_j)\log_2(p(u_i|\phi_j)), \tag{6}$$

where u_i is the i'th unit in our SWU inventory and ϕ_j is the j'th phoneme in the reference set. $H(u|\phi_j)$ can be interpreted as a measure of how spread out the corresponding conditional distribution is, ranging between zero where only one SWU is used to code the given phoneme, and $H_{\max}$, when all SWU's coincide equally with that phoneme. Thus, if a good correspondence with the reference phonemes is desired, $\overline{\eta}_w$ must be minimised.

4.3. Coding coincidence mutual information

As an additional figure of merit, we consider the mutual information between the incidence of the reference phonemes and our set of SWU's:

$$I_m(u;\phi) = \sum_i \sum_j p(u_i,\phi_j)\log_2 \frac{p(u_i,\phi_j)}{p(u_i)p(\phi_j)}. \tag{7}$$

The mutual information $I_m(u;\phi)$ is maximised when the random variables u and ϕ uniquely determine each other, i.e. when each reference phoneme corresponds to exactly one SWU.

4.4. Pronunciation consistency of extracted lexicon

Finally, we consider the consistency with which our SWU inventories transcribe the input audio into word pronunciations. We use the time-aligned word transcriptions included in TIMIT to extract word-level pronunciations from the SWU transcriptions to form a lexicon. This lexicon is then evaluated in terms of its average pronunciation entropy $\overline{H}_p$, as defined by Lee et al[13]:

$$\overline{H}_p = \frac{-1}{|V|}\sum_{w\in V}\sum_{b\in B(w)} p(b)\log_2 p(b), \tag{8}$$

with V the vocabulary of the task and $B(w)$ the observed pronunciations for word w.

The average pronunciation entropy gives an impression of the variation and spread of the pronunciations in the lexicon. It will produce lower values when there is a compact, dominant set of pronunciations for each word.

5. Results

Table 1 summarises the results of the following experiments:

1. Baseline SWU transcriptions determined through sparse coding as described in[1].

2. 40 iterations of sequence-level bigram constrained Viterbi training (SLB) as described in Section 3.1.
3. 40 iterations of word-level unigram constrained Viterbi training (WLU) as described in Section 3.2.
4. 40 iterations SLB Viterbi training followed by 40 iterations of WLU Viterbi training.

In all cases, the models were initialised from the sparse coding SWU inventory and transcriptions, and the HTK tools were used for acoustic modelling and decoding.

Table 1. Summary of experimental results

Experiment	$\overline{\eta}_w$	I_m (bits)	$\overline{H}_p$ (bits)
1) Baseline	0.655	2.030	2.383
2) Baseline + SLB	0.529	2.622	2.380
3) Baseline + WLU	0.501	2.644	2.266
4) Baseline + SLB + WLU	0.501	2.750	2.314
TIMIT reference transcript	—	—	1.180
CMUDICT	—	—	0.181

It can be seen that we have produced substantially improved SWU inventories in all cases. However, it is hard to pick a clear winner from the approaches examined here. In terms of inventory quality (i.e. entropic coding efficiency and reference phoneme mutual information) both approaches work equally well, which is promising, since the sequence-level bigram training is fully unsupervised.

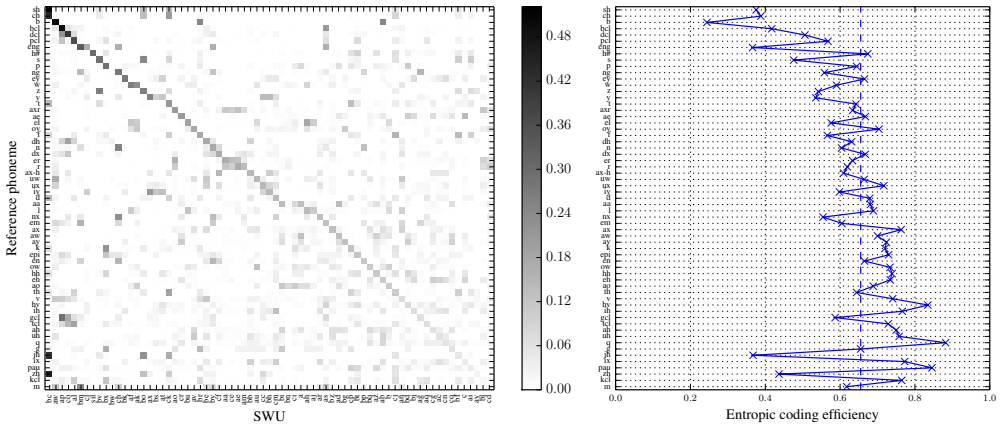
In terms of pronunciation consistency, the WLU system performs best, while the combined SLB + WLU system does slightly worse. However, the overall improvement seen in pronunciation consistency was not as great as anticipated. This may be a symptom of an overly simplistic pronunciation model, which models only the frequency of incidence of SWU's, and not their order. In order to put $\overline{H}_p$ into context, we also include this figure for a lexicon extracted from TIMIT's phone transcriptions, as well as for a hand-crafted lexicon defined by experts (CMUDICT).

6. Summary and conclusion

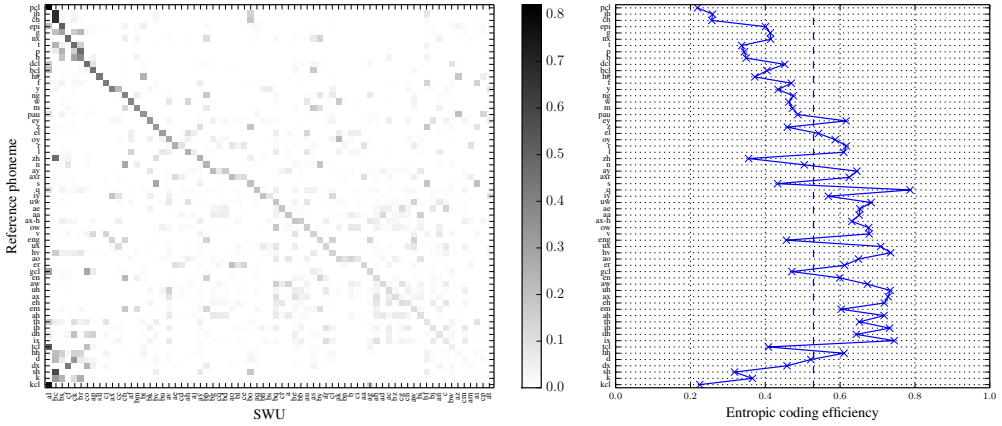
We proposed two novel lattice-constrained Viterbi training strategies for the refinement of automatically-induced SWU inventories and transcriptions. The first of these strategies attempts to jointly learn a bigram SWU language model along with the evolving SWU inventory, while the second approach attempts to jointly infer an SWU pronunciation model for each word in the vocabulary, and to constrain transcription using these models. We found that both approaches yielded substantial increases in correspondence with reference phonemes and we were able to extract more consistent pronunciations from the transcriptions. Future work will investigate more sophisticated pronunciation models as well as evaluate the ASR performance of the automatically-determined lexicons.

Acknowledgements

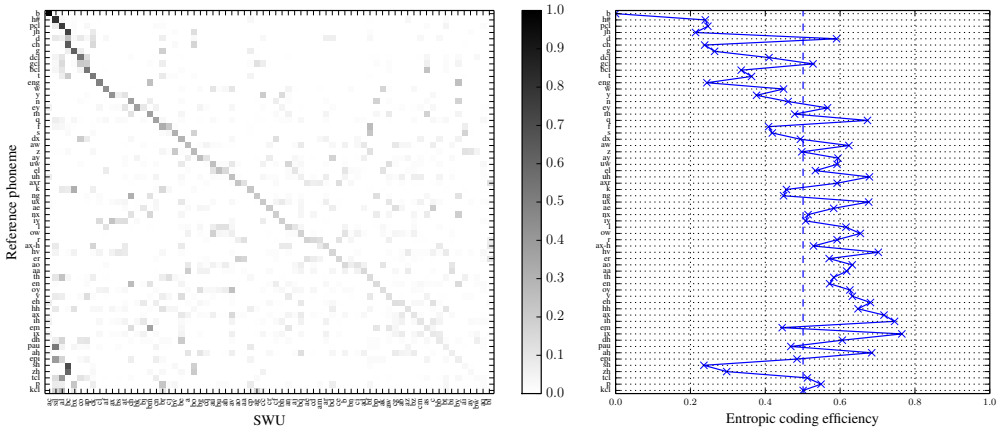
This work was supported by Telkom South Africa. Some computations were performed using the University of Stellenboschs Rhasatsha HPC.



(a) Baseline sparse coding SWU inventory



(b) After 40 iterations of sequence-level bigram constrained Viterbi training



(c) After 40 iterations of word-level constrained Viterbi training

Fig. 3. Comparison of phoneme-SWU coincidence matrices and their corresponding entropic coding efficiencies for experiments 1, 2, and 3 in Table 1. The dashed lines show the weighted mean coding efficiencies.

References

1. Agenbag, W., Niesler, T.R.. Automatic segmentation and clustering of speech using sparse coding and metaheuristic search. In: *Proceedings of Interspeech*. 2015, .
2. Grosse, R.B., Raina, R., Kwong, H., Ng, A.Y.. Shift-invariance sparse coding for audio classification. *CoRR* 2012;**abs/1206.5241**.
3. Smit, W., Barnard, E.. Continuous speech recognition with sparse coding. *Computer Speech & Language* 2009;**23**(2):200–219. doi:10.1016/j.csl.2008.06.002.
4. Sivaram, G.S.V.S., Nemala, S., Elhilali, M., Tran, T., Hermansky, H.. Sparse coding for speech recognition. In: *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. 2010, p. 4346–4349. doi:10.1109/ICASSP.2010.5495649.
5. Vinyals, O., Deng, L.. Are sparse representations rich enough for acoustic modeling? In: *Proceedings of Interspeech*. 2012, .
6. Goussard, G., Niesler, T.R.. Automatic discovery of subword units and pronunciations for automatic speech recognition using TIMIT. In: *Proceedings of the Annual Symposium of the Pattern Recognition Society of South Africa (PRASA)*. 2010, .
7. ten Bosch, L., Cranen, B.. A computational model for unsupervised word discovery. In: *Proceedings of Interspeech*. 2007, p. 1481–1484.
8. Lerato, L., Niesler, T.R.. Clustering acoustic segments using multi-stage agglomerative hierarchical clustering. *PLoS ONE* 2015; **10**(10):e0141756. doi:10.1371/journal.pone.0141756.
9. Bacchiani, M., Ostendorf, M.. Joint lexicon, acoustic unit inventory and model design. *Speech Communication* 1999;**29**(24):99 – 114. doi:http://dx.doi.org/10.1016/S0167-6393(99)00033-3.
10. Razavi, M., et al. An HMM-Based Formalism for Automatic Subword Unit Derivation and Pronunciation Generation. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, .
11. Torbati, A.H.H.N., Picone, J., Sobel, M.. Speech acoustic unit segmentation using hierarchical Dirichlet processes. In: *Proceedings of Interspeech*. 2013, p. 637–641.
12. Wang, H., Lee, T., Leung, C.C., Ma, B., Li, H.. Unsupervised mining of acoustic subword units with segment-level gaussian posteriorgrams. In: *Proceedings of Interspeech*. 2013, p. 2297–2301.
13. Lee, C.y., Zhang, Y., Glass, J.R.. Joint learning of phonetic units and word pronunciations for asr. In: *Proceedings of Empirical Methods on Natural Language Processing (EMNLP)*. 2013, p. 182–192.
14. Singh, R., Raj, B., Stern, R.. Automatic generation of subword units for speech recognition systems. *IEEE Transactions on Speech and Audio Processing* 2002;**10**(2):89–99. doi:10.1109/89.985546.
15. Young, S.J., Evermann, G., Gales, M.J.F., Hain, T., Kershaw, D., Moore, G., et al. The HTK book, version 3.4 2006;.