IDIAP RESEARCH REPORT

Silence Models in Weighted Finite-State Transducers

Philip N. Garner [a]

IDIAP–RR 08-19

April 2008

to appear in
Interspeech 2008

[a] IDIAP Research Institute

IDIAP Research Institute www.idiap.ch
Av. des Prés–Beudin 20 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 08-19

Silence Models in Weighted Finite-State Transducers

Philip N. Garner

April 2008

to appear in
Interspeech 2008

Abstract. We investigate the effects of different silence modelling strategies in Weighted Finite-State Transducers for Automatic Speech Recognition. We show that the choice of silence models, and the way they are included in the transducer, can have a significant effect on the size of the resulting transducer; we present a means to prevent particularly large silence overheads. Our conclusions include that context-free silence modelling fits well with transducer based grammars, whereas modelling silence as a monophone and a context has larger overheads.

Contents

1 Introduction

A recent trend in Automatic Speech Recognition (ASR) research has been to use decoders with precompiled grammars. Such grammars are generated using the Weighted Finite-State Transducer (WFST) methodology of Mohri *et al.* [1]. The advantage over traditional decoders is that various optimisations such as language model lookahead, prefixing and suffixing are subsumed into generic WFST operations such as composition and determinisation. This in turn can vastly reduce the complexity required in the decoder itself. The composition process typically deals with four transducers: A grammar, G, a lexicon, L, a context dependency graph, C and a Hidden Markov Model (HMM), H. The four transducers are composed into a single transducer in an operation that is normally written $H \circ C \circ L \circ G$. The decoder then (typically) only has to maximise the likelihood of a path through the combined transducer given acoustic observation likelihoods.

Juicer [2] is a WFST decoder developed at IDIAP. It is designed to handle both HMMs typically produced by HTK [3], and HMM/MLP hybrids where the observations are posterior probabilities. Juicer uses a type of WFST denoted $C \circ L \circ G$; that is, the HMM graph is handled in the decoder, and the WFST transduces from words to models (rather than to states or PDFs, as would a $H \circ C \circ L \circ G$ transducer). This type of transducer is described by Mohri *et al.* [4], where the authors state that the final transducer should have around 2.1 times the number of arcs as G for a bigram grammar; 2.5 times for a trigram.

This paper is motivated by our work on using Juicer in the AMI (Augmented Multi-party Interaction) system [5]. The AMI language model is typically a 50,000 word trigram, pruned to fit speed and memory constraints. In building even heavily pruned language model WFSTs, however, we were finding that the process was using several gigabytes of core memory and producing WFSTs significantly larger than predicted in [4]. Although some of the difficulties could be alleviated by careful tuning of the composition process, one significant problem turned out to be to do with silence modelling. Our investigation followed an initial observation that removal of the silence models resulted in almost a 50% reduction in the size of the final transducer.

In this paper, we discuss silence modelling in general and in the context of WFSTs. We show how different silence modelling strategies affect the size of the resulting WFSTs, and discuss implications for the decoder, and for the ASR system in general.

2 Silence modelling

2.1 General practice

Silence models are necessary in ASR to accommodate periods of silence at the beginning and end of utterances, and between words. Typically, any sound not included in the phone set of the decoder is included in the definition of silence; it might better be termed non-speech or noise. There may also be an element of garbage modelling in a silence model. The key, however, is the way the model is used. Silence is (trivially) placed at the beginning and end of the grammar, and can be included in G. Silence is also placed between words; the most convenient way to enable this is to duplicate lexicon entries such that each pronunciation has one unmodified phonetic spelling, and one either beginning or ending in silence.

The HTK system, as described by Young *et al.* [3], advocates the use of two silence models:

1. A silence model, `sil`, with the same structure as the other phonetic models, and contextually a monophone. i.e., silence acts as a context, but is context independent.

2. A short pause model, `sp`, that is essentially tied to the silence model, but is context free, and has a 'skip' transition that optionally omits any emitting states.

The silence model is used at the beginning and end of an utterance, or when prescribed in the grammar by a specific token. The short pause model is used in the lexicon at the end of every word to allow optional silence states that do not break context.

The use of the short pause model at the end of each word was revised by Hain *et al.* [6] to advocate the use of both short pause and silence at the word ends. This gives the option of either breaking context or not between words at decode time, and leads to a small improvement in recognition accuracy. In fact, Hain *et al.* distinguish the case where neither silence nor short pause are used, although this is included in the short pause skip structure.

2.2 Silence in WFSTs

Silence in WFSTs is discussed briefly by Allauzen *et al.* [7]. In that paper, silence is represented as a loop that can be placed at the end of each word in the lexicon. Further, it is stressed that the loop must be weighted to allow weight pushing in the composition process. The silence class transducer, figure 4 in [7], bears a close similarity to the short pause phone of the HTK system. Both allow zero or more instances of a silence state after each word, where the transition probabilities are learned from data.

It follows that including silence in the grammar is often not feasible as the grammar does not generally contain probability information for silence. However, including a short pause model after each word in the lexicon implements both the HTK short pause model as described, and has the same effect as the AT&T silence transducer. A silence model can be trivially included after each word in the lexicon in the same manner as the short pause model.

2.3 Juicer

For the common case of a three emitting state model, an HTK HMM actually has five states with the first and last being non-emitting. The short pause model is normally implemented with a 'skip' transition from the first state to the last, allowing it to be skipped completely. Juicer was designed to be compatible with HTK style HMMs. However, for simplicity in the decoder, skip transitions were not considered. Rather, a skip transition could easily be included in the WFST at the lexicon level.

The AMI system uses the double silence method of Hain *et al.* [6]. This was implemented in Juicer as shown in figure 1. The example is for just the word 'NO' in the lexicon WFST L. Notice that the other symbols are standard in the WFST literature: The `<eps>` symbol refers to an epsilon transition and `#1` is an auxiliary symbol to distinguish otherwise identical pronunciations.

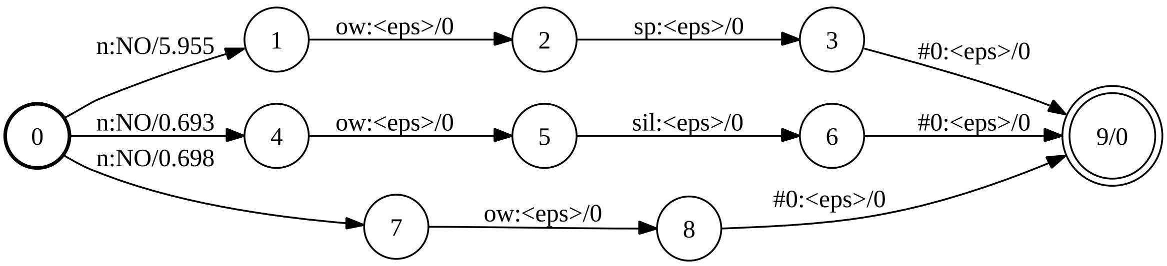

Figure 1: Three pronunciation options accommodating silence for a single word in the lexicon.

3 Context dependency

3.1 Background

The context dependency transducer, C, maps context independent phones (triphones in our case) to context dependent phones. Although it is not normally associated with silence modelling, it is pertinent here because the short pause and silence phones are handled by different parts of C.

3.2 Construction of context transducer

Context dependency is discussed by Riley *et al.* [8]. Note that the transducer in figure 4 of that paper is synchronous, i.e., the input phone corresponds to the centre phone of the output triphone. It also requires determinisation.

By contrast, the algorithm used for generation of C in Juicer does not require determinisation. Conceptually, it is identical to the way that a trigram back-off language model would be represented, and is illustrated in figure 2. The central vertex labeled 'sil' (silence phone) represents a 'context independent' state. The label is only to aid understanding; it means that that vertex encodes that contextual state. All transitions leaving this state output the context independent label 'sil' and lead to states on a first concentric circle. All transitions leaving the first concentric circle either 'back-off' to the context independent state, or lead to context dependent states on another (outer) concentric circle. In general, sequences of context dependent phones are matched by moving around the outer concentric circle only. This figure is a compromise between readability and completeness; many of these outer circle transitions are not shown.

3.3 Short pause in context dependency

The context dependent input labels are delayed by one phone w.r.t. their corresponding output labels. This is of no concern normally as WFST composition tends to push labels out of synchronisation.

However, the synchronisation issue causes a slight difficulty in the insertion of the short pause phone into C. The solution is to insert an extra parallel path for each triphone that also delays the short pause. This is illustrated in figure 3.

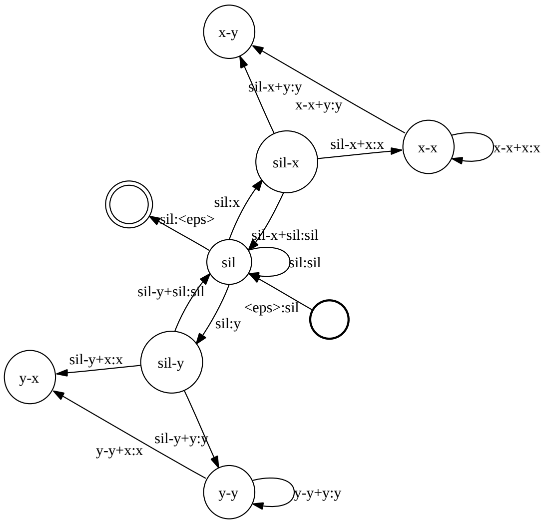

Figure 2: Skeleton of a context dependency WFST. Note that many transitions from the outer states are not drawn.

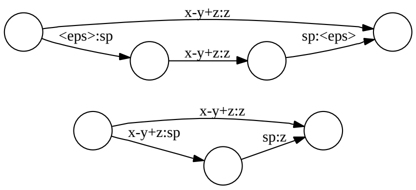

Figure 3: Above: Insertion of sp into context dependent transition in C. Below: slightly more compact form.

The insertion of short pause clearly increases the size and complexity of C, but this complexity is not necessarily passed on to the final WFST. Notice that short pause and silence follow different paths in C, so their effects in the size of $C \circ L \circ G$ are additive.

4 Quantitative evaluation

4.1 Transducer size

We had already observed that removing the silence models completely from the WFST resulted in a significant size reduction. In order to quantify this further, experiments were done to test the effects of various silence model strategies. The tests were done on the well known Wall Street Journal (WSJ) 20,000 word task, identical to that used by Moore *et al.* [2]. The language model, G, was the bigram supplied with the task; it had 1,473,622 transitions. The WFST construction was done using the AT&T toolkit [9], although we have verified that the MIT toolkit [10] gives similar results. The acoustic model for which the context dependency transducer was built had 25,869 distinct models.

During composition of $C \circ L \circ G$, auxiliary symbols were removed after minimisation of $L \circ G$, producing a non-determinisable $C \circ L \circ G$. It is also possible to leave in the auxiliary symbols, allowing $C \circ L \circ G$ to be minimised. However, we find that the latter method produces larger final transducers.

The size tests comprised all reasonable combinations of silence, short pause and skip transitions. In addition, we evaluated the effect of placing the silence models at the beginning of the word in L, as

Table 1: WFST sizes for various silence model configurations in the 20,000 word WSJ bigram task. The grammar transducer, G, has 1,473,622 arcs.

	L $\times 10^3$ arcs	$L \circ G$ $\times 10^3$ arcs	$C \circ L \circ G$ $\times 10^3$ arcs	Silence overhead
No silence model	146	2,434	2,565	0.0%
Short pause + skip in WFST	313	2,693	3,437	34.0%
Silence + skip in WFST	313	2,693	3,869	50.8%
Short pause + silence + skip in WFST	479	2,904	4,762	85.6%
Short pause only	167	2,466	2,801	9.2%
Short pause + silence	333	2,677	4,105	60.0%
Short pause at start	167	2,454	2,815	9.7%
Short pause + silence at start	333	2,754	3,810	48.5%

shown in figure 4. As back-off contexts fan-out to all words in the vocabulary, whereas word contexts

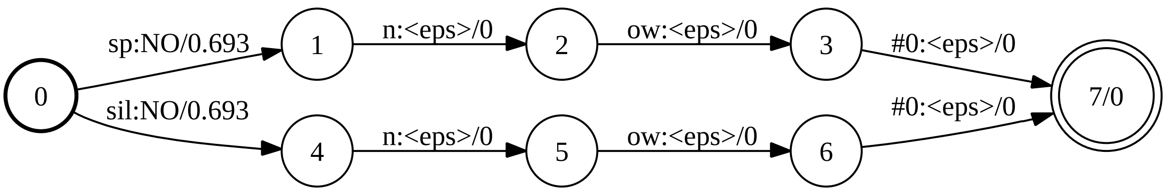

Figure 4: Lexicon fragment for the word 'NO' with silence elements at the start of the word.

only fan out to observed n-grams, we hypothesised that such silence placement could lead to more sharing of arcs.

Table 1 shows the effect of the silence model strategy on the WFST size. In particular, the first line shows the size with no silence at all in L. The silence overhead is then a percentage measure in excess of this baseline for each $C \circ L \circ G$ WFST. The results show some variation around the sizes predicted in Mohri *et al.* [4].

The most striking result, the one that prompted the investigation, is that our original configuration of silence, short pause and the skip transition in the WFST causes a disproportionately large silence overhead. Conversely, the inclusion of only a short pause introduces a rather minor silence overhead, disproportionately smaller than one might expect from the increased complexity in the context dependency WFST.

The final two lines in the table show the effect of placing the silence arcs at the beginning instead of the end of each word in the lexicon. As was hypothesised, this produces a significant saving when the silence monophone is used, with a reduction in silence overhead of about 10%.

4.2 Decoder modification

There are large differences in the sizes of WFSTs that include and do not include the skip transition. This raises the question of whether the inclusion of the skip as a special case in the decoder is worthwhile. It turns out that such a skip is not as onerous as might be imagined.

A WFST used for ASR generally contains epsilon arcs—transitions in the grammar without associated labels. When consulting the WFST for successive arcs, a decoder must traverse any arcs with input epsilons—arcs that have no associated model. It follows that the skip in an HMM can be handled within the decoder by treating it as an input epsilon arc.

In order to evaluate whether the combination of smaller WFST and marginally more complex decoder had any performance impact, Juicer was modified to handle skip transitions as epsilon arcs as described above. We were then able to run experiments on the three WFST configurations for which

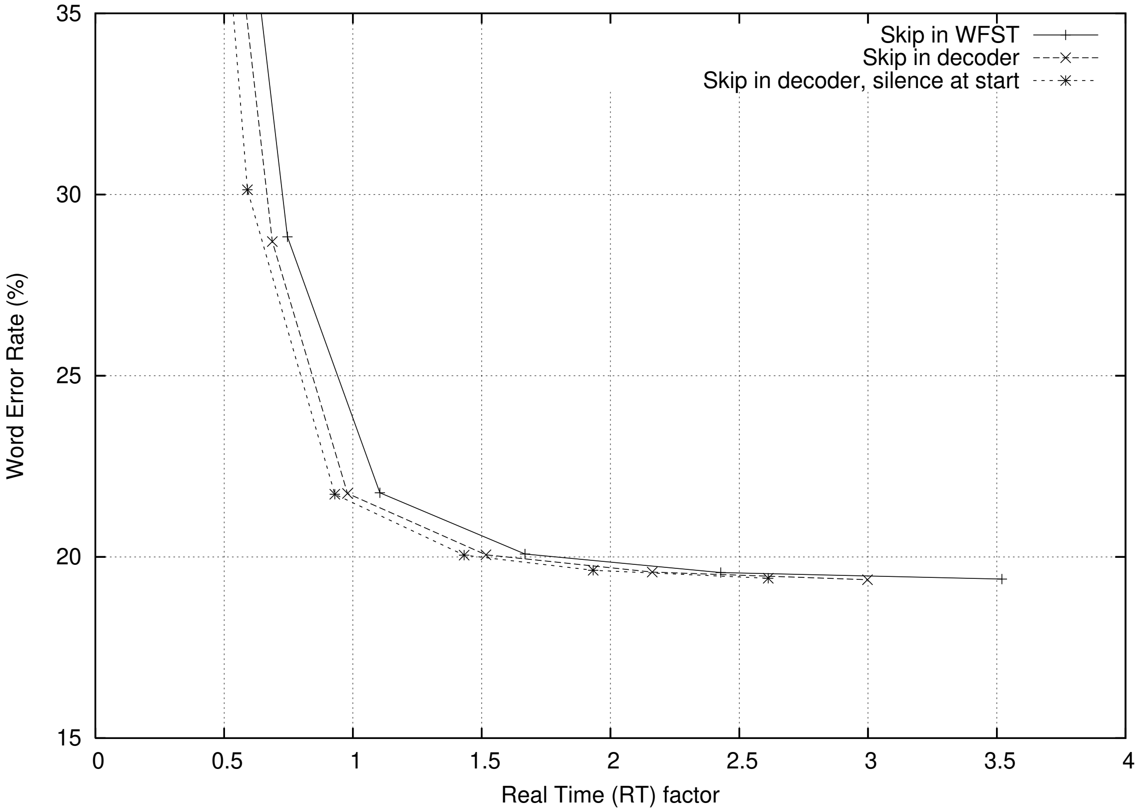

Figure 5: Beam pruning profile for three silence strategies.

our acoustic model was trained—those with both a silence and short pause associated with each word. The acoustic model was a rather standard Gaussian mixture HMM trained on 39 dimensional HTK PLP features. The WSJ training data had been aligned using the same double silence model strategy. The 503 utterance speaker independent test set was used for evaluation. The results are illustrated in figure 5. The first plot is the baseline with the skip transition in the WFST. The second plot shows the effect of moving the skip into the decoder; there is a small but consistent speed improvement. The third plot is like the second except that the silence and short pause models are at the start of each word in L. Again, there is a marginal speed improvement in doing this. All plots asymptote to the same minimum error rate.

4.3 Discussion

Table 2 shows the same silence overheads as above, but for the AMI trigram language model. This model is significantly larger than the WSJ bigram, with close to 5,000,000 arcs. The lexicon is increased to 50,000 words, and C is compiled with 21,250 distinct context dependent models (slightly smaller than that for WSJ). The overheads for the trigram, whilst following the general trend of the bigram model, are larger. In particular, the largest of these, our original configuration, required more than 4 GB of memory to compose.

The large size associated with skip transitions in the WFST is probably associated with epsilon removal. When an epsilon arc is removed, the fan-out from the end of the epsilon arc can be duplicated at the beginning of the arc. The same effect causes the networks to increase in size when auxiliary symbols on back-off arcs are removed.

With hindsight, we hypothesise that the use of an explicit skip arc in the lexicon with an auxilliary

Table 2: WFST sizes for various silence model configurations in the 50,000 word AMI trigram task. The grammar transducer, G, had 4,975,958 arcs.

	L $\times 10^3$ arcs	$L \circ G$ $\times 10^3$ arcs	$C \circ L \circ G$ $\times 10^3$ arcs	Silence overhead
No silence model	381	6,732	7,411	0.0%
Short pause + skip in WFST	814	7,941	11,598	56.5%
Silence + skip in WFST	814	7,941	12,215	64.8%
Short pause + silence + skip in WFST	1,246	8,688	16,607	124.1%
Short pause only	432	7,151	8,642	16.6%
Short pause + silence	864	7,898	13,469	81.7%
Short pause at start	432	6,970	8,495	14.6%
Short pause + silence at start	864	7,862	11,929	61.0%

symbol to prevent its removal might reduce the overhead associated with the skip arcs. The composition process would keep such arcs with the short pause or silence arcs leading to an overhead similar to that of short pause or silence. We have not attempted to verify this as it was not necessary for our twin silence strategy. However, such an approach may be necessary if only context-independent silence models were used.

5 Conclusions

We have shown that the silence model strategy can have a significant effect on the size of $C \circ L \circ G$ level WFSTs. A silence skip transition encoded in the WFST can lead to a particularly large WFST. Conversely, silence encoded only as a context free short pause can have a particularly small overhead. The size of the WFST can be significantly reduced by treating the skip as a special case in the decoder.

The potential reduction in silence overhead is significant in both WFST construction and ASR. In the former, it can make a difference between requiring 32 bit instead of 64 bit computer architecture. In the latter, it can allow a grammar with perhaps 50% more complexity for a given memory footprint.

The memory reductions also lead to small improvements in decoding speed.

6 Acknowledgements

Juicer and its associated tools, including the deterministic context dependency transducer and delayed short pause model, were written by Darren Moore. The author is grateful to John Dines for many useful discussions.

This work was supported by the European Union 6^{th} FWP IST Integrated Project AMIDA (Augmented Multi-party Interaction with Distance Access, FP6-033812) and the Swiss National Center of Competence in Research (NCCR) on Interactive Multi-modal Information Management (IM2).

References

[1] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, 2002.

[2] Darren Moore, John Dines, Mathew Magimai Doss, Jithendra Vepa, Octavian Cheng, and Thomas Hain. Juicer: A weighted finite-state transducer speech decoder. In *Proceedings of the 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, 2006.

[3] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Lui, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book.* Cambridge University Engineering Department, version 3.4 edition, December 2006.

[4] Mehryar Mohri, Michael Riley, Don Hindle, Andrej Ljolje, and Fernando Pereira. Full expansion of context dependent networks in large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.

[5] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan. The AMI meeting transcription system: Progress and performance. In *Proceedings of the NIST RT06 Spring Workshop*, 2006.

[6] T. Hain, P. C. Woodland, G. Evermann, and D. Povey. The CU-HTK March 2000 hub5e transcription system. In *Proceedings of the Speech Transcription Workshop*, 2000. College Park.

[7] Cyril Allauzen, Mehryar Mohri, Michael Riley, and Brian Roark. A generalized construction of integrated speech recognition transducers. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.

[8] Michael Riley, Fernando Pereira, and Mehryar Mohri. Transducer composition for context-dependent network expansion. In *Proceedings of EUROSPEECH*, 1997. Rhodes, Greece.

[9] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, (231):17–32, January 2000.

[10] Lee Hetherington. The MIT finite-state transducer toolkit for speech and language processing. In *Proceedings of the International Conference on Spoken Language Processing*, 2004.