Weighted Finite-State Transducers

Alexander Seward
Centre for Speech Technology
KTH
Stockholm, Sweden

Why Weighted Finite-State Transducers?

1. Efficiency and Generality of Classical Automata Algorithms

Efficient algorithms for a variety of problems (e.g. string-matching, compilers, ,parsing, pattern matching, process industri, design of controllability systems in aircrafts).

General algorithms: rational operations, optimizations.

2. Weights
Handling uncertainty: text, handwritten text, speech, image, biological sequences.
Increased generality: finite-state transducers, multiplicity/indeterminism.

3. Applications
Text: pattern-matching, indexation, compression.
Speech: Large-vocabulary speech recognition, speech synthesis.
Image: image compression, filters.

* credits to M.Mohri

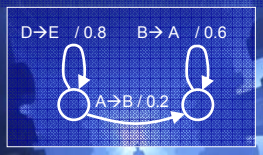
TRANSDUCERS
IN AUTOMATIC SPEECH RECOGNITION

In ASR: Mathematical models for speech-to-text translation

A uniform composition of different information sources:
HMM-data, lexica, language models, etc..

Flexible: reduces decoder depedencies,
multiple layers,
generic optimization methods.

What is a Weighted Finite-State Transducer (WFST) ?



A finite-state machine where each arc is a weighted transduction consisting of an input, an output, and a probability/weight

Simply put: ***A translation device***

A WFSA is a transducer without output

WFSTs in recognition

- I want a ticke..#noise"..Boston from New York

FST trained on acoustics and language corpus:
#noise" must be "t...to" !

WFSTs in recognition

The bare was bear naked?

- FST trained on language corpus:
- The bear was bare naked

Recognition Cascade (simplified)

- I: Input feature vectors
- H: HMM
- C: Context-Dependency Model
- L: Lexicon
- G: Grammars

Feature vectors

Feature vectors —w→ *CD-HMMs*

CD-HMMs —w→ *Transcr. syms*

Transcr. syms —w→ *Words*

Words —w→ *Words*

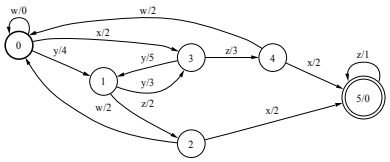
Use Weighted FST Composition
to compose the parts into one

Weighted FST operations

Best-path	Difference	Weight pushing
Closure	Equivalence	Label pushing
Compaction	Hadamard product	Reversal
Composition	Inversion	Epsilon removal
Concatenation	Minimization	Topological sort
Connection	Projection	Union
Determinization	Pruning	

Language model WFSA

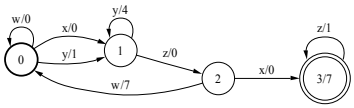
Model a priori weights for different word sequences (n-grams)



4 fictitious words: *w, x, y, z*

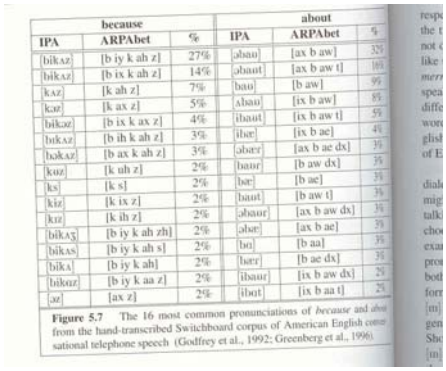
LM WFSA - Minimized

Model a priori weights to different word sequences (n-grams)



4 fictitious words: *w, x, y, z*

Pronunciation knowledge



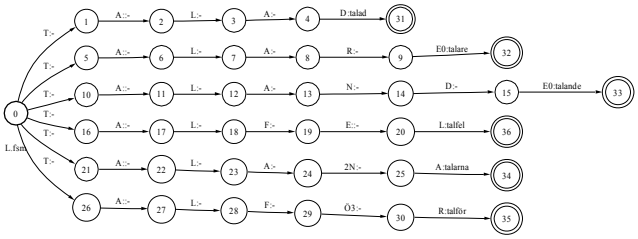
because			about		
IPA	ARPAbet	%	IPA	ARPAbet	%
[bikəz]	[b iy k ah z]	27%	[əbaʊ]	[ax b aw]	32%
[bikʌz]	[b ix k ah z]	14%	[əbaʊt]	[ax b aw t]	16%
[kəz]	[k ah z]	7%	[baʊ]	[b aw]	9%
[kəz]	[k ax z]	5%	[ʌbaʊ]	[ix b aw]	8%
[bikəz]	[b ix k ax z]	4%	[ibaʊt]	[ix b aw t]	5%
[bikʌz]	[b ih k ah z]	3%	[ibɛ]	[ix b ae]	4%
[bokʌz]	[b ax k ah z]	3%	[ʌbɛr]	[ax b ae dx]	3%
[kʊz]	[k uh z]	2%	[baʊr]	[b aw dx]	3%
[ks]	[k s]	2%	[bɛ]	[b ae]	3%
[kiz]	[k ix z]	2%	[baʊt]	[b aw t]	3%
[kɪz]	[k ih z]	2%	[əbaʊr]	[ax b aw dx]	3%
[bikʌʒ]	[b iy k ah zh]	2%	[ʌbɛ]	[ax b ae]	3%
[bikʌs]	[b iy k ah s]	2%	[bɑ]	[b aa]	3%
[bikʌ]	[b iy k ah]	2%	[bɛr]	[b ae dx]	3%
[bikʊz]	[b iy k aa z]	2%	[ibaʊr]	[ix b aw dx]	2%
[ʌz]	[ax z]	2%	[ibʊt]	[ix b aa t]	2%

Figure 5.7 The 16 most common pronunciations of *because* and *about* from the hand-transcribed Switchboard corpus of American English conversational telephone speech (Godfrey et al., 1992; Greenberg et al., 1996).

Use different weights to model likelihood of pronuciations!

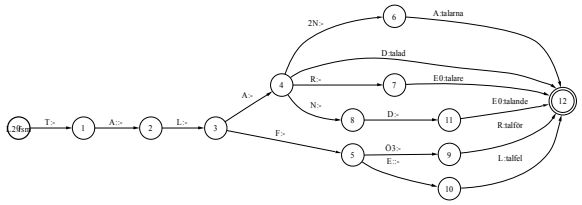
Lexicon transducer

Some phonetically similar words



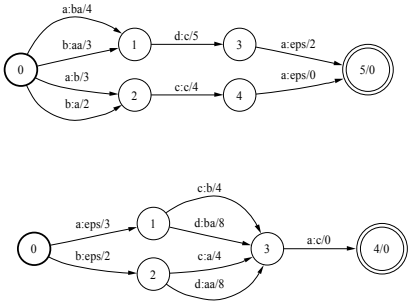
36 states

Euqivalent lexicon transducer – Deterministic

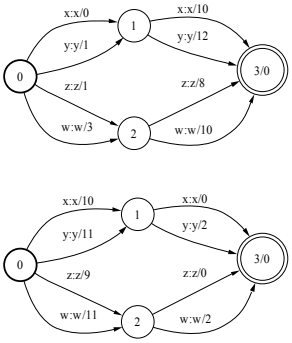


36 → 13 states

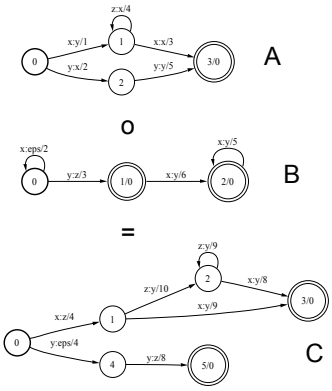
Weighted determinization



Weight pushing



Weighted composition



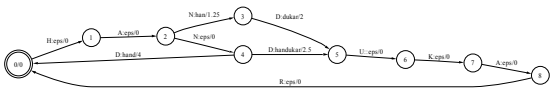
A: (x, z, z, x) → (y, x, x, x)
weight: 1 + 4 + 4 + 3 = 12.

B: (y, x, x, x) → (z, y, y, y)
weight: 3 + 6 + 5 + 5 = 19.

A ○ B: (x, z, z, x) → (z, y, y, y)
total weight: 12 + 19 = 31.

C = A ○ B: (x, z, z, x) → (z, y, y, y)
same total weight: 4 + 10 + 9 + 8 = 31.

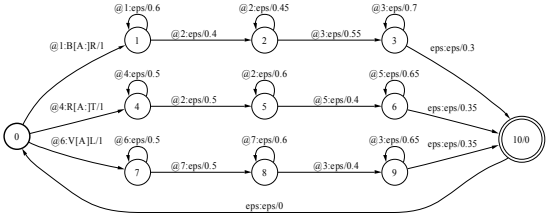
Lexicon + Grammar: L ○ G



han	H A N	(Eng: he)
hand	H A: N D	(Eng: hand)
handdukar	H A: N D U: K A R	(Eng: towels)
dukar	D U: K A R	(Eng: set the table)

Sequence:
han dukar H A: N D U: K A R (Eng: he sets the table)

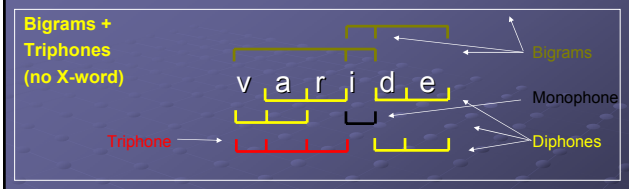
HMMs as WFSTs



Context-Dependendency Modeling

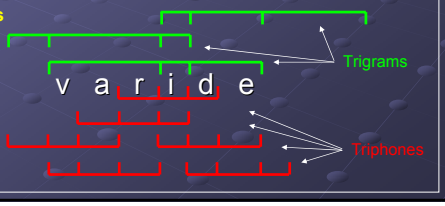
"…var i det…"

Bigrams + Triphones (no X-word)



Trigrams + X-word Triphones

r [i] d
a [r] i
x [v] a , i[d]e
v[a]r , d[e]x



ASR systems

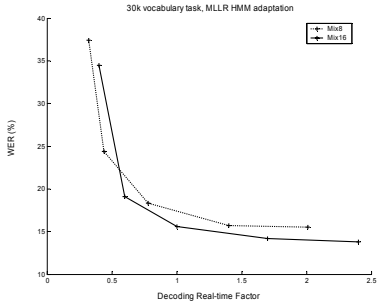
System	Starlite (N.Ström)	Ace (A.Seward)	Daytona (A.Seward)
Type	MVCSR (< 2kw)	MVCSR (< 5kw)	LVCSR (>30 kw)
Decoder tech	2-pass, Viterbi + A*	1-pass, Token passing (mod)	1-pass, WFST
Source code	C	Java	C# /C/SIMD Asm
AM	HMM+GM, ANN	HMM+GM	HMM+GM
Compatible toolkits	HTK	HTK	HTK, SRI LM, AT&T FSM
Bigrams	Class-pair		✓
Trigrams			✓
Multi-level N-gram (backoff)			✓
Context-Free Grammars		✓	✓
Dynamic LMs		✓	✓
Incremental results		✓	✓
X-word N-phone decoding			✓
Dictation (infin. input)			✓

Daytona LVCSR system

- Built for integration in online systems
  - real-time performance
  - time-synchronous decoding
- Based on Weighted Finite-State Transducers
- Incremental output (word by word)

Large vocabulary CSR (30k task)

Backoff 4-gram LM, triphones (cross-word)



Total FST search space: (language model, lexicon, context-dependencies, HMMs etc.)

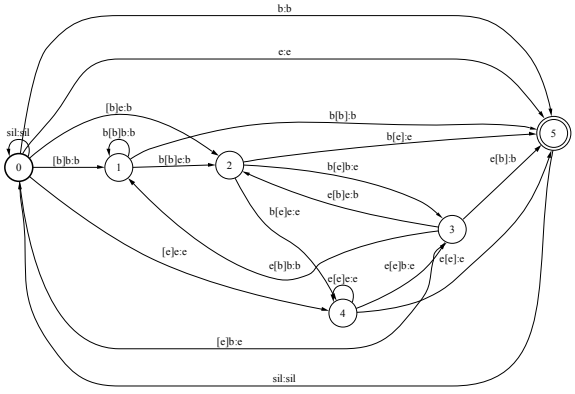
space ~ 10^8-10^9 states

Without FST optimizations ~ 10^9-10^{11} states

End

Super-miniature decoder: Transducer C' (only b, e + sil)

CD-phones (8 triphones, 4 diphones, 3 monophones) → transcr symbols

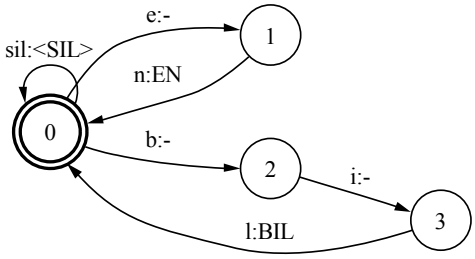


A miniature two word decoder example

- Lexicon:
  - bil : '*b*' '*i*' '*l*'
  - en : '*e*' '*n*'
  - <SIL> : '*sil*'
- 6 transcription symbols (5+1)
  - b e i l n (context dependent)
  - sil (context independent)

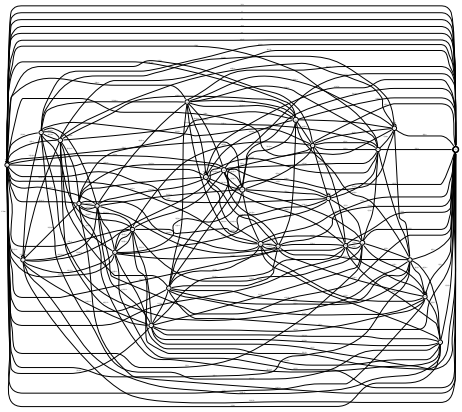
  AM: 125 (5 ^{3}) triphones, 25(5 ^{2}) diphones, 6 monophones (5+1)
- Simple bigram language model

Miniature decoder: Transducer L

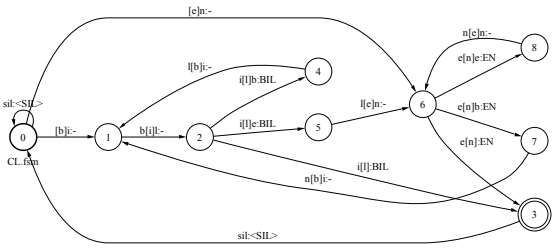


Miniature decoder: Transducer C

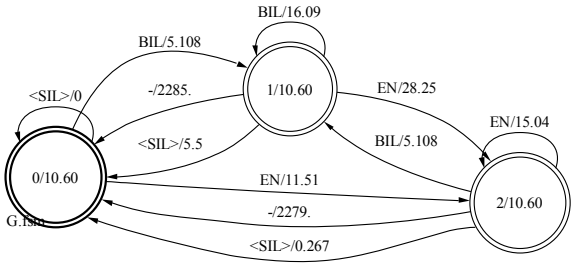
CD-phones (125 triphones, 25 diphones, 6 monophones) → transcr symbols



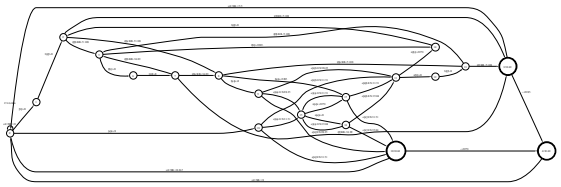
Transducer C ○ L



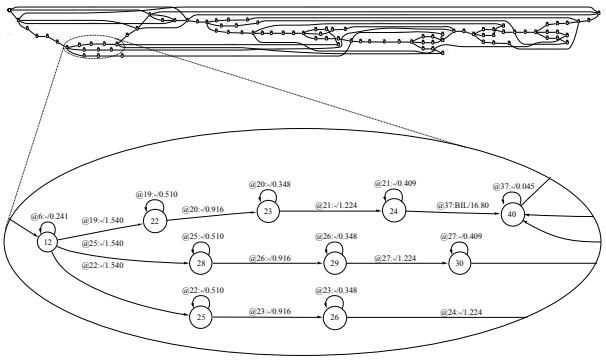
Transducer G



Transducer C ○ L ○ G



Optimized WFST H ○ C ○ L ○ G



A weighted transducer PDFs → words

Speech Recognition Grammar Specification

W3C Recommendation

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE grammar PUBLIC "-//W3C//DTD GRAMMAR 1.0//EN"
                  "http://www.w3.org/TR/speech-grammar/grammar.dtd">

<grammar xmlns="http://www.w3.org/2001/06/grammar"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.w3.org/2001/06/grammar
                             http://www.w3.org/TR/speech-grammar/grammar.xsd"
         xml:lang="en" version="1.0" root="city_state" mode="voice">

   <rule id="city" scope="public">
     <one-of>
       <item>Boston</item>
       <item>Philadelphia</item>
       <item>Fargo</item>
     </one-of>
   </rule>

   <rule id="state" scope="public">
     <one-of>
       <item>Florida</item>
       <item>North Dakota</item>
       <item>New York</item>
     </one-of>
   </rule>

   <!-- Reference by URI to a local rule -->
   <!-- Artificial example allows "Boston, Florida"! -->
   <rule id="city_state" scope="public">
     <ruleref uri="#city"/> <ruleref uri="#state"/>
   </rule>
</grammar>
```