3-2 SprinTra WFST 音声デコーダ開発について

3-2 Development of the SprinTra WFST Speech Decoder

ポール・R・ディクソン　堀　智織　柏岡秀樹
DIXON Paul Richard, HORI Chiori, and KASHIOKA Hideki

要旨

本稿では、NICT の「重み付き有限状態トランスデューサ（WFST)」による音声デコーダ、名付けて SprinTra について述べる。本稿はまず、様々な WFST とそれに付随する数学的記述について簡単に紹介する。続いて音声認識における WFST の使用方法について述べ、典型的な音声認識システムで使用される WFST 要素の紹介、そして効果的なデコーディング探索領域を捻出する為の WFST 結合化、最適化について述べる。その後に SprinTra の特性と構成図についての高レベル詳細を述べる。注目点は探索機能、処理機能に適したエンジンを設計、実装するという点である。最新の音声認識技術をできるだけ多くのユーザーに提供するため、SprinTra は様々なプラットフォームで動作することができ、さらに異なるプログラミングインターフェースやスクリプト階層にアクセスする事ができる。また支援ツールは確率性（probability）や使い易さを元に設計されている。この事により音声認識の専門家でなくても最新式の音声認識システムを構築する事が可能となる。SprinTra の主な特性についての説明として我々が提唱している on-the-fly アルゴリズムがクラス N-gram 合成スキームによるメモリの効果的合成を可能にする、ということを概説する。

In this article we describe the NICT Weighted finite state transducer (WFST) based speech decoder named SprinTra. The paper starts with a brief introduction to WFSTs and the accompanying mathematical notation. This is followed by an introduction to the use of WFSTs in speech recognition, here give a brief description of the WFST components used in a typical speech recognition system, and explain how they are combined and optimized to yield very efficient decoder search spaces. After describing these preliminaries we move on to a high level description of the features and architecture of SprinTra. Our focus was to design and implement an engine suitable for research and deployment usage. To bring the state-of-the art speech recognition technology to as many users as possible SprinTra can run on many platforms and be additionally accessed through various programming interfaces and scripting layers. The supporting tools are also designed with portability and good usability, and this allow users and non-speech recognition experts to easily construct state-of-the-art speech recognition systems. The description of SprinTra's core features includes a description our on-the-fly algorithm we have proposed to allow for memory efficient composition of class N-gram models.

[キーワード]
重み付き有限状態トランスデューサ，音声認識，デコーダ，on-the-fly 合成
WFST, Speech recognition, Decoder, On-the-fly composition

1　はじめに

本稿では、現在 NICT で開発中であり最先端技術の音声デコーダ「SprinTra」を紹介する。この「SprinTra」の開発の主な目的は以下の通りである。

・新しいアイデアを実現し拡げる為の最先端研究プラットフォームを提供できるような音

声認識エンジンを作ること。

・VoiceTra のような実用サービスに使用され、単体エンジンとして認可されるような、商用目的に応じたエンジンを開発すること。

エンジンは「重み付き有限状態トランスデューサ」(Weighted Finite State Transducer: WFST)[1]を用いて動作するように設計され、WFST とは不安定さを表す重みを持たせた記号列をマッピングできる、有限状態マシンのことである。近年、音声認識において WFST が広く利用されるようになり、その主な利便性の 1 つとして個々のモデルを最適化させ合成させるという統合性にある。さらにデコーディングに先立ち最適化することにより、音声認識エンジンの開発が進み、従来の動的デコーダと比較した際よりスピードのある認識ができるようになる[2]。しかしながら、この統合性アプローチにはいくつかの欠点がある。1 つは、認識時の合成（探索）ネットワークを保持するための大量のメモリーが必要とされ、合成(composition)、最適化する際のオフラインのメモリ使用量がとてつもなく大きいということである。一旦情報ソースへの探索アクセスが切られると、オンライン化させることは非常に困難となる。この問題に対処するために、さまざまな on-the-fly 合成アプローチが提案されてきた[3]-[12]。

本稿では、N-gram 言語モデル[13]を用いて、メモリを有効に使用するため 3 方向の合成種類にわけて開発した、特定のアルゴリズムを説明する。また本稿は以下のように構成される。**2** では、WFST の説明とそれに使われる記号について述べる。**3** では、簡単に音声言語で使用されている WFST の説明をする。**4** では、SprinTra の特徴と構成図について詳細を述べる。その後、メモリ効率のよい N-gram 合成スキームについて説明する。本稿は、**5** において結論を述べる。

2 重み付き有限状態トランスデューサ

ここでは本稿で後に紹介するアルゴリズムの説明に必要な論理的基礎となる、WFST の概要について簡単に述べる。音声認識における WFST のさらに詳細な説明については[1][14][15]を参照のこと。

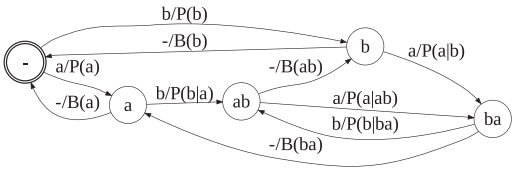

図 1 WFST の 1 つである簡単なバックオフ N-gram 言語モデル

WFST とは、有限オートマータの汎用型であり、それぞれの遷移が出力記号と重みの付いた入力記号をもつものである。正式にはトランスデューサ T は 8 要素として定義される[14][15]。

$$T = (\Sigma,\ \Delta,\ Q,\ I,\ F,\ E,\ \lambda,\ \rho) \qquad (1)$$

それぞれは、

・Σ 有限入力アルファベット。
・Δ i 有限出力アルファベット。
・Q 状態の有限セット。
・$I \subseteq Q$ 開始状態の集合。
・$F \subseteq Q$ 最終状態の集合。
・$E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \times \mathbb{K} \times Q$ 状態遷移の有限集合。
・λ： $I \rightarrow \mathbb{K}$ 初期の重み関数。
・ρ： $F \rightarrow \mathbb{K}$ 最終重み関数。

3 音声認識のための WFST

ここで我々は[14]にて述べる構築スキーム（scheme）を使用する。認識キャスケード（cascade）は次の要素でできている。認識文法規則を表す言語モデル G、発音辞書から作られ音素列を単語に変換（map）するレキシコンの L、文脈依存音素を文脈非依存音素に変換するためのトランスデューサ C、である。WFST の基本的概念を知るために、簡単な N-gram 言語モデル G を図 1 にて紹介する。この図では、アーク（arc）記号は記号もしくは重みを示す。WFST の N-gram 言語モデルを効果的に示すために、それぞれの状態は N-gram 履歴を示すのに使われ、N-gram かバックオフ確率（probability）かを表している。例えば、$a/P(a|b)$ のアークはバイグラム確率（probability）を与える。入力記号 a は我々が入力テープから a を使用したことを示し、

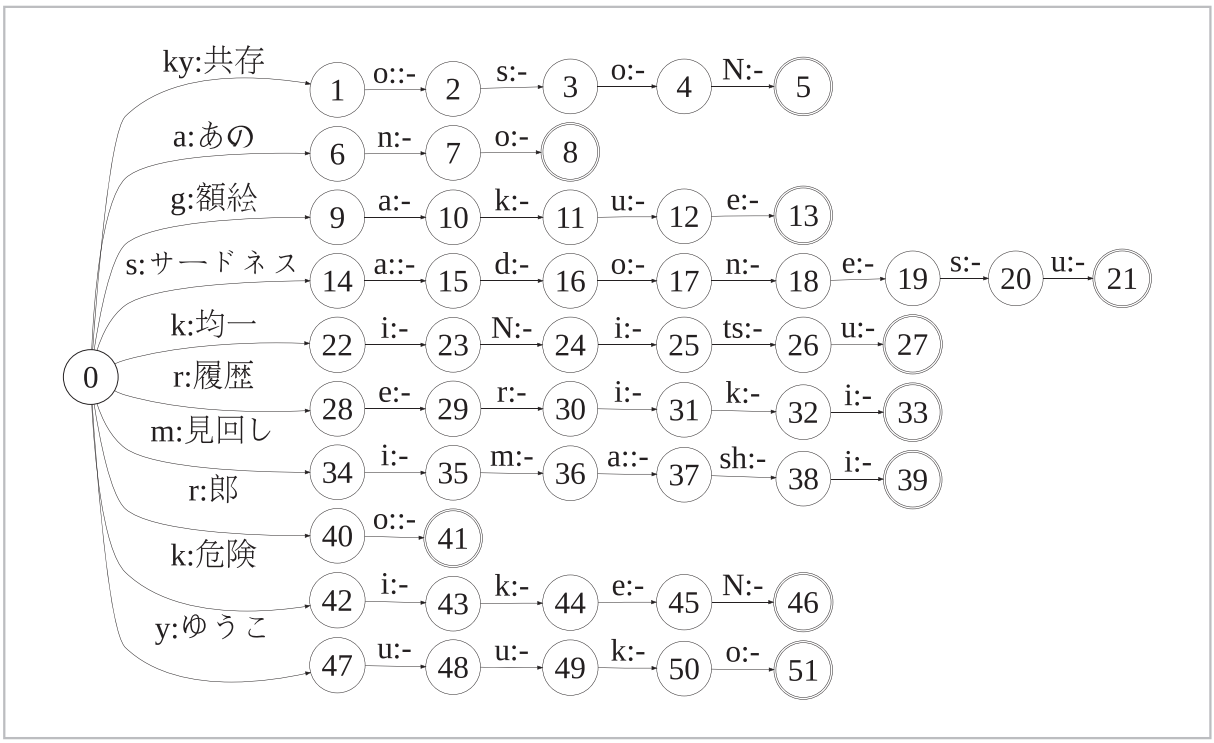

図2 簡単なレキシコントランスデューサ

ソース状態を示す、B は現在のユニグラム履歴を表す。また最終状態記号 *ba* は新たな履歴を作る。High order の N-gram 遷移が現在のシンボルに合わない時は、[16] に見られるようにエプシロン遷移としてバックオフ確率（probability）をエンコード（encode）する。これらのエプシロンアークは入力も使わず、バックオフ遷移だけで order N-gram 履歴を下げて、現在の入力シンボルが合うようにできる。G、L、そして C のトランスデューサは次のように演算される。

$$\pi\left(C \circ min(det(L \circ G))\right)$$

det が決定化するもの、*min* は最小化するもの、$\circ$ は合成化するものとなる。π は補助シンボルを除く演算である。この決定化演算は接頭辞（prefix）sharing 演算と同じであり、また最小化演算は tail sharing 演算と同じである。これらの演算を図式化するため、最小単位の音声認識レキシコン L を図2に示す。ここではそれぞれの単語は、アークと状態を線上（chain）につないだものであり、それぞれの線の入力側は、音素列であり、単語出力シンボルに変換するものである。ここで示すレキシコンは1組の線上（chain）でできる組み合わせとして出来上がったものである。図3では、決定化と最小化の効果例が示されており、共通の接頭辞（prefix）と接尾辞（suffix）がより効果的な、WFST として使用されていることがわかる。ただしそれらは同義ではない。

4　エンジン特性についての概要

4.1　コア特性

コア特性（core feature）には、近年の認識エンジンから期待される機能全てが含まれる。

- 入力（Input）—音声信号特性抽出用のフロントエンド。ファイルやネットワークソケットからの前計算処理された特性を使用する。
- 出力（output）—SprinTra は 1-best の認識結果または多くの代替（alternative）音声仮説を表すラティスを出力する。ラティスからは、N-best リストや混乱（confusion）ネットワークなどの有効な情報が抽出できる。
- フレキシビリティ（Flexibility）—フレキシビリティを持つ為に、我々は静的探索ネッ

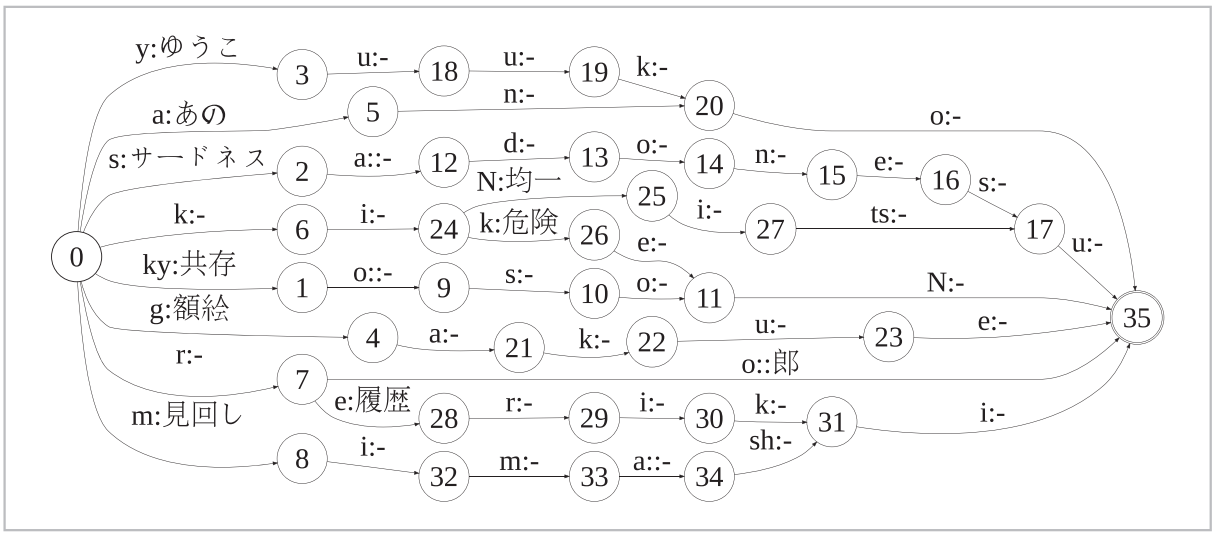

図３ 決定化と最小化の等価レキシコン

トワークを用いて演算し、言語を動的に合成する、もしくは特定化したクラスモデル合成アルゴリズムを使用する。

4.2 Portability

SprinTra で最も必要とされるものの１つとして、できるだけ多くのプラットフォームに適合する技術を持たせるという事がある。これを達成するために我々はバイナリライブラリ依存をやめ、デコーダは次のように走らせる事ができるようにした。

- デコーダは、コマンドラインアプリケーションバッチモード（command line application batch mode）処理として走らせることができ、またオフライン探索設定でも使うことができる。
- 我々はまた Windows でも Unix でもいずれのプラットフォームでも使用できるような開発レベルのプログラミングインターフェースを提供する。
- 最終的に高速処理のサーバーアプリケーションでも使用できる高レベルのパイソンで書かれた（python scripting）インターフェースを提供する。

4.3 支援ツール

統合された合成 WFST を構築することは、多くの技術と経験を必要とする。特にある種の合成と最適化演算は延々と続き、巨大な量のメモリマシンを消費することになる。SprinTra ツールボックスは、いくつかの統合されたビルドツールでできており、生成モデル（raw model）から WFST へと変換させることが可能である。このツールは、アプリケーションの仕様に応じて必要な合成、最適化過程を施す。このツールを使用することによりドメインを問わない（non-domain）専門家や開発者が高速で安全に高性能な WFST 音声認識システムを構築することが可能となる。最終探索ネットワークを用いて大きな効果を得るための、選択肢はたくさんある。我々は１つのコマンドにより我々の経験を使用してもっとも優れた最適化を選ぶこととする。デコーダと同じように、ビルドツールは移動可能で全てのプラットフォーム上で動作する。現在の NICT のシステム構造とスムーズに統合するために、我々は HTK [17] と ATRASR 音響モデル、ARPA フォーマット言語などの標準かつ伝統的フォーマットをサポートした。WFST を表現するために、ATT テキストフォーマットと OpenFst バイナリフォーマットを使用した。

4.4 融合された（Fused）合成アルゴリズム

クラス N-gram モデルは２つのトランスデューサがある。クラス記号 G の N-gram モデルとクラス記号から単語記号へと変換（map）させるトランスデューサ T である。T を伴った G

の拡張は、標準合成と投影（projection）演算を使用して行われる。

$$GT = Sort_1(Proj_2(G \circ T)) \quad (2)$$

ここで下付き文字、1、2の意味するものは、入力と出力記号をそれぞれ表す。$Proj_2$ は GT からの出力記号全てを単語記号と入れ替える。最終 *Sort* とは、*CL* との合成をより効果的にするために必要なものである。音声デコーディングする際の全カスケードは次の通り。

$$CL \circ (Sort_1(Proj_2(G \circ T))) \quad (3)$$

CL が $C \circ det(L)$ の略称である。クラスN-gramの静的拡張は、*G* における各クラス記号が *T* での全ての遷移に積算される可能性があるので、しばしば大きなメモリを要することがある。したがって、*GT* の拡張はOn-the-flyにて行われる。我々の提唱するアルゴリズムは、この *G* と *T* のWFSTを使った限られたトポロジーを活用しようとするものである。我々が仮定するのは、*T* は単体の状態をもち、それぞれのアークがクラス記号から1つの単語記号への変換（mapping）を表す、ということである。よって、合成 $G \circ T$ によって作られたWFSTは、*G* と同じ数の状態を持つ。この制限下では合成と整理（sort）することは1組のソートリストをマージすることと同義であり、投影（projection）演算をマージ処理に融和させることに追加されたものである。

クラスラベルN-gramのWFST *G* と、単語とその単語が属するクラスのマッピングを行うWFST *T* とを融合したWFST *GT* は、次のように動作する。*T* は、単語リストの集合として表現され、その各リストはソートされ、対応するクラスラベルが付与される。デコーディング中に *GT* 内の状態 *s* がアクティブになったとき、最初に、*G* の状態 *s* を出て行く *k* 個のアークに対して各クラスに対応する *k* 個の単語リストを作る。次に、出力ラベル（単語）をキーとするmin-heapを用いて、*k* 個のリストをマージする。マージに必要な計算量は、状態 *s* を出て行くアーク数を *k* 個のリストに含まれる単語数の総和を *n* とすると、$O(nlog(k))$ である。この融合されたアルゴリズムは、従来、射影演算やソート演算に必要とされたメモリも排除できる。

5　結論

本書では、SprinTraデコーダを紹介した。我々はNICT向けに今後の音声認識と音声翻訳研究のための最先端エンジンを提供するという目標を達成できた。加えて、可動性の高いエンジンと、使いやすいツールキットにより、ドメインを問わない（non-domain）専門家、開発者が、最新のWFSTベースの音声認識を簡単に活用できるようになった。今後のバージョンでは、もっと効果的なアルゴリズムに着手しダイナミック語彙に対応したい。これはある特殊なアプリに非常に重要とされている。また現在は、SprinTraとWFSTベースの会話や翻訳システムとの結合のための新しいメソッド研究に焦点を当てている。開発前線においては、SprinTraの可動性を打ち出し、スタンダローン型から、スマートフォンやタブレットのようなより小サイズのプラットフォームに持っていきたいと考えている。

参考文献

1 M. Mohri, F. Pereira, and M. Riley, “Weighted finite-state transducers in speech recognition,” Computer Speech and Language, 16(1): 69–88, 2002.
2 S. Kanthak, H. Ney, M. Riley, and M. Mohri, “A comparison of two LVR search optimization techniques,” In Proc. ICSLP, pp. 1309–1312, 2002.
3 Diamantino Caseiro and Isabel Trancoso, “Transducer composition for on-the-fly lexicon and language model integration,” In Proc. ASRU, pp. 393–396, 2001.
4 Diamantino Caseiro and Isabel Trancoso, “Using dynamic WFST composition for recognizing broadcast news,” In Proc. ICSLP, pp. 1301–1304, 2002.

5 Takaaki Hori, Chiori Hori, and Yasuhiro Minami, "Fast on-the-fly composition for weighted finite-state transducers in 1.8 millionword vocabulary continuous speech recognition," In Proc. Interspeech, pp. 289–292, 2004.

6 T. Hori and A. Nakamura, "Generalized fast on-the-fly composition algorithm for WFST-based speech recognition," In Proc. Interspeech, pp. 847–850, 2005.

7 D. A. Caseiro and I. Trancoso, "A specialized on-the-fly algorithm for lexicon and language model composition," IEEE Transactions on Audio, Speech, and Language Processing, 14(4): 1281–1291, 2006.

8 Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," IEEE Transactions on Audio, Speech and Language Processing, 15: 1352–1365, 2007.

9 T. Oonishi, P. R. Dixon, K. Iwano, and S. Furui, "Implementation and evaluation of fast on-the-fly WFST composition algorithms," In Proc. Interspeech, pp. 2110–2113, 2008.

10 T. Oonishi, P. R. Dixon, K. Iwano, and S. Furui, "Generalization of specialized on-the-fly composition," In Proc. ICASSP, pp. 4317–4320, 2009.

11 C. Allauzen, M. Riley, and J. Schalkwyk, "A generalized composition algorithm for weighted finite-state transducers," In Proc. Interspeech, pp. 1203–1206, 2000.

12 Hasim Sak, Murat Saraclar, and Tunga Gungor, "On-the-fly lattice rescoring for real-time automatic speech recognition," In Proc. Interspeech, pp. 2450–2453, 2010.

13 Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai, "Class-based n-gram models of natural language," Computer Linguistics, 18(4): 467–479, Dec 1992.

14 M. Mohri, F. C. N Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," Springer Handbook of Speech Processing, pp. 1–31, 2008.

15 M. Mohri, "Weighted automata algorithms," Springer Handbook of weighted automata, (to appear) 2009.

16 C. Allauzen, M Mohri, and B. Roark, "Generalized algorithms for constructing statistical language models," In Proc. of 41st Annual Meeting of the Assocication for Computational Linguistics, pp. 40–47, 2003.

17 S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK Book (for HTK Version 3.2)," Cambridge University Engineering Department, 2006.

（平成 24 年 6 月 14 日 採録）

DIXON Paul Richard
ユニバーサルコミュニケーション研究所
音声コミュニケーション研究室研究員
博士（工学）
音声認識および言語処理
paul.dixon@nict.go.jp

堀　智織（ほり　ちおり）
ユニバーサルコミュニケーション研究所
音声コミュニケーション研究室
主任研究員
博士（学術）
音声認識、音声翻訳、音声対話技術
chiori.hori@nict.go.jp

柏岡秀紀（かしおかひでき）
ユニバーサルコミュニケーション研究所
音声コミュニケーション研究室室長
博士（工学）
音声言語処理、音声翻訳、音声対話
hideki.kashioka@nict.go.jp