論文 / 著書情報
Article / Book Information

論題(和文)	WFSTを用いた音声認識デコーダの機能拡張
Title(English)	
著者(和文)	Paul Dixon, 大西 翼, 古井 貞熙
Authors(English)	Paul R. Dixon, Oonishi Tasuku, Sadaoki Furui
出典(和文)	日本音響学会 2007年秋季講演論文集, Vol. , No. 2-3-22, pp. 105-106
Citation(English)	, Vol. , No. 2-3-22, pp. 105-106
発行日 / Pub. date	2007, 9

WFSTを用いた音声認識デコーダの機能拡張*

◎ Paul R. Dixon, 大西 翼, 古井 貞熙 (東工大)

1 はじめに

近年，高精度で柔軟性の高い音声認識手法としてWFST(Weighted Finite State Transducer) の利用が検討され，その有効性が確認されている [1].

我々は実用的な音声認識デコーダの実現に向けて，WFST を利用したデコーダの開発と性能評価を行っている [2]. 本稿では，本デコーダの機能拡張として，高速化，省メモリ化及び様々なアプリケーションとの親和性の向上について種々の手法の検討と性能評価を行った結果について報告する.

2 WFSTに基づく音声認識

WFST(Weighted Finite State Transducer) とは，入力記号列に対して状態遷移を繰り返し，それに対応した出力記号列と重みを出力する有限状態オートマトンの一種である. 音声認識に用いる際には，利用される様々な情報 (音響モデル，発音辞書，N-gram など) をそれぞれ WFST で表現し，それらを合成することで一つの探索ネットワークを作成する. デコーダはこのネットワークを用いて最尤となる単語列を探索し音声認識を行う. 探索ネットワークを構築する際には，WFST の基本操作である決定化 (determinization)，最小化 (minimization) 等を用いてネットワークの最適化を行う.

この最適化によって探索の効率化が計られ高速に音声認識を行うことができる. また探索ネットワークに統一的な枠組を用いることにより，デコーダの変更を伴わずに様々なモデルを柔軟に利用することができる.

一方，様々な情報を一つに合成することにより探索ネットワークのサイズが肥大化し，認識時に多くのメモリが必要となる場合がある.

3 機能拡張

WFST に基づく音声認識では，メモリ消費量を削減する様々な手法が提案されている [3–5]. 本研究ではその中で探索ネットワークをディスクに保持したまま探索を行うディスクベースドサーチについての検討と性能評価を行った.

また，音声認識の高速化を行うため GPU(Graphics Processor Unit) を用いた音響尤度計算を提案し，その性能評価を行った.

様々なアプリケーションとの親和性を高めるためラティス形式での認識結果出力を実装し，出力を行う際の実効速度についての評価を行った.

3.1 ディスクベースドサーチを用いた音声認識手法の検討

WFST に基づく音声認識では探索ネットワークの保持に多くのメモリが消費されている. そこで本研究では探索ネットワークをディスクに保持し，それを探索時に参照するディスクベースドサーチについての検討を行った.

今回の実装では探索時に到達した状態，状態遷移のデータをディスクからメモリに読み込み，不要になり次第データをメモリから開放する. このようなディスクベースドサーチは，消費メモリが探索ネットワークのサイズに依存しないため，消費メモリを大幅に削減することが可能であり，超大語彙連続音声認識など巨大な探索ネットワークを用いて音声認識を行う際に有効であるといえる.

3.2 GPUを用いた音響尤度計算の高速化の検討

大語彙連続音声認識では音響尤度の計算に多くの時間が消費されているため，その計算を高速化することで効率的に認識時間を削減することができる. 近年 GPU の浮動小数点速度は，CPU と比較し飛躍的に向上しており，様々なアプリケーションに GPU を利用する試みが行われている [6].

我々は，音響尤度計算高速化の一つのアプローチとして GPU を用いた音響尤度計算手法を提案し，実装と性能評価実験を行った. 実装では，あるフレームの音響尤度計算を行うにあたり，仮説毎に必要となるガウス分布に対する音響尤度を逐一計算するのではなく，モデル内の全てのガウス分布に対する音響尤度計算を一括して行った. これにより CPU と GPU の間のデータ転送回数を削減させた.

3.3 ラティス形式の出力の検討

音声検索やリスコアリング処理など様々なアプリケーションでの利用を想定し，ラティス形式での認識結果の出力の実装及び性能評価を行った. 実装は [7] と同様の手法で行った.

4 実験

性能評価実験には日本語話し言葉コーパス (Corpus of Spontaneous Japanese) [8] を用いた. 学習用データとして学会講演 228 時間 953 講演を用いた.

音響特徴量として フレームシフト 10ms, ウィンドウ幅 25ms，MFCC 12 次元+Δ MFCC 12 次元，ΔΔ MFCC 12 次元+Δ対数パワー+ΔΔ対数パワーの計 38 次元を用いた. 音響モデルは 3000 状態 16 混合の triphone HMM を用いた. 言語モデルは語彙サイズ 25,000 単語の trigram を用いた.

評価データには，男性の学術講演 10 講演 (perplexity=57.8) を用いた. 実験は Intel Core2 2.4GHz 2GB メモリ，グラフィックプロセッサ NVIDIA 8800GTX で行った.

4.1 ディスクベースドサーチ評価実験

ディスクベースドサーチを行った場合 (Disk Based) と行わなかった場合 (Memory Based) のビーム幅と使用メモリの関係を Fig. 1 に，処理時間と認識精度の関係を Fig. 2 に示す. Fig. 1 から，ディスクベースドサーチを行うことにより平均して使用メモリを 60%程度削減できていることが分かる. また認識精度がほぼ収束している RTF=0.4 付近での認識速度の低下は 10%程度であり，ディスクアクセスに伴うオーバーヘッドが抑制されていることが分かる. これは CPU などのキャッシュ機能が有効に働きディスクアクセスに伴うオーバーヘッドが抑制できたためであると考えられる.

*Improvements in a WFST-based speech decoder. by Paul R. Dixon, Tasuku Oonishi, Sadaoki Furui (Tokyo Institute of Technology)

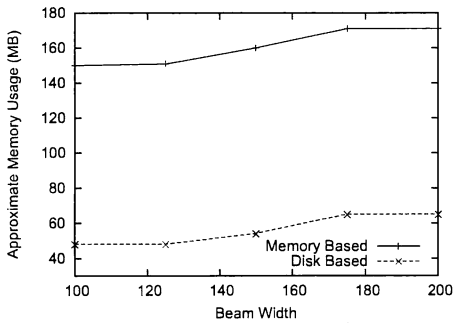

Fig. 1　ディスクベースドサーチを利用した場合のメモリ消費量

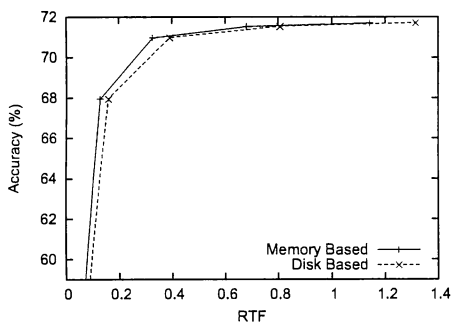

Fig. 2　ディスクベースドサーチを利用した場合の認識時間と認識精度の関係

4.2　GPU を用いた音響尤度計算評価実験

浮動小数点演算を高速に計算する手法の一つである SIMD(Single Instruction Multiple Data) を利用した音響尤度計算手法 [9] との比較を行った．ビーム幅と認識時間の関係を Table 1 に示す．

これから GPU を用いることによりビーム幅が大きい場合で 30%程度の速度向上が確認できる．一方，ビーム幅が小さい場合では，認識速度の低下が見られた．これは GPU を用いた場合では全てのガウス分布の音響尤度計算を行っているため，無駄な音響尤度計算が発生し，尤度計算の高速化の効果を打ち消したためであると考えられる．

4.3　ラティス形式出力の性能評価実験

ラティスを生成する場合と生成しない場合の認識時間と認識率の関係を Fig. 3 に示す．両者を比較すると認識時間にほぼ差が見られないことから，今回の実装では認識時間の増加を抑えつつラティス形式の出力が可能であることが分かった．

Table 1　GPU による計算時間削減の効果

Beam	SIMD RTF	GPGPU RTF	RTF Reduction (%)
100	0.045	0.11	-57.48
125	0.14	0.15	-9.49
150	0.35	0.29	23.37
175	0.75	0.57	29.8
200	1.26	0.99	26.05

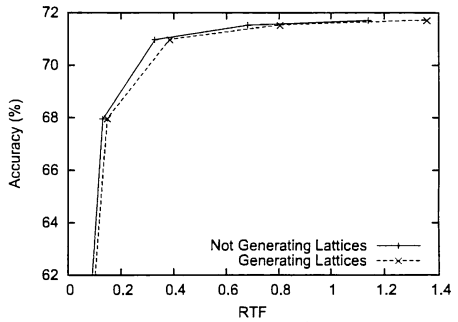

Fig. 3　ラティス生成時の認識時間と認識率の関係

5　まとめ

本稿では WFST を利用した音声認識デコーダの機能拡張についての評価及び検討を行った．その中で，省メモリ化，高速化，アプリケーションとの親和性の向上についての検討を行い，それぞれの手法で性能評価実験を行い，有効性を確認した．今後は超大語彙連続音声認識など大規模な探索ネットワーク利用を想定した場合の効率的な探索手法の検討を行っていきたい．

謝辞　本研究は 21 世紀 COE プログラム「大規模知識資源の体系化と活用基板構築」及び経産省「情報家電センサー・ヒューマンインターフェースデバイス活用技術開発・音声認識基盤技術」プロジェクトの支援により行った．

参考文献

[1] M. Mohri et al., "Weighted finite-state transducers in speech recognition," Computer Speech and Language, vol. 16, no. 1, pp. 69–88, 2002.
[2] D. A. Caseiro et al., "Preliminary evaluations of a WFST speech decoder," 春季音響論, pp. 19–20, 2007.
[3] T. Hori et al, "Generalized fast on-the-fly composition algorithm for WFST-based speech recognition," INTERSPEECH, pp. 847–850, 2005.
[4] D. A. Caseiro et al, "A specialized on-the-fly algorithm for lexicon and language model composition," IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 4, pp. 1281–1291, 2006.
[5] D. Willett et al., "Time and memory efficient Viterbi decoding for LVCSR using a precompiled search network," EUROSPEECH, pp. 847–850, 2001.
[6] J. D. Owens et al., "A survey of general-purpose computation on graphics hardware," Eurographics, pp. 21–51, 2005.
[7] A. Ljolje et al., "Efficient general lattice generation and rescoring," EUROSPEECH, pp. 1251–1254, 1999.
[8] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," ISCA and IEEE Workshop on Spontaneous Speech Processing and Recognition, pp. 7–12, 2003.
[9] S. Kanthank et al., "Using SIMD instructions for fast likelihood calculation in LVCSR," ICASP, pp. 1531–1534, 2000.