

Internal Use Only (非公開)

TR-SLT-0024

在宅健康管理のための音声対話システムの実装
Implement of Spoken Dialogue System for Home Health Care

高橋伸弥

Shin-ya Takahashi

2002年9月30日

概要

福岡大学で現在開発中の在宅健康管理支援システムの音声認識モジュールを ATR SPREC へと変更した音声対話システムのプロトタイプシステムの実装を行った。本システムは高齢者を対象として音声インターフェースを用いて健康管理に関する対話を行うものである。システムはユーザの音声発話を認識した後、対話プランと認識結果との間でマッチングを行い、ユーザの状態に応じて適切な対話計画をする。本稿では、システムの概要を説明した後、音声認識モジュールの組み込み方法について報告する。

(株) 国際電気通信基礎技術研究所
音声言語コミュニケーション研究所

〒619-0288 「けいはんな学研都市」 光台二丁目 2 番地 2 TEL : 0774-95-1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288, Japan
Telephone: +81-774-95-1301
Fax : +81-774-95-1308

©2002 (株) 国際電気通信基礎技術研究所

©2002 Advanced Telecommunication Research Institute International

目次

1	はじめに.....	1
2	システムの概要.....	2
2.1	ケアエージェント(対話マネージャ).....	2
2.2	健康診断サーバ.....	2
2.3	測定サーバ.....	3
3	効率的な対話の生成.....	4
3.1	対話プランニング.....	4
3.2	検診・問診順序の動的な制御.....	5
4	音声認識モジュールの変更.....	8
4.1	ATR SPREC との接続.....	8
4.2	認識用文法・辞書ファイルの作成.....	9
4.3	ATR SPREC を組み込んだプロトタイプシステムの実装.....	10
5	まとめ.....	11
	謝辞.....	11
付録	ATR SPREC 認識用設定ファイル.....	13

1 はじめに

近年、高齢者人口の増加にともない、高齢者の医療や健康管理が大きな社会問題となりにつつある。多くの高齢者は、入院するほどの重病ではないものの、何らかの持病を持っていたり健康上の不安を抱えていたりする場合が多い。そういった高齢者が自宅においても安心して頼れるような在宅健康管理システムが実現できれば、患者のみならず医療行政面、医療サービス面でも大きなメリットがある。このような状況において、これまでも、在宅・遠隔医療支援のための多くの試み[1]がなされているが、医師または患者の負担やシステムに対する患者の不満といった面で不十分な点も少なくない。

現在、福岡大学では、上記のような問題点を解決することを目的として、システムとの対話を通して患者の健康状態を見守ることのできるような在宅健康管理支援システムの開発を行っている[2]。このシステムは、システムに接続された測定機器を用いて定期的な検診を行い、必要に応じてユーザへの問診を行うものであり、次のような特徴を持つ。

- ① ユーザの健康状態に関する医学的に適切な対話を行う
- ② 不要な質問や冗長な問診を少なくし、効率的な対話を行う
- ③ ユーザからの体調不良の訴えなどの割り込みに対処する

①および②は、医学的知識に基づいて検診／問診項目を決定する健康診断プログラムと連携して対話プランを生成する事により実現する。③は、ユーザからの割り込みに対して、現在進行中の対話プランと照らし合わせ、次に実行すべきプランを動的に決定することにより実現する。

本研究は、上記の音声対話システムにおける音声認識モジュールを ATR SPREC へと変更したプロトタイプシステムの実装を行ったものである。本報告では、システムの概要を説明した後、音声認識モジュールを ATR SPREC へと変更した際の組み込み方法について報告する。

2 システムの概要

図1に、本システムの構成を示す。本システムは、対話管理のためのケアエージェントと、医学知識や医療機器を取り扱うためのコンポーネント、入出力を制御するコンポーネントから構成されている。システムは、これらのコンポーネントの間で、メッセージ通信を行いながら、処理を進めていく。以下、図に示したコンポーネントのうち、主要なもの役割について説明する。

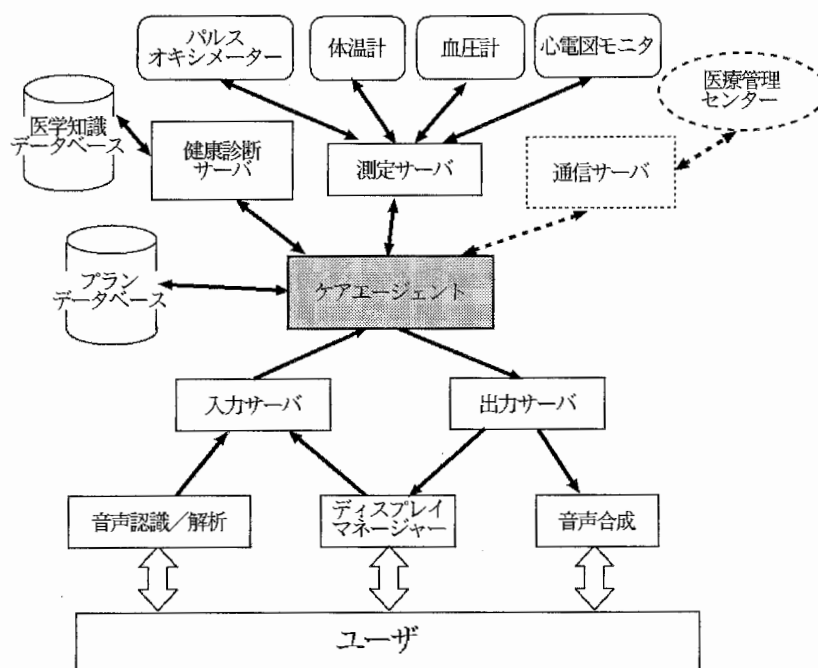


図1 システムの構成

2.1 ケアエージェント (対話マネージャ)

本システムを中心となるコンポーネントであり、全体のプランの管理を行う。ユーザや他のコンポーネントとのやりとりを制御し、次に実行すべきプランを動的に決定する。

2.2 健康診断サーバ

ケアエージェントからの要求に伴い、医学知識を参照して、適切な診断プランを作成する。また測定データや問診結果をもとに疑わしい疾患名を切り分け、緊急性の高いものかどうか判断する(一次診断機能)。ただし、これはあくまで病院に緊急に連絡すべき事態か

どうかを判断し、また病院に連絡しておいた方が望ましい情報を収集するものであって、最終的な医療診断は人間の医師が行うものとする。

2.3 測定サーバ

体温計、血圧計、パルスオキシメーター(血中酸素飽和度 SpO₂ を測定する機械)などの各種医療機器を制御し、ケアエージェントからの要求に従って測定を行う。

3 効率的な対話の生成

3.1 対話プランニング

システムは、基本的に、図2に示すように木構造で表された定期検診プランに従って対話を進める。このプランでは、まず簡単な挨拶の後、体温、血圧、SpO2などの基本的な生体情報の測定を行い、特に異常がなければ、検診結果を履歴データに追加し、最後に簡単な挨拶を行ってセッションを終了する。検診結果に異常が見られた場合やユーザから何らかの異常の訴えがあった場合には、実行すべき検診項目や問診項目を、健康診断サーバに問い合わせ、プランを再構成する。

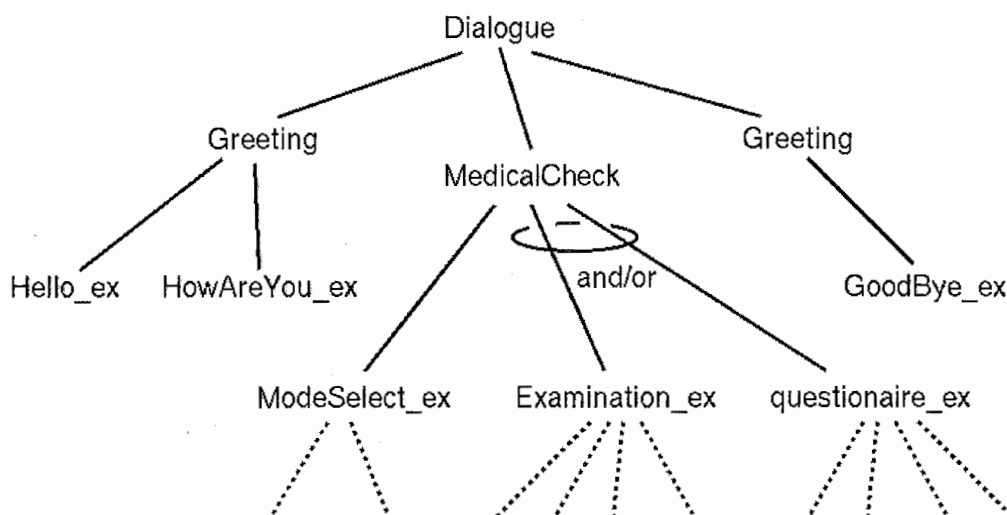


図2 定期検診プランの例

ケアエージェントは、あらかじめ定義されたプラン・スキーマを参照し、その状況にマッチングするプランを順次展開しながら、システム全体の処理を進める。ユーザからの発話に応じて、適切なプランを展開するために、システムは未展開のノードを次発話の範囲として予測しておき、実際の発話が接続可能なノードをボトムアップ的に判定し、そのノードを展開して入力された発話を接続する。

システムは、入力された発話を形態素解析し、その中から名詞句や動詞句、肯定/否定表現などのキーフレーズを抽出する。ここでは、形態素解析には、奈良先端科学技術大学院大学で公開されている“茶筌”を使用した[3]。このキーフレーズと実行可能な予測プランとをマッチングして、マッチするプランがあれば、対話プランのツリーを展開し、そのプ

ランを実行する。ここで、キーフレーズとマッチするプランがなければ、ユーザに対して再問い合わせを行う。

また、システムはユーザの発話を誘導するために、選択可能な答をメニュー項目として表示する。このメニュー項目は、システムの発話（質問）に対する答の候補を参照して、動的に作成される。ユーザは、システムに対して音声で答えるだけでなく、画面上に表示されたメニュー項目を選択することでも、対話を進めることができる。

図3に、トピックシフトの例を示す。ここで、Sはシステムの発話、Uはユーザの発話を示している。この対話例では、前述したようにセッションの開始後にシステムがまず簡単な挨拶を行っている(S1,S2)。これに対し、ユーザは挨拶を返しても良いが、その部分をスキップし、この例のように直ちに健康状態に関する話題を始めてもよい(U3)。

S1: こんにちは
S2: 今日は、とても暑いですね
U3: ちょっと、具合が悪いんだけど [健康状態に関するトピックにシフト]
S4: それはよくないですね
S5: どんな症状ですか?
U6: 頭が痛いです
S7: とりあえず、熱を測っておきましょう

図3 トピックシフトの例

3.2 検診・問診順序の動的な制御

ケアエージェントは、ユーザから何らかの不具合の訴えを受けると、その訴えに対する適切なプランを実行するために、健康診断サーバに問い合わせのメッセージを送る。診断サーバは、診断規則を参照して、次に実行すべき検診または問診の項目を決定する。さらに、検診や問診の結果から、疑わしき病名を絞り込むことを行う。

一次診断のための診断規則は高齢者が罹りやすい疾患（または頻度が高い症状）で、かつ緊急性の高いものの抽出に重点を置いて構成している。表1および表2に、システムに組み込まれている測定項目および問診項目と最終的に判断される（疑わしき）病名の例を示す。一次診断のための診断規則は、現場の医師の専門的な知識に基づいて、予め作成しておく。診断規則は、医師が定義しやすいよう、表3のような形式となっている。表の第

1行目の各列には、測定・問診項目が記されている。2行目以降の各行は病名に対応し、各列には対応する測定・問診項目の値（すなわち症状）が記されている。最右の欄は疑わしい病名である。一次診断では、各列に対応する測定・問診を順次行い、その症状を持つ行を絞り込んで行く。ユーザの答が、候補として残っている病名の条件に反する場合には、その病名が候補から取り除かれる。ここで、取り除かれた病名に関連した問診項目も同時に除外されるので、不要な質問を減らすことができ、効率的な対話を行なうことができる。例えば、「めまいがする」といった症状の場合、めまいの種類によって疑わしい病名を絞ることができるが、ユーザが「めまいはしない」と答えた場合には、めまいの種類を尋ねることなく、他の問診項目を質問する。

表1 測定、問診項目の例

分類	測定項目／問診項目
測定項目	体温、拍数、血圧、SpO ₂
問診項目	頭痛の有無、腹痛の有無、便秘の有無、吐気の有無、風邪症状の有無、胸の痛みの有無、胸の痛む状況（息を深く吸った時、咳をした時、横になった時）、咳の有無、痰の有無、痰の色、痰に血液が混入しているかどうか、息切れ（身体を動かした時）、喉の痛みの有無、関節の痛みの有無、めまいの有無、耳鳴りの有無、聴力低下の有無

表2 判断される病名の例

種別	病名
緊急性が高い、または要注意のもの	心筋梗塞、発作性頻拍症、心不全、髄膜炎、小脳血管病変 病的徐脈、肺炎、急性気管支炎、気管支喘息、肺気腫 感染性胃腸障害、腸閉塞
上記以外のもの	貧血、頻脈、感冒発熱、上気道炎、慢性気管支炎 気管支拡張炎、肋膜炎、肋間神経痛、咽頭炎、扁桃炎 緊張性頭痛、低血圧

表 3 診断規則の例

体温	心拍数	・ ・	頭痛	痰	・ ・	吐き気	病名
>37	—		あり	—		なし	風邪
:							
≧38	—		なし	あり		なし	肺炎疑い
:							
>37	—		あり			あり	髄膜炎疑い

4 音声認識モジュールの変更

4.1 ATR SPREC との接続

ここでは、前章までに説明した音声対話システムの音声認識モジュールを、HMM Tool kit (HTK) [4]の認識エンジン HVite から ATR SPREC へと変更した際の組み込み方法を説明する。なお、ここで使用した ATR SPREC のバージョンは、Ver08r02 である[5]。

ATR SPREC では、複数のモジュールを組み合わせることにより、オンライン音声認識を行うことができる。音声対話システムの一部として、この ATR SPREC による音声認識プログラムを組み込む場合には、認識結果を対話システムへと受け渡すための機構が必要となる。そこで、ATR SPREC の出力の中から音声認識結果を切り出し、ソケットによるプロセス間通信を用いて対話システムの入力サーバへと送信するフィルタプログラムを作成した。このプログラムは、UNIX のシェルの機能であるパイプを用いて、標準出力を受け取り、認識結果の文を切り出して、入力サーバへと送信するものである(図3)。

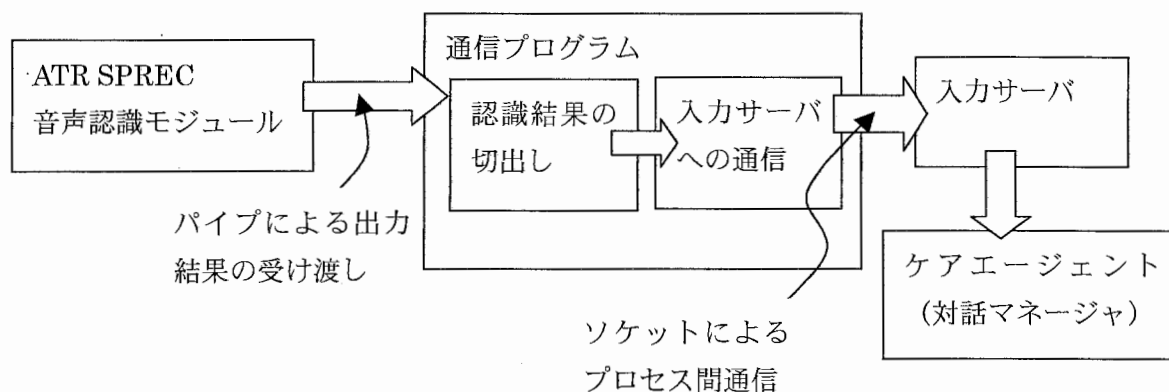


図3 ATR SPREC 認識モジュールと対話システムとの接続

音声認識モジュールは、以下の5つのモジュールを組み合わせで作成した。

- ATRepd
- ATRwave2cep
- ATRcep2para
- ATRlattice
- ATRresult

基本的には、ATR SPREC のソースファイルに付属の sample/ATRall と同様の構成である。
また、マイクからの音声入力には、DATLink を使用した。

4. 2 認識用文法・辞書ファイルの作成

次に、HTK 用の認識用ネットワーク文法および辞書ファイルを ATR SPREC 用に変更した。まず、各単語の品詞等を定義したクラス分類ファイルとクラス間の接続規則を記述した CFG 定義ファイルから、クラスレベル FSA 言語モデルを作成し、その後、単語レベル FSA を作成する。下図に処理の流れを示す。

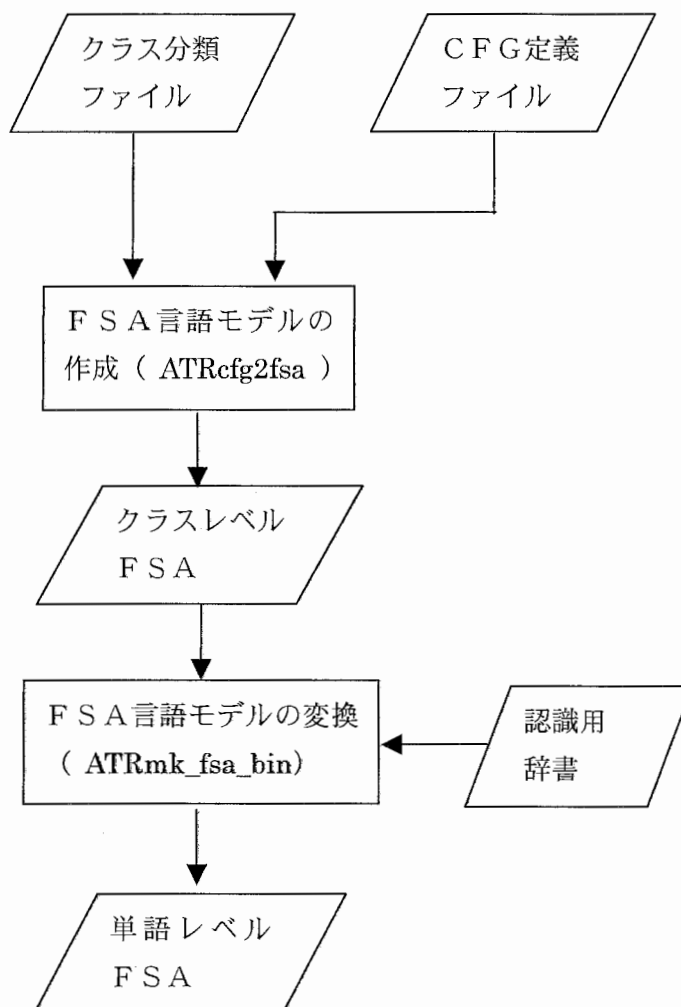


図4 認識用 F S A 言語モデルの構築

また、ATR SPREC と HTK とでは、使用する音節の表記体系が異なるため、若干の変更を行う必要がある¹⁾。

4. 3 ATR SPREC を組み込んだプロトタイプシステムの実装

前節までに説明した変更を行い、HVite の代わりに ATR SPREC を音声モジュールに組み込んだ在宅健康管理音声対話システムのプロトタイプシステムを実装した。組み込んだ FSA 言語モデルの語彙は約 200、受理しうる文パターン数は約 50 である。また音響モデルも、これまでに使用していた Julius に附属のモデル[6]から、ATR SPREC 用のものに変更してある。ATR SPREC 実行時に使用した設定ファイルを付録に添付しておく。基本的な対話プランには、図 2 に示した定期検診プランを使用した。実行可能なプラン数（プランツリーの葉の部分に相当する）は、約 50 である。プロトタイプシステムでは、一次診断のための規則として、36 種類の病名と 32 の検診/問診項目をテーブル化したものを用いた。図 4 に実装したプロトタイプシステムのユーザインタフェースを示す。



図 3 プロトタイプシステム

¹ “ん” N (HTK) → ng (SPREC)、 “よ” yo (HTK) → jo (SPREC) など

5 まとめ

本研究では、福岡大学で現在開発中の在宅健康管理のための音声対話システムの音声認識モジュールを ATR SPREC へと変更したプロトタイプシステムの実装を行った。実装にあたって、ATR SPREC の認識結果を対話システム本体に受け渡すためのプログラムを作成した。

今後は、より完成度の高いシステムを目指して改良を進めていくとともに、より自然な対話の実現のため、断片的な発話の処理[7,8]を検討する予定である。また、実際の診断データを用いて一次診断規則を検証すること、高齢者が5、6人で生活しているグループホームへと導入することを予定している。さらに、画像情報を積極的に取り入れ、システムからの出力として、より人間らしい表情や動きを持った擬人化エージェントを表示するだけでなく、システムへの入力として、ユーザの動作や顔画像の認識を行う機能も組み入れる予定である。高齢者音声の認識に関しては、現在いくつかの研究機関や大学で精力的に研究が進められている[9]ので、その進展を待つこととしたい。

謝辞

本研究は、福岡大学から2ヶ月間の長期主張という形で ATR に出向した間に行ったものである。変則的な短期間の滞在を受け入れてくださいました音声コミュニケーション研究所の山本誠一所長、第二研究室の菊井玄一郎室長には深く感謝致します。また、貴重な経験を積む機会を与えてくださった福岡大学工学部の森元逞逞教授に感謝します。滞在期間中には竹澤寿幸氏をはじめ第二研究室の皆様には色々と便宜をはかっていただきました。心から感謝いたします。

参考文献

- [1] <http://square.umin.ac.jp/~enkaku/proj/enkproj-hom.html>
- [2] 森元逞, 鶴田直之, 小川健一, 高橋伸弥, 前田佐嘉志, 嘉悦智隆: "高齢者のための在宅健康管理支援システム", 電子情報通信学会技術報告 WIT, 101 (264), pp. 13--18, 8 2001.
- [3] <http://chasen.aist-nara.ac.jp/>
- [4] <http://htk.eng.cam.ac.uk>
- [5] <http://winnie.kuis.kyoto-u.ac.jp/pub/julius/>
- [6] ATR 音声翻訳通信研究所: ATR SPREC (Ver08r02) ユーザマニュアル
- [7] M. Nakano, N. Miyazaki, J. Hirasawa, K. Dohsaka, and T. Kawabata: "Understanding unsegment user utterances in real-time spoken dialogue systems", Proceedings of 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), pp. 200--207, 1999.
- [8] D. Byron: "Improving discourse management in TRIPS-98", Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech-99), volume 3, pp. 1379--1382, 1998.
- [9] 馬場朗, 芳澤伸一, 山田実一, 李晃伸, 鹿野清宏: "高齢者音響モデルによる大語彙連続音声認識", 電子情報通信学会論文誌, J85-D-11 (3), pp. 390--397, 3 2002.

付録 ATR SPREC 認識用設定ファイル

```
ATRSPREC Copyright 1997, 1998 ATR Interpreting Telecommunications Research Laboratories
I/Ocontrol:inputFormat=NoHeader
I/Ocontrol:inputParamSize=90
I/Ocontrol:inputParamType=short
I/Ocontrol:inputFd=process("/usr/local/datlink/bin/narecord -o right -p 16 -e linear -f
raw -s 16000 -u pxn246:0")
# I/Ocontrol:inputFd=$ATRSPREC/sample/ATRepd/wave
# I/Ocontrol:inputFd=stdin
I/Ocontrol:inputByteorder=BigEndian
I/Ocontrol:outputFormat=String
I/Ocontrol:outputFd=stdout
I/Ocontrol:outputByteorder=BigEndian
I/Ocontrol:inputEOFexit=ON # OFF
I/Ocontrol:rpcNumber=2

#ATRepd config :
ATRepd:SamplingFrequency=16000
# ATRepd:energyThreshold=100
# ATRepd:energyThreshold=50
ATRepd:energyThreshold=30
# ATRepd:upperDispersionThreshold=30
ATRepd:lowerDispersionThreshold=15
ATRepd:orderInMs=300
# ATRepd:alpha=1.04
# ATRepd:alpha=1.06
ATRepd:alpha=1.10
ATRepd:skewInMs=300
ATRepd:s1TimerLimitInMs=200
ATRepd:epdFramesInMs=2000
ATRepd:framePointsInMs=10
ATRepd:sendNonspeech=0
ATRepd:track=0

#ATRwave2cep config :
ATRwave2cep:AnalysisType=fft
ATRwave2cep:CalcMean=off
ATRwave2cep:CentroidFreqGamma=0.5
ATRwave2cep:CentroidFreqOrder=0
ATRwave2cep:CentroidFreqType=linear
ATRwave2cep:CepstrumOrder=12
ATRwave2cep:CutoffHighFrequency=8000
ATRwave2cep:CutoffLowFrequency=0.0
ATRwave2cep:DebuggingLevel=0
ATRwave2cep:FilterBankOrder=20
ATRwave2cep:FrameLength=20
ATRwave2cep:FrameShift=10
ATRwave2cep:FrequencyWarping=mel
ATRwave2cep:LagWindowFactor=0.01
ATRwave2cep:LpcOrder=16
ATRwave2cep:MeanInFile=$ATRSPREC/sample/ATRcard/CM.TRS.CDB05.mean
ATRwave2cep:Preemphasis=0.98
```

```

ATRwave2cep:SamplingFrequency=16000
ATRwave2cep:Subtract=logpow+cep
ATRwave2cep:TimeWindow=hamming

#ATRcep2para config :
  ATRcep2para:CepstrumOrder=12
  ATRcep2para:DDCepstrumPadding=zero
  ATRcep2para:DDCepstrumWindow=5
  ATRcep2para:DebuggingLevel=10
  ATRcep2para:DeltaCepstrumWindow=5
  ATRcep2para:LDA=
  ATRcep2para:OutputParameter=cep (12) +dpow+dcep (12)
  ATRcep2para:WindowType=rectangular
  ATRcep2para:deltaCepstrumPadding=zero
  ATRcep2para:rho=1.0

#
  ATRlattice:lexicon=$ATRSPREC/sample/ATRcard/GRAMMAR/card.lex
#
  ATRlattice:FSA=$ATRSPREC/sample/ATRcard/GRAMMAR/card.fsa.bin
  ATRlattice:lexicon=$ATRSPREC/sample/ATRA12/GRAMMAR/Test3/test.lex
  ATRlattice:FSA=$ATRSPREC/sample/ATRA12/GRAMMAR/Test3/test.fsa.bin
  ATRlattice:ngram=
  ATRlattice:dimension=25
  ATRlattice:aname=/RR/Recognition/ResearchJ/amodel/19991111/trainSLFfromTRS.BLA/AM.F.
5. 1400.bin./RR/Recognition/ResearchJ/amodel/19991111/trainSLFfromTRS.BLA/AM.M.5.1400.bin
  ATRlattice:lm scale=4.0,8.0
#
  ATRlattice:work_area=20,50
  ATRlattice:work_area=400,500
#
  ATRlattice:beam=50,50
  ATRlattice:beam=150,200
  ATRlattice:word_merge=all
  ATRlattice:null_trans=OFF
  ATRlattice:phone_boundary=OFF

#
  ATRresult:answer=$ATRSPREC/sample/ATRresult/ans1.ANS,$ATRSPREC/sample/ATRresult/answe
r. TRS
  ATRresult:N_best=5
  ATRresult:N_best_out=stdout
  ATRresult:lattice_out=lattice.test
### EOF

```