Speech Mashups
A Compositional Approach to Speech, Semantic, and Web Services

Giuseppe (Pino) Di Fabbrizio
pino@research.att.com
IP & Voice Services Research
AT&T Labs – Research, Inc.
Outline

• Background
• Human Computer Interaction Overview
• Speech Technology & Speech Services
• Migrating Speech to the Web
• (Semantic Web Integration)
• Demonstrations
• Q&A
Background - Technology Drivers
Why internet services are (subtly) pervasive

Interoperability & Distribution
• Publicly available web services provide web users with an easy and convenient way to discover and exploit new services and concepts (OS independent, mashups or web application hybrids)
• While many traditional web players are publishing their APIs (Google, Yahoo, Amazon, etc), mobile telecommunication provides (Vodafone, British Telecom, etc.) are entering this service space by making telephony and mobile-based APIs available

IP Multimedia Subsystem (IMS)
• IMS Wireline and Wireless network seamless mobility and convergence of services and devices in one consistent architecture is an ideal environment for advanced speech and multimodal services
• Potential synergies: Location Based Services (LBS), WiFi/3G (GPRS/EGPRS(EDGE)) mobility, Instant Messaging, Presence Information, Video/picture sharing, conferencing, IPTV, etc.

Industry 3(4)-screens Strategy (TV/Mobile/Desktop/(Car))
• Promotes the use of multimedia devices with broadband access (IPTV set-top box, SmartPhones, iPhone, etc.)
Background – Market Drivers
Why enterprises are investing in speech automated services

- Competitive global market require enterprises to reduce call center costs, increase agent productivity, and deliver better customer services experience.

- Voice-enabled IVRs provide more natural user navigation and shorter interactions resulting in a better user experience and higher call completion (examples: Travel Reservations, Finance, Directory asst, E-mail access).

- Natural Language Spoken Dialogue Systems address more challenging user’s intent interpretation relying on large vocabulary speech recognition and natural language understanding technologies (examples: customer care, help desks, e-commerce).

- Mobile broadband communication and web technology are enabling new information services and new revenue models based on advertising (pay per click / pay per call). User friendly interfaces are key to win market shares.
Human Computer Interaction Overview
Human Computer Interaction – An Interdisciplinary Challenge

- Electrical Engineering
  - Digital Signal Processing
  - Information Theory
- Computer Science
  - Artificial Intelligence
  - Algorithm
  - Software/Hardware
- Mathematics
  - Statistics
  - Machine Learning
- Linguistics
  - Theoretical
  - Computational
- Physics
  - Acoustics
- Psychology
  - Psycholinguistics
- Cognitive Science
  - Decision Theory
  - User State
- Human Computer Interface
- Social science
  - Anthropology

Ref. G. Riccardi, Interspeech 2007 Tutorial
2001: A Space Odyssey (1968)
Human-Human Communication

Rabiner and Juang “Fundamentals of Speech Recognition”, 1993
Fundamental Concepts

- Communication Channel (speech)
- Speech Recognition (speech-to-text)
- Speech Understanding (text-to-concepts)
- Dialogue (concept-to-action)
- Natural Language Generation (action-to-text)
- Speech Generation (text-to-speech)
- Hidden Context (users’ beliefs, discourse, intentional structure, attentional state, etc.)
Human-Human Communication ...
Speech Technology & Speech Services
Human Computer Interaction
Spoken Dialogue System

System Response

Caller’s Request

I would like to purchase a flight ...

Data

Words Spoken

TTS

Surface Realization

Which airport in New York city?

ASR

Text Plan (actions)

#RequestAirport (NYC)

SLG

DM

SLU

Meaning

@action=BUY
@origin=New York
Speech Services Complexity
Past, Present and (near) Future

- Simple ASR; isolated words, connected digits
  - Command And Control
    - (e.g., Simple call Routing; VRCP; Voice dialing)
  - Larger vocabulary, defined grammars

- Prompt Constrained Natural Language
  - (e.g., Travel Reservations, Finance, Directory asst, E-mail access)

- Spoken Language Understanding
  - (e.g., Customer Care, Help Desks, E-Commerce)

- Large vocabulary, NL, DM, TTS

- Voice Search
  - (e.g., Information Retrieval and Extraction, DA)

- Problem Solving
  - (e.g., Human-like Problem Solving, Troubleshooting)

1990, Complexity and Functionality

1 min

2009
Spoken Dialogue System Architecture
Automatic Speech Recognition (ASR) Challenges

ASR systems must cope with many sources of variability:

- **Inter-Speaker**
  - Physiological differences among speakers introduces variability in vocal tract characteristics

- **Acoustic Environment**
  - Street noise, car noise, background speech and music

- **Communication Channel**
  - Transducers, speech coders, linear channel effects, and fading

- **Intra-Speaker**
  - Prosody not well modeled in current ASR systems

- **Coarticulation**
  - Pronunciation variability caused by influence of surrounding articulatory events
Automatic Speech Recognition (ASR)
ASR Basic Components

Acoustic Models

- Store thousands of phoneme combinations. A phoneme is a basic unit of sound that makes up words. The “r” in red, bring, and round is a phoneme.

Dictionaries

- Specify how each word is pronounced in terms of the phonemes in the acoustic model.

Language Models

- **Rule-based grammar** explicitly defines the set of sentences and the order of words that can be recognized by the recognizer. WATSON supports two rule-based grammar formats: its own BNF (Backus–Naur form) and a standard XML format.

- **Statistical language model** (SLM) does not use explicit rules, but instead uses the statistical properties of thousands of transcribed utterances to help infer the language.
Grammars

- Specifies a set of utterances that a user may speak to perform an action or supply information, and
- For a matching utterance, returns a corresponding semantic interpretation. This may be a simple value (such as a string), a flat set of attribute-value pairs (such as day, month, and year), or a nested object (for a complex request).
- W3C Speech Recognition Grammar Specification (SRGS) comes in two interchangeable flavors: the Augmented BNF (ABNF) and the XML form
- ABNF is a refinement of the Java Speech Grammar Format JSFG
SRGS Example in ABNF

```xml
<grammar mode="voice" type="application/srgs">
   #ABNF 1.0;
   language en-US;
   mode voice;
   root $command;
       public $command = $action $object;
       $action = open | close | delete | move;
       $object = [the | a] (window | file | menu);
</grammar>
```
SRGS Example in XML Form

```xml
<grammar mode="voice" xml:lang="en-US" version="1.0"
root="command">
  <!-- Command is an action on an object --><!-- e.g. "open a window" -->
  <!-- Rule id="command" scope="public" -->
  <rule id="command" scope="public">
    <ruleref uri="#action"/>
    <ruleref uri="#object"/>
  </rule>

  <rule id="action">
    <one-of>
      <item>
        open </item>
      <item>
        close </item>
      <item>
        delete </item>
      <item>
        move </item>
    </one-of>
  </rule>

  <rule id="object">
    <item repeat="0-1">
      <one-of>
        <item> the </item>
        <item> a </item>
      </one-of>
    </item>
    <one-of>
      <item> window </item>
      <item> file </item>
      <item> menu </item>
    </one-of>
  </rule>

</grammar>
```
Migrating Speech to the Web
Web 2.0 memes cloud as constructed by Markus Angermeier on 11/11/2005
Web 2.0 Main Concepts

a) Web as Platform
   - Web services (SOAP, REST, scrAPIs, etc.)

b) Dynamic User Interfaces
   - Rich internet applications (AJAX)

c) Social Networks
   - blogs, wikis, CMS

\[ a + b \sim \text{web hybrids (mashups)} \]
Mashups (web hybrids)

- Mashups are **web applications** that leverage the compositional nature of public web services.
- A mashup is created when several **data** sources and **services** are "mashed up" (combined) to create a new service.
- Common **technologies** used in a mashup environment: SOAP, REST, AJAX, JavaScript, JSON (e.g., JavaScript Object Notation: lightweight data-interchange format), and various public web services (Google, Yahoo, Amazon, etc.).
- Service and data **aggregation** is typically done at the server level, but there is an increasing interest in providing web-based **composition engines** (Yahoo! Pipes, Microsoft Popfly, etc.).
Client Side Mashups
Server Side Mashups
Server-Side Mashup aggregators
Speech Mashups

- **Public API**
  - Speech resources accessible through simple REST interface (SOAP works as well) – no telephony
  - **Input**
    - HTTP header – User ID, speech recognition parameters
    - HTTP body – coded user’s speech
  - **Output**
    - XML/JSON/EMMA documents describing the recognition results, confidence scores, etc.

- **Audio capture**
  - Fat browser clients (full DOM & Javascript support) – IE, safari, firefox
  - Thin client – J2ME for mobile, .NET, Perl, Python, Java clients

- **Transport**
  - **Server** side support for sending/receiving **speech packets over HTTP**
    - similar to Real Time Streaming Protocol (RTSP), a session ID is used to keep track of the session when needed
  - **Client** side support for sending/receiving **speech packets over HTTP**
    - using AJAX pseudo-threading in the browser, or any other HTTP client technology

- **Speech Mashup Portal**
  - User management, resources control (disk quota, processing usage), logging, transcriptions
Watson Speech Mashup Architecture - Mobile

- **Audio Manager**
  - Captures the audio from the device in the native coding format
  - Communicates with the Watson ASR web service and other web applications

- **SOAP/REST Client**
  - Manages secure sockets, encrypted data transmission and allows to access SIM info

- **GUI Manager**
  - Abstracts the device graphical interface through J2ME Polish. Includes maps rendering and caching

- **Authentication & Security Manager**
  - Manages secure sockets, encrypted data transmission and allows to access SIM info

- **Trust Authority**
  - Trusted Application (access to SIM info)
    - Manually validated by AT&T
  - Signed Application
    - Automatically signed by AT&T

- **AT&T Trusted Authority**

- **Application Server**
  - WMM Servlet
    - Watson Wireline
    - Watson Speech
  - SOAP/REST Server
  - SOAP/REST Client
  - WSDL

- **WSDL**
  - User’s Profiles, Models & Grammars
    - Watson Server
    - Yahoo Maps
    - Movies Info
    - Yellow Pages

- **API Stub**
  - Web Services Description Language File - Published web service endpoint descriptor (e.g., API XML schema)
Watson Speech Mashup Architecture - Desktop

- **IE**
- **WMM**
  - .NET Audio Manager
  - AJAX Client
  - JS Watson
  - Captures the audio from the desktop audio interface
  - Communicates with the Watson ASR web service and other web applications
  - Manages the synchronization with Watson, audio capture and the GUI (DHTML)

- **Application Server**
  - WMM Servlet
  - SOAP (AXIS)/REST Server
  - SOAP/REST Client
  - Watson Wireline
  - Watson Server
  - Yahoo Maps
  - Movies Info
  - Yellow Pages
  - WSDL

- **User’s Profiles, Models & Grammars**
Watson Speech Mashup Architecture – OS X

- **Safari**
  - **WMM**
    - WebKit Audio Manager Plug-In
    - AJAX Client
    - JS Watson

- **Application Server**
  - WMM Servlet
    - SOAP (AXIS)/REST Server
    - SOAP/REST Client

- **Cascaded Diagram**
  - Communicates with the Watson ASR web service and other web applications
  - Captures the audio from the desktop audio interface
  - Manages the synchronization with Watson, audio capture and the GUI (DHTML)

- **Integration Points**
  - Safari
  - WMM
  - Application Server

- **External Services**
  - Watson Wireline
  - Yahoo Maps
  - Movies Info
  - Yellow Pages
  - User’s Profiles, Models & Grammars

**Diagram Note:**
- **HTTP** connection for data exchange.
Delegation Model

http://127.0.0.1:6666/?sid=jGS8NP5V86ZHu&cmd=start&url=www.yp.com
Watson Speech Mashup Architecture - IPTV

HTTP

Application Server

WMM Servlet

SOAP (AXIS)/REST Server

WSDL

User’s Profiles, Models & Grammars

Tasman Browser

Yahoo Maps

Movies Info

Yellow Pages

Watson Server

SOAP/REST Client

Watson Wireline

YELLOWPAGES.COM

Which City and State do you want to search?

Woodstock, GA 30188

Please say the name of the business:
walmart

2795 Chastain Meadows Pkwy
Marietta GA 30062

Wal-Mart Supercenter - Woodstock
Wal-Mart
Wal-Mart Supercenter

Wal-Mart Supercenter
Wal-Mart Supercenter
Wal-Mart
Wal-Mart Supercenter

walmart

Giuseppe Di Fabbrizio

AT&T Labs PROPRIETARY INFORMATION – Do not release without the author’s permission
# Watson REST API (request)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appid</td>
<td>string</td>
<td>Application ID, unique application identifier released to registered users</td>
</tr>
</tbody>
</table>
| cmd       | string | The ASR command string, any of the following:  
  - start – starts the ASR  
  - stop – stops the ASR and return the results  
  - audio – audio buffer available (in the HTTP body)  
  - compile – compile the specified grammar |
| control   | string | A sequence of Watson controls. Example:  
  ```
  set asr.coding = linear;
  set asr.byteOrder = LE;
  set asr.sampleRate = 8000;
  set config.nBest = 2;
  ``` |
| grammar   | string | Grammar reference (URL, local file, abnf source) |
# Watson REST API (response)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultSet</td>
<td>Contains all of the extracted terms.</td>
</tr>
<tr>
<td>Result</td>
<td>The text of each extracted term. Terms are returned in order of importance.</td>
</tr>
</tbody>
</table>
SM response

```xml
<entry>
    <key>interpretation</key>
    <value/>
</entry>
<entry>
    <key>norm_score</key>
    <value>32</value>
</entry>
<entry>
    <key>prompt_audio_time</key>
    <value>0.0</value>
</entry>
<entry>
    <key>reco</key>
    <value>new york new york</value>
</entry>
<entry>
    <key>reco_start_time</key>
    <value>1214492765.878</value>
</entry>
<entry>
    <key>slot.resultId</key>
    <value>1 : 0</value>
</entry>
<entry>
    <key>trigger</key>
    <value>userTimeout</value>
</entry>
<entry>
    <key>uttr_end_audio_time</key>
    <value>1.92</value>
</entry>
```
AT&T Watson Speech Mashup Portal

- User accounts management
- Direct access (outside firewall) to speech mashups API for registered users
- Quota control (disk space and number of hits/day/account)
- Built-in set of basic grammars
- Web-based grammar compiler
- Grammars and language model sharing (public/private)
- Real-time data logger
- Utterance recorder and online transcription tools
The Simplest SM Client

$ wget --post-file=audio-5496.amr --header 'Content-Type: audio/amr' --server-response
'http://ss-1:8080/smm2/watson?cmd=rawoneshot&grammar=test.lm&
uuid=EFB54DE1423111DD87B6F7306F54AB5F' -O response.xml

100%[============================================== ===============================>
2,049 --.--K/s

11:06:07 (16.18 MB/s) - `response.xml' saved [2049/2049]
iPizza – step by step example
0) service registration
iPizza – step by step example

1) speech grammar

<START> = [hes] [preamble] (<pizza> | <appetizers> | <wings> | <drinks> | <desserts> ) | <topmenu>;

hes = uh | um | ahh;

preamble = hi please | ([I want to [order]]) | I would like [a to order | place an order];

<pizza> = <pizzaname> [pizza];

<topmenu> =
    _{ "$='FeaturedItems'" } (featured items) _
    _{ "$='Specialties'" } (specialties) _
    _{ "$='SignatureCrust'" } (signature crust) _
    _{ "$='AppetizersSauces'" } (appetizer and sauces|apetizers|sauces) _
    _{ "$='WingStreetWings'" } (wingstreet wings|wings) _
    _{ "$='WingStreetWings'" } (wingstreet sides|sides) _
    _{ "$='DessertDrinks'" } (dresses and drinks|dresses | drinks) _
;

<pizzaname> =
    _{ "$='spcPizzaMiasm'" } (pizza mia | [a] [pizza] [with] mozzarella cheese and cheddar |)
    _{ "$='spcChesselovers'" } ([a] [pizza] [with] cheese [lover's] ) |
    _{ "$='spcMeatLovers'" } (meat [lover's] | [a] [pizza] ([with] italian sausages|sausage|bacon)
    [topping] ) |
    _{ "$='spcPepperoni'" } ([a] [pizza] [with] pepperoni ([lover's ]|[toppings] ) }
    _{ "$='spcVeggieLovers'" } (veggie [lover's] |...
iPizza – step by step example

2) upload and compile

The controls below allow for the manipulation of grammar files for your application. Files may be viewed, renamed, deleted or uploaded from your workstation. To upload multiple files, ZIP the files together and upload the zip file. The files will automatically be unZIPPed in the destination directory. Uploaded grammar files are automatically compiled, and this compilation log may also be viewed. Files listed in red have failed grammar compilation. You can also share or remove sharing on your grammar files, and provide descriptions for grammar files. Shared grammar files are listed in blue.

Grammar Context:

- View
- Delete File(s)
- View Log File
- Update Description
- Make Shared
- Remove Shared
- Rename

Upload File: [Browse] (maximum allowed size is 50 Mbytes)

Submit Upload  □ Overwrite
iPizza – step by step example

3) web frontend

```javascript
function onClick() {
    var speakButton = document.getElementById("speakButton");
    if (speakButton.innerHTML == 'Speak') {
        start();
        speakButton.innerHTML = 'Stop';
    } else {
        if (speakButton.innerHTML == 'Stop') {
            stop();
            speakButton.innerHTML = 'Speak';
            var results = asr();
            // alert(results);
            var interpretation = processAsrResponse(results);
            processUserChoice(interpretation);
        }
    }
}
</script>
</head>

<body onload="onLoad();">
    <embed name=audio id=audio type="audio/wav" width=1 height=1>
    <div id=info name=info>TEST</div>
    <div class=toolbar>
```
iPizza – step by step example
4) service usage / utterance transcription

<table>
<thead>
<tr>
<th>id</th>
<th>timestamp</th>
<th>filename</th>
<th>appname</th>
<th>field</th>
<th>grammar</th>
<th>reco</th>
<th>transcription</th>
</tr>
</thead>
<tbody>
<tr>
<td>14616</td>
<td>2008-06-25 17:15:07.000</td>
<td>20080625/audio-14616.amr</td>
<td>def001</td>
<td>business</td>
<td>pizza</td>
<td>pizza italian sausages</td>
<td></td>
</tr>
<tr>
<td>14617</td>
<td>2008-06-25 17:15:14.000</td>
<td>20080625/audio-14617.amr</td>
<td>def001</td>
<td>business</td>
<td>pizza</td>
<td>uh veggie pizza</td>
<td></td>
</tr>
<tr>
<td>14618</td>
<td>2008-06-25 17:15:23.000</td>
<td>20080625/audio-14618.amr</td>
<td>def001</td>
<td>business</td>
<td>pizza</td>
<td>Diet Pepsi</td>
<td></td>
</tr>
<tr>
<td>14619</td>
<td>2008-06-25 17:15:30.000</td>
<td>20080625/audio-14619.amr</td>
<td>def001</td>
<td>business</td>
<td>pizza</td>
<td>buffalo wings</td>
<td></td>
</tr>
</tbody>
</table>
(Semantic Web Integration)
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:vra="http://purl.org/vra/3.5/"
rdf:seeAlso="http://www.w3.org/2001/sw/BP/IM/vracro3.rdf"
>

<!-- Description of the painting -->

<vra:Work rdf:about="#painting;">

<!-- General information -->
<vra:title>Jardin à Sainte-Adresse</vra:title>
<vra:title.translation>Garden at Sainte-Adresse</vra:title.translation>
<vra:creator>Monet, Claude</vra:creator>  <!-- ULAN ID: 500019484 -->
<vra:creator.role>artist</vra:creator.role>  <!-- ULAN ID: 31100 -->
<vra:date.creation>1867</vra:date.creation>

<!-- Technical information -->
<vra:measurements.dimensions>98.1 x 129.9 cm</vra:measurements.dimensions>
<vra:material.support>unprimed canvas</vra:material.support>  <!-- AAT ID: 300238097 -->
<vra:material.medium>oil paint</vra:material.medium>  <!-- AAT ID: 300015050 -->
<vra:Type>oil paintings</vra:Type>  <!-- AAT ID: 300033799 -->
<vra:technique>oil painting (technique)</vra:technique>  <!-- AAT ID: 300178684 -->

<!-- Associated style etc -->
<vra:stylePeriod>Impressionist</vra:stylePeriod>  <!-- AAT ID: 300021503 -->
<vra:culture>French</vra:culture>  <!-- AAT ID: 300111188 -->

<!-- Subject matter: (who/what is depicted by this work -->
<vra:subject>Jeanne-Marguerite Lecadre (artist's cousin)</vra:subject>
<vra:subject>Madame Lecadre (artist's aunt)</vra:subject>
<vra:subject>Adolphe Monet (artist's father)</vra:subject>

<!-- Provenance -->
<vra:location.currentSite>Metropolitan Museum of Art, New York</vra:location.currentSite>
<vra:location.formerSite>Montpellier</vra:location.formerSite>
</vra:Work>
</rdf:RDF>
Demonstrations
iPizza
YellowPages.com
iMOD – Multimodal Movie on Demand

Ref. Michael Johnston (johnston@research.att.com)
Watson Speech MASHUP - DA Prototype
Watson Speech MASHUP - DA Prototype
Final Remarks

• Speech (and Multimodal) technologies are mature to provide robust performances in a scalable fashion
• Broadband wireless network availability makes possible to concentrate speech processing in the network
• Web mashup paradigm simplifies web and speech integration