“TEXT-TO-SPEECH ENGINES AS TELECOM SERVICE ENABLERS”


* ERICSSON Telecommunications Romania S.R.L.
** Faculty of Electronics, Telecommunications and IT, University “Politehnica” of Bucharest, ROMANIA
Text-to-Speech Engines as Telecom Service Enablers

**OUTLINE:**

- Service types
- Implementation view
- Network view
- Making use of open protocols
- E-mail reader platform
- Carrier-grade platform
- TTS algorithm implementation issues
- Conclusions
Voiced interaction with non-human peer is a complex task involving all speech technology domains.

The areas of interest are completely independent: database info retrieval, navigation aid or health & public administration.

Speech services has similar requirements to fulfill. This could be unified by abstracting speech technology as telecom enablers.
Service types

- Simple taxonomy of TTS based services
  - Notify/warning services
    - News reading or location based traffic information
    - The information is pushed to the terminal
  - Legacy terminal adaptation
    - Reading e-mail or SMS from legacy POTS terminals
    - The user pop the information from the source
- Accessibility
The services could be implemented as

- **Embedded system**
  - **Pros:** low bandwidth requirements
  - **Cons:** terminal resources, intellectual property

- **Network-based systems**
  - **Pros:** convergent services
  - **Cons:** higher communication costs for end-user (but these costs are constantly dropping)

- **Mixed solutions**
Network view

- Networks are evolving to all-IP, but slowly and costly
- The TTS based services must rely on open and future-proof protocols
- SIP/RTP based protocols such as MRCP are quite suited
Making use of open protocols

- The client requires the generation and/or consumption of media streams
- Media Resource Server has the relevant resources to process the input stream or to generate the output stream: synthesis engines, ASR engines, SV/SI engines
E-mail reader platform (1)

- Single threaded platform
- Permits to listen e-mail messages on legacy POTS terminals, being a simple example of convergent service

Useful for
- Verifying the end-to-end concept with minimal resources
- Receiving feedback from end-user about speech quality
E-mail reader platform (2)

- **TTS Engine**
  - C-based TTS algorithms (synthesis in Romanian language)

- **HTTP server**

- **Media Server**
  - Collection of Perl scripts for
    - Service control
    - E-mail connector
    - POTS connector
E-mail reader platform (3)

A typical call:

1) Platform is triggered by calling an access number
2) E-mail connector opens a session to the predefined e-mail account
3) Several options are presented to the user during an interactive voice response session
4) Service logic opens an HTTP connection to the TTS engine and sends the text to be translated
5) The media server will re-encode the audio payload from the HTTP response and will fetch it to the POTS connector
Carrier-grade platform

- **Features:**
  - Multi-threaded and multi-process on carrier grade hardware and software platform

- **Layered architecture:**
  - Horizontal processing layer: speech engines, communication and middle layer
  - Vertical layer: provisioning, O&M, statistics

- Intra-operator versus hosting deployments
Conclusions – 1

- Speech services could be unified as telecom enablers
- Making use of the new enablers to build convergent services
- Implementation leveraging open protocols for future-proof solutions
- TTS technology – an important component in network-based applications development
- We developed and end-to-end e-mail reader application using a proprietary TTS system in Romanian
TTS algorithm implementation issues

- An e-mail or SMS reader application needs to meet an important constraint: the missing diacritics problem
- Usually, the majority of users still disregard the diacritics
- Synthesizing a text generated without diacritics generally leads to a poor intelligibility
- The automatic restoration of diacritics is a difficult problem, as there are not evident linguistic rules to accomplish this task
Specific features in Romanian language

- Romanian language makes use of three diacritic marks (a breve, a circumflex accent, and a cedilla), leading to five letters with diacritics: ă, â / î, ş and ţ.

- Some diacritics indicate only a different noun form (e.g., casă – house, and its pair casa – the house), others lead to a distinct meaning (e.g., fata – the girl, but fața – the face).

- The percentage of words written with diacritics in a Romanian text is substantial: between 25% and 40% of the total number of words.

- An interesting particularity: there are words always written with diacritics (câteva – some, științific – scientific), and also words where some of the diacritics are always present (cămașă / cămașa – shirt / the shirt).
An automatic diacritic restoration algorithm in Romanian (1)

- The algorithm is based on a hybrid (dictionary and rule set) approach.
- A large electronic dictionary of the most used Romanian words (D1) was first built; it contains more than 120,000 words.
- Three dictionaries were then iteratively inferred. The last one (D4) contains all the possible forms for each word, indexed twice according to the basic position of the word in the dictionary and to the number of the possible diacritic patterns. A few examples:

<table>
<thead>
<tr>
<th>Word in D4</th>
<th>Index $n$</th>
<th>Index $i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>câteva</td>
<td>9514</td>
<td>cateva câteva</td>
</tr>
<tr>
<td>ştiinţific</td>
<td>56240</td>
<td>stiintific ştiinţific</td>
</tr>
<tr>
<td>două</td>
<td>18594</td>
<td>doua doua două</td>
</tr>
<tr>
<td>cămaşa</td>
<td>8326</td>
<td>camasa cămaşă cămaşa</td>
</tr>
<tr>
<td>până</td>
<td>41624</td>
<td>pana pana până</td>
</tr>
<tr>
<td>rama</td>
<td>48543</td>
<td>rama rama râmă</td>
</tr>
<tr>
<td>țara</td>
<td>58334</td>
<td>tara tară țara țără</td>
</tr>
</tbody>
</table>
An automatic diacritic restoration algorithm in Romanian (2)

- The algorithm is part of the preprocessing module in the TTS system
- The incoming word is searched first in D4. If it is not found, the word remains unchanged. If it is found, it is processed in accordance with the maximum value of index $i$, either automatically assigning the diacritics that are always present, or making use of rules based on the word context
Conclusions – 2

- The requirement for restoring missing diacritics to text is a common problem for most languages that use the Latin alphabet.

- The overall accuracy of the proposed algorithm is currently about 94% (tests were performed on three texts containing about 12,000 words; each diacritic missed or incorrectly assigned was considered as an error).

- We anticipate even better results by increasing the dictionary size and by supplementary using a morphological analysis for word disambiguation.