Multilingual Speech Processing

Chapters 2, 4, 6, 7
Outline

- Motivation and Introduction
  - Why Speech Processing in many Languages?
  - What are the Challenges and possible Solutions?
- LVCSR in many languages
- Truly Multilingual Speech Interfaces
  - Multilingual Acoustic Modeling
- Rapid Language Adaptation
  - Acoustic Models
  - Automatic Dictionary Generation
  - Language Models
  - Synthesis
- SPICE: web-based tools for rapid adaptation
Outline

- Motivation and Introduction
  - Why Speech Processing in many Languages?
  - What are the Challenges and possible Solutions?
- LVCSR in many languages
- Truly Multilingual Speech Interfaces
  - Multilingual Acoustic Modeling
- Rapid Language Adaptation
  - Acoustic Models
  - Automatic Dictionary Generation
  - Language Models
  - Synthesis
- SPICE: web-based tools for rapid adaptation
Pronunciation rules:
- hi /h//ai/
- you /j/u/
- we /w//i/

Input: Speech

Text data

Phone set & Speech data

Speech Processing Systems

AM
Lex
LM

NLP / MT

TTS

Output: Speech & Text
Speech Recognition in 17 Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Error Rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japanese</td>
<td>10</td>
</tr>
<tr>
<td>German</td>
<td>11.8</td>
</tr>
<tr>
<td>English</td>
<td>14</td>
</tr>
<tr>
<td>Thai</td>
<td>14</td>
</tr>
<tr>
<td>Korean</td>
<td>14</td>
</tr>
<tr>
<td>Ch-Mandarin</td>
<td>14</td>
</tr>
<tr>
<td>Turkish</td>
<td>14.5</td>
</tr>
<tr>
<td>French</td>
<td>16.9</td>
</tr>
<tr>
<td>Portuguese</td>
<td>18</td>
</tr>
<tr>
<td>Croatian</td>
<td>19</td>
</tr>
<tr>
<td>Spanish</td>
<td>20</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>20</td>
</tr>
<tr>
<td>Russian</td>
<td>29</td>
</tr>
<tr>
<td>Afrikaans</td>
<td>33.5</td>
</tr>
<tr>
<td>Chinese</td>
<td>20</td>
</tr>
<tr>
<td>Arabic</td>
<td>21.7</td>
</tr>
<tr>
<td>Iraqi</td>
<td>23.4</td>
</tr>
<tr>
<td>Croationian</td>
<td>29</td>
</tr>
</tbody>
</table>

Word Error Rate [%]
Polyphone Numbers for 10 Languages

Phonotactics

Anzahl der Sub-Polyphone vs. Kontextbreite for 10 languages:
- Chinesisch
- Deutsch
- Französisch
- Kroatisch
- Portugiesisch
- Spanisch
- Türkisch
- Japanisch
- Englisch
- Koreanisch
Outline

- Motivation and Introduction
  - Why Speech Processing in many Languages?
  - What are the Challenges and possible Solutions?
- LVCSR in many languages
- Truly Multilingual Speech Interfaces
  - Multilingual Acoustic Modeling
- Rapid Language Adaptation
  - Acoustic Models
  - Automatic Dictionary Generation
  - Language Models
  - Synthesis
- SPICE: web-based tools for rapid adaptation
Truly Multilingual ASR System
Can we build a language independent ASR system?

- **Language Independent Acoustic Modeling**
  - International Phonetic Alphabet (IPA)
    - simple to implement, easy to port to other languages
   - Fully data-driven procedure
    - considers spectral properties and statistical similarities
    - apply to context independent and dependent models

- **Universal Language Model**
  - Combine LMs of languages to allow code switching

- **Approach:**
  - Train language dependent and independent AM+LM
  - Evaluate in monolingual and multilingual mode
Speech Recognition in Multiple Languages

**Goal:** Speech recognition in a many different languages

**Problem:** Data collection is time consuming and expensive

---

Sound system
Speech data
(≥ 10 hours)

<table>
<thead>
<tr>
<th>AM</th>
<th>Lex</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ela /e/1/a/</td>
<td>eu /e/u/</td>
<td>sou /s/u/</td>
</tr>
<tr>
<td>eu sou</td>
<td>você é</td>
<td>ela é</td>
</tr>
</tbody>
</table>

---

Pronunciation rules
Text data
Speech Recognition in Multiple Languages

- Sound system
- Speech data

Pronunciation rules:
- \( /e/l/a/ \)
- \( /e/u/ \)
- \( /s/u/ \)

Text data:
- \( \text{ela} /e/l/a/ \)
- \( \text{eu} /e/u/ \)
- \( \text{sou} /s/u/ \)

AM
Lex
LM
Multilingual Acoustic Modeling

Step 1:
- Uniform multilingual speech database
- Monolingual recognizers in many languages

Step 2:
- Combine monolingual acoustic models
- Share data across languages
Acoustic Model Combination

Sound production is human not language specific:

→ International Phonetic Alphabet (IPA)

1) Build **Universal** sound inventory based on IPA
485 sounds are reduced to 162 IPA-sound classes

2) Each sound class is represented by one “phoneme”
which is trained through **data sharing** across languages
- \(m,n,s,l\) occur in all languages
- \(p,b,t,d,k,g,f\) and \(i,u,e,a,o\) occur in almost all languages
- no sharing of triphthongs and palatal consonants
Acoustic Model Combination

2) Each sound class is represented by one “phoneme” which is trained through **data sharing** across languages

- **m,n,s,l** occur in all languages
- **p,b,t,d,k,g,f** and **i,u,e,a,o** occur in almost all languages
- no sharing of triphthongs and palatal consonants
Acoustic Model Combination

ML-Mix

ML-Sep

ML-Tag
Acoustic Model Combination

Croatian
Japanese
Spanish
Turkish

Word Error Rate [%]

Mono  ML-Tag7500  ML-Tag3000  ML-Mix3000

27  30  32  35
13  14  15  20
28  30  32  37
20  21  21  29
Outline

- Motivation and Introduction
  - Why Speech Processing in many Languages?
  - What are the Challenges and possible Solutions?
- LVCSR in many languages
- Truly Multilingual Speech Interfaces
  - Multilingual Acoustic Modeling
- Rapid Language Adaptation
  - Acoustic Models
  - Automatic Dictionary Generation
  - Language Models
  - Synthesis
- SPICE: web-based tools for rapid adaptation
Challenges

- Algorithms are language independent but require data
  - Dozens of hours audio recordings and corresponding transcriptions
  - Pronunciation dictionaries for large vocabularies (>100,000 words)
  - Millions of words written text corpora in various domains in question
  - Bilingual aligned text corpora

- BUT: Such data are only available in very few languages
  - Audio data ≤ 40 languages, Transcriptions take up to 40x real time
  - Large vocabulary pronunciation dictionaries ≤ 20 languages
  - Small text corpora ≤ 100 languages, large corpora ≤ 30 languages
  - Bilingual corpora in very few language pairs, pivot mostly English

- Additional complications:
  - Combinatorical explosion (domain, speaking style, accent, dialect, ...)
  - Few native speakers at hand for minority (endangered) languages
  - Languages peculiarities, social factors
Solution: Learning Systems

⇒ Intelligent systems that learn a language from the user

- Effizient learning algorithms for speech processing
  - Learning:
    - Interactive learning with user in the loop
    - Statistical modeling approaches
  - Efficiency:
    - Reduce amount of data (save time and costs): by a factor of 10
    - Speed up development cycles: days rather than months
  ⇒ Rapid Language Adaptation from universal models

- Bridge the gap between language and technology experts
  - Technology experts do not speak all languages in question
  - Native users are not in control of the technology
Speech Processing:
Interactive Creation and Evaluation toolkit

- National Science Foundation, Grant 10/2004, 3 years
- Principle Investigator Tanja Schultz

- Bridge the gap between technology experts → language experts
  - Automatic Speech Recognition (ASR),
  - Machine Translation (MT),
  - Text-to-Speech (TTS)

- Develop web-based intelligent systems
  - Interactive Learning with user in the loop
  - Rapid Adaptation of universal models to unseen languages

- SPICE webpage http://cmuspice.org
Welcome to SPICE

Getting started
Welcome to SPICE homepage. In this page, you can develop the speech system (including Automatic Speech Recognition and Text-To-Speech) specific to your own language.

First thing to your attention:

Please ENABLE YOUR BROWSER TO ACCEPT COOKIES because in SPICE web site, we use cookies to identify different users and different projects. If your browser has not been enabled to accept cookies, then you cannot login and use SPICE at all.

Please input your name (Developer Name) and the name of your language (Language Name) and the name of the Project (Project Name) below. The basic idea is that each developer can work on several different languages, and in each language you can develop several different version of your speech system, differentiated by "Project Name".

You don't have to finish all the work at one time. For example, you can do Acoustic Model at one day, Language Model tomorrow ... etc. Every time you come back to SPICE page, Use your "Developer Name", "Language Name" and "Project Name", you can locate your previous work and continue to do the rest.

Developer Name: 

Language Name: 

Project Name: 

Submit
Speech Processing Systems

Phone set & Speech data

Pronunciation rules

Text data

Input: Speech

AM
Lex
LM

NLP / MT

TTS

Output: Speech & Text

Hello
Input: Speech

Phone set & Speech data

Hello

AM | Lex | LM | NLP / MT | TTS

Output: Speech & Text
Rapid Portability: Acoustic Models

Phone set & Speech data

Input: Speech

AM
Lex
LM
NLP
MT
TTS

Output: Speech & Text

Hello

สวัสดี ด้วย

AM
Lex
LM
NLP
MT
Speech Production is independent from Language $\Rightarrow$ IPA

1) IPA-based **Universal Sound Inventory**

2) Each sound class is trained by **data sharing**

- Reduction from 485 to 162 sound classes
- $m,n,s,l$ appear in all 12 languages
- $p,b,t,d,k,g,f$ and $i,u,e,a,o$ in almost all

**Problem:**
Context of sounds are language specific
How to train context dependent models for new languages?

**Solution:**
1) Multilingual Decision Context Trees
2) Specialize decision tree by Adaptation
Polyphone Coverage

The graph illustrates the weighted coverage of Portuguese monophones, triphones, and quinphones across different languages. The x-axis represents the number of involved languages, while the y-axis shows the polyphone coverage in percentage.

- **SP**: Spanish
- **FR**: French
- **GE**: German
- **TU**: Turkish
- **JA**: Japanese
- **KR**: Korean
- **EN**: English
- **KO**: Korean
- **CH**: Chinese

The lines indicate the coverage as the number of involved languages increases.
Context Decision Trees

- Context dependent phones ($\pm n = \text{polyphone}$)
- Trainability vs Granularity
- Divisive clustering based on linguistic motivated questions
- One model is assigned to each context cluster
- Multi-lingual case: should we ignore language information?
- Depends on application
- Yes in case of adaptation to new target languages
Rapid Language Adaptation

Model mapping to the target language
1) Map the multilingual phonemes to Portuguese ones based on the IPA-scheme
2) Copy the corresponding acoustic models in order to initialize Portuguese models

Problem:
Contexts are highly language specific, how to apply context dependent models to a new target language

Solution:
1) Train a multilingual polyphone decision tree
2) Specialize this tree to target language using limited data (Polyphone decision tree specialization - PDTS)
Polyphone Decision Tree Specialization

English Polyphone Tree
Polyphone Decision Tree Specialization

English

Other languages
Polyphone Decision Tree Specialization

Polyphones found in Portuguese
Polyphone Decision Tree Specialization

1. Tree Pruning
Select from all polyphones only the ones which are relevant for the particular language
Polyphone Decision Tree Specialization

2. Tree
Regrowing
Further
specialize the
tree according to
the adaptation
data
Rapid Portability: Acoustic Model

Word Error rate [%]

- Ø Tree
- ML-Tree
- Po-Tree
- PDTS

0 0:15 0:25 1:30 16:30

69.1 57.1 49.9 40.6 32.8 28.9 19.6 19
**Phoneme set specification**

This is a tool which will display all IPA phoneme. As a naive user, you can choose and give names to phonemes you wish your Speech Engine to use. After you have finished, you can click the "Submit" button to create the new acoustic model on the fly.

Current PhoneMapFile

Consonants (Pulmonic): Please choose the consonant sounds you’d like to have in your new acoustic models by giving it a name in the textbox next to it.

<table>
<thead>
<tr>
<th></th>
<th>Bilabial</th>
<th>Labiodental</th>
<th>Dental</th>
<th>Alveolar</th>
<th>Postalveolar</th>
<th>Retroflex</th>
<th>Palatal</th>
<th>Velar</th>
<th>Uvular</th>
<th>Pharyngeal</th>
<th>Glottal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Plosive</strong></td>
<td>p</td>
<td>p2</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>t</td>
<td>t'</td>
<td>d</td>
<td>d'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>p</td>
<td>p'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rapid Portability: Pronunciation Dictionary

Pronunciation rules

Input: Speech

Output: Speech & Text

AM  Lex  LM  NLP / MT  TTS

Hello

現在广播完了

Hello

"adios"  →  /a/ /d/ /i/ /o/ /s/

"Hallo"  →  /h/ /a/ /l/ /o/  →  ???
* Follow the work of Davel & Barnard

* Word list: extract from text

* G-2-P
- explicit map rules
- neural networks
- decision trees
- instance learning (grapheme context)

* Update after each $w_i$ → effective training
Currently you are logging in as tanja, you are working on language German, Project Name = 1

We use an iterative procedure to gather information to create dictionary for you.

First of all, please input an initial Grapheme to Phoneme (G2P) rule of your language.

Based on this rule, our system will "guess" the correct pronunciation of words in your language. You are able to view the predicted pronunciation, change it, delete it, or type a correct pronunciation for this word. The correct pronunciation will be saved into your dictionary and our system will make use of this information to make a better "guess" in predicting pronunciation of new words.

Now please type in Grapheme to Phoneme rule (G2P) for us. Just type one of the most common pronunciation for each grapheme. Thanks.

- e [ ] uppercase [x] lowercase [ ] punctuation mark [ ] number [ ] others
- n [ ] uppercase [ ] lowercase [ ] punctuation mark [x] number [ ] others
- r [ ] uppercase [ ] lowercase [ ] punctuation mark [x] number [ ] others
- i [ ] uppercase [ ] lowercase [ ] punctuation mark [ ] number [ ] others
- t [ ] uppercase [ ] lowercase [ ] punctuation mark [ ] number [ ] others
- s [ ] uppercase [ ] lowercase [ ] punctuation mark [ ] number [ ] others
- a [ ] uppercase [ ] lowercase [ ] punctuation mark [ ] number [ ] others
new word:

Merck

system suggested pronunciation: merck

Listen to it  Accept Pronunciation

If you want to skip this word and work on it later, please click

Skip this word

If you don't think it's a valid word in your language, please click

Remove this word
Issues and Challenges

- How to make best use of the human?
  - Definition of successful completion
  - Which words to present in what order
  - How to be robust against mistakes
  - Feedback that keeps users motivated to continue

- How many words to be solicited?
  - G2P complexity depends on the language (SP easy, EN hard)
  - 80% coverage
    hundred (SP) to thousands (EN)
  - G2P rule system perplexity

<table>
<thead>
<tr>
<th>Language</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>50.11</td>
</tr>
<tr>
<td>Dutch</td>
<td>16.80</td>
</tr>
<tr>
<td>German</td>
<td>16.70</td>
</tr>
<tr>
<td>Afrikaans</td>
<td>11.48</td>
</tr>
<tr>
<td>Italian</td>
<td>3.52</td>
</tr>
<tr>
<td>Spanish</td>
<td>1.21</td>
</tr>
</tbody>
</table>
Rapid Portability: LM

Resource rich languages ↔ Resource low languages:

Input: Speech

AM
Lex
LM

Hello

NLP / MT

TTS

Output: Speech & Text
Crosslingual LM Adaptation

Internet / TV contemporaneous articles

Cross-Lingual IR

ASR

Ported Paschto AMs

Grapheme based Paschto dictionary

Baseline Paschto Language Model

Aligned Transcripts

Translation lexikon

SMT

Paschto Language Model

Build language model

Build a language model using either a web ‘crawler’ or by uploading a file

We will try to grab text data from internet and build language model from those text data.

Given a link to some webpage composed of your own language, our web spider can automatically go into this web page, grab all text from this page, and also follow the hyperlink inside this page then go to another linked page to collect more text data. We need huge text data in order to build a robust language model. As we often said in speech research community, “no data is like more data”

Please provide us a URL of webpage which is composed by your language:

Please specify how many levels would you like our web spider to get into:

- 1
- 2

Please verify that the URL you gave above is correct and valid, select the level you want the web spider to get. The deeper the value is, the longer it takes to finish the task. So our suggestion is to select a smaller value unless you care about the amount of training text more than the time to finish the task.

Another way is to submit a text file from your local machine. This file should contains large amount of text of your language. We can train Language Model from this text as well. But if you want to submit a text file, please make it such that every line contains a single sentence. This is the standard input format our LM toolkit will take.

You can also train LM via both method. Just input both the URL and the file name. Our toolkit will use both information.

Choose a file to upload:  

Browse...
Rapid Portability: TTS

Phone set & Speech data

Input: Speech

Hello

AM
Lex
LM

NLP
/MT

TTS

Output: Speech & Text
Parametric TTS

- Text-to-speech for G2P Learning:
  - Technique: phoneme-by-phoneme concatenation, speech not natural but understandable (Marelie Davel)
  - Units are based on IPA phoneme examples
    - PRO: covers languages through simple adaptation
    - CONS: not good enough for speech applications

- Text-to-speech for Applications:
  - Common technologies
    - Diphone: too hard to record and label
    - Unit selection: too much to record and label
  - New technology: **clustergen** trajectory synthesis
    - Clusters representing context-dependent allophones
    - PRO: can work with little speech (10 minutes)
    - CONS: speech sounds buzzy, lacks natural prosody
Manual speaker selection

⇒ For all languages monolingual TTS performs best
⇒ Multilingual Models perform well …
  … only if knowledge about language is preserved (Multi+)
  (only small amount of sharing actually happens)
SPICE: Afrikaans – English

- Goal: Build Afrikaans – English Speech Translation System with SPICE
  - Cooperation with University Stellenbosch and ARMSCOR
  - Bilingual PhD visited CMU for 3 months
  - Afrikaans: Related to Dutch and English, g-2-p very close, regular grammar, simple morphology

- SPICE, all components apply statistical modeling paradigm
  - ASR: HMMs, N-gram LM (JRTk-ISL)
  - MT: Statistical MT (SMT-ISL)
  - TTS: Unit-Selection (Festival)
  - Dictionary: G-2-P rules using CART decision trees

- Text: 39 hansards; 680k words; 43k bilingual aligned sentence pairs;
  Audio: 6 hours read speech; 10k utterances, telephone speech (AST)
Good results: ASR 20% WER; MT A-E (E-A) Bleu 34.1 (34.7), Nist 7.6 (7.9)

Shared pronunciation dictionaries (for ASR+TTS) and LM (for ASR+MT)

Most time consuming process: data preparation → reduce amount of data!

Still too much expert knowledge required (e.g. ASR parameter tuning!)

Time Effort

![Bar chart showing time effort for different tasks: Data (11 days), Training (3 days), Tuning (7 days), Evaluation (5 days), Prototype (5 days). The legend includes AM (ASR), Lex, LM (ASR, MT), TM (MT), TTS, S-2-S.]
Conclusion

- Intelligent systems to learn language
  - SPICE: Learning by interaction with the (naive) user
  - Rapid Portability to unseen languages
- Truly Multilingual Systems
  - Systems and data in multiple languages
  - Universal language independent models
- Other Applications and Issues in Multilinguality
  - Speech translation (see MT Vorlesung)
  - Multilingual Interfaces
    - Code switching, non-native / accented speech
  - Alternative model units (AF, Graphemes)