

Databases for speech synthesis

- Case study: using a keyword lexicon (Unisyn)

Phonemes, phones, accent variation and the IPA

- In the lab, many of you may have realised that none of the provided lexicons (a.k.a. pronunciation dictionaries) is a good match to the way *you* speak
 - Why?
 - What effect will this have?
- Inter-speaker variation
 - tuning the lexicon to an individual speaker (or group of speakers, e.g. an accent)
- Intra-speaker variation
 - dealing with a speaker's individual variation

Recap: lexicons and letter-to-sound

- Lexicon:
 - “lives” nns ((l ai v z) 1)
 - “lives” vbz ((l i v z) 1)
- LTS “rules” are often a model learned from data
 - e.g., classification tree (CART)
 - predicts the pronunciation for each grapheme-in-context

Obvious: speakers say things differently!

Less obvious: find systematic patterns we can exploit

	British English	Canadian English
par	pɑ	pɑɪ
pa	pɑ	pɑ
paw	pɔ	pɑ
pore	pɔ	pɔɪ
pour	pɔ	pɔɪ
poor	pɔ	pɔɪ

Keyword lexicons (Wells, 1982), (Fitt & Isard, 1999)

- Rather than using phonetic symbols in the lexicon, **key symbols** are used
- A key symbol is defined for each sound that behaves differently across different accents
- The base lexicon contains pronunciations in terms of key-symbols, with every possible distinction encoded
- A set of pronunciation rules maps the key symbol representation into a surface representation of phonetic symbols for a given accent
 - e.g., multiple key-symbols merge into one surface symbol
- Each key-symbol is characterised by a keyword. This is an example word that exhibits the behaviour that the key-symbol is designed to represent.

Lexical sets

A keyword lexicon

	Key-symbol entry	British English	Canadian English
par	p ar r	pɑ	pɑɪ
pa	p aa	pɑ	pɑ
paw	p oo	pɔ	pɑ
pore	p our r	pɔ	pɔɪ
pour	p our r	pɔ	pɔɪ
poor	p ur r	pɔ/pɔɪ	pɔɪ

rules :

keysymbol	British English	Canadian English
r	-	ɹ
ar	ɑ	ɑ
oo	ɔ	ɑ

More details

- Rules can be hierarchical
 - Rhoticity rule for all rhotic accents
 - Specific rules for just one accent, or just one speaker
- Some rules need to be post-lexical (e.g., if they operate across word boundaries)
 - “My father is ...” in RP (r → /ɹ/ and is not deleted)
- Rule sets are generally specified for specific accents, but can be tuned for individual speakers

Keyword lexicons in practice

- Festival uses Unisyn or Combilex
 - large, manually-written lists of words and pronunciations in terms of **key symbols**
 - plus small set of rules to generate surface forms
- Complete **surface form** lexicons can be generated for specific accents - e.g., unilex-rpx, unilex-gam, unilex-edi
- Editing a keyword lexicon is very skilled
 - Adding a new word is harder than with a traditional lexicon, because the pronunciation is in key-symbols, not phonemes
 - Adding a new accent may be **very** hard, if it involves new distinctions not already encoded

The screenshot shows the Edinburgh Research and Innovation website. The header includes the logo and tagline "A UK leader in commercialising research and entrepreneurship". A navigation bar contains "Home » Software » Combilex GA". The main content area features the product title "Combilex GA" and subtitle "lexicon for the General American Pronunciation accent of English". Below this are social media sharing buttons for Facebook, Twitter, Google+, and LinkedIn. An "OPTIONS" section includes a printer icon and a warning: "Please check carefully that the terms you select correspond to your intended use of the product." A "Commercial Licence Agreement for Combilex GA" is highlighted with a "View Terms" button. The footer of the page states "Non-exclusive licence for internal product development".

Databases for speech synthesis

- Group activity: design the script to be recorded

Group activity: design the script to be recorded

- Step 1
 - find a **source** of text
 - things to consider include: copyright, domain, size, readability, NSWs, ...
- Step 2
 - **clean** the text
 - things to consider include: vocabulary, readability, normalisation, ...
- Step 3
 - design a simple '**richness**' measure
 - write this as a function that computes a score for one sentence