1 TTS Architectures

1.1 Linear Architectures

- Data: strings / lists
- Processing model: serial
- Advantages:
  - Often simple and therefore fast
  - Modularity enforced by processing model
- Disadvantages:
  - Linear data structures do not capture linguistic phenomena well
  - Strict modularity is perhaps too strong a requirement

1.2 Hierarchical Architectures

- Data: shared, multilevel (MLDs / FSs)
- Processing model: serial (pseudo-parallel)
- Advantages:
– Simple, fast
– Modular
– Shared data eliminates redundancy

• Disadvantages:
  – Strict modularity is perhaps too strong a requirement

1.3 Heterarchical Architectures

• Data: shared, multilevel (MLDs / FSs)
• Processing model: parallel
• Advantages:
  – Shared data eliminates redundancy
  – Parallel model allows data exchange between subprocesses
• Disadvantages:
  – Complex, often slow.

2 NLP Block

2.1 Preprocessor

Tokenizes input text, identifies acronyms, markup, etc.

2.2 Morphological Analyzer

Segments input tokens into roots and suffixes, possibly restricting range of choice for subsequent steps.

2.3 Part-of-Speech Tagger

Assigns part-of-speech label to each input token; can disambiguate some heterophonic homographs (e.g. “record” noun vs. verb).

2.4 Syntactic-Prosodic Parser

Generates prosodic groups by use of PoS, syntactic, and/or semantic information.
Figure 2: General architecture of the NLP block
2.5 Letter-to-Sound Transducer

Generates a broad phonetic transcription (phone string) from analysed tokens.

2.6 Prosody Generator

Computes narrow phonetic transcription (phone durations, F0 envelopes, etc.) from output of previous stages.