1 Text Preprocessing

1.1 Tokenization

- **Task:** Split arbitrary input text into (linguistically salient) *tokens*.
- **Motivation:** Dealing with unrestricted input texts in a holistic manner is unfeasible.
- **Method:** Traditionally, whitespace and punctuation have been used to identify token boundaries.
- **Problems:**
  - Numbers ("42.24", "1,001"),
  - Times ("4:20", "15:00")
  - Abbreviations ("M.I.T.", "Ph.D.")
  - Collocations & Idioms ("New Haven", "kind of")
- **Workaround:** 2-stage analysis
  - *Stage 1:* Identify *broad segmentation units (BSUs)*: split input on all punctuation characters and whitespace subsequences.
  - *Stage 2:* Identify *final segmentation units (FSUs)*: map BSU subsequences to FSUs.
- **Example**
  - *Input:*
    1,000 Ph.D.s will meet in New Haven.
  - *Stage-1 Output:*
    (1)(,)(000)()(Ph)(.)(D)(.)(s)()(will)()(meet)()(in)()(New)()(Haven)(.)
  - *Stage-2 Output:*
    (one)()(thousand)()(P)(H)(Ds)()(will)()(meet)()(in)()(New Haven)(.)

1.2 Expansions

- **Task:** Expand numbers, abbreviations, and acronyms to canonical orthographic representations – typically performed as part of the 2*nd* stage of tokenization.
- **Motivation:** Allows uniform treatment of input data by later stages.
- **Methods:**
  - Full-form token subsequence rewrite grammar.
  - Finite-state transduction.
- **Problems:**

  - *Numbers*
    - Full-form treatment not possible for infinite set.
    - Differing conventions for pronunciation of cardinals, ordinals, times, and years.
  - *Abbreviations*
    - $1 : n$ mapping from abbreviations to canonical orthographic forms, e.g.
      - German "tgl." → {*täglich, täglicher, tägliches*, ...}
      - English "St." → {*street, saint*}
  - *Acronyms*
    - Full-form treatment not practical.
    - Some acronyms are spelled out ("CPU", "BA"), while others are spoken as single words ("RAM", "SCSI").

- **Workaround(s):**
  - Probabilistic determination of "best" expansion (error-prone).
  - Context-dependent expansion heuristics (pre-empts "real" contextual analysis).
  - Additional markup for problematic tokens to be treated at a later stage.

1.3 Sentence Boundary Detection

- **Task:** Identify and mark sentence boundaries in input text.
- **Motivation:** Sentence boundaries (and types) influence prosodic parameters.
- **Method:**
  - *Sentence-terminal punctuation:* Traditionally, the punctuation characters ".", "?", "!", and ":" have been used as sentence end markers.
  - *Sentence-initial capitalization:* For English, capitalization is often used as a cue for sentence boundary detection.
- **Problems:**
  - Token-internal punctuation (numbers, times, abbreviations, *etc.*)
  - Language-specific capitalization conventions
- **Workaround(s):**
  - Good tokenization and expansion modules can help reduce punctuation misinterpretation.
  - Probabilistic method described in Liberman and Church (1992).

1.4 Collocations

- **Task:** Identify and label collocations and idioms in the input token stream.
- **Motivation:** Prosody for collocations often does not conform to the "normal" rules their surface forms.
- **Method(s):**
  - *Condensation:* Treat like numbers and acronyms, "expansion" becomes "condensation".
  - *Procrastination:* Ignore in preprocessing stage, analyze later (using output from morphology / tagger / chunker / parser).
- **Problems:**
  - *Condensation:* For some languages (German, French), collocational surface forms depend on morphosyntactic context (inflection), which makes them hard to detect early on.
  - *Procrastination:* Correct identification of collocations can provide crucial information for later stages of contextual analysis.

2 Morphological Analysis

- **Task:** Segment input tokens into *morph*[1] sequences.
- **Motivation:**
  - Pronunciation dictionary minimization.
  - Identification of morph boundaries can improve letter-to-sound transduction accuracy:
    * English "bo**TH**er" *vs.* "ho**T**/**H**ouse",
    * German "Neben/**S**trasse" *vs.* "Demon**S**tra/tion"
  - Identification of inflectional morphology can restrict search space for later contextual analysis stages (PoS tagging), thereby improving efficiency.
  - Root/affix differentiation improves stress assignment accuracy.
- **Methods:**
  - *Full-form lexicon:* Impractical, inaccurate, and costly, but fast – useful for closed-class items.
  - *Procedural rules:* Must be painstakingly hand-crafted, impractical for heavily inflected languages.
  - *Declarative stem/affix association lexicon:* Minimal storage, but difficult to construct and inefficient to access at runtime.

[1] A morph is either a root or an affix

  - *Two-level morphology:* Expressable as a finite-state transducer (FST), quite efficient, and can even handle compounding.

- **Problems & Workarounds:**
  - *Efficiency:* Many morphological analysis strategies require a good deal of processing power in their raw forms; workarounds usually involve precompiled indices used to speed lookup operations.
  - *Indexing:* Indexing of roots and affixes for efficient access can result in large memory requirements (also applies FST methods); workarounds generally result in greater time complexity.
  - *Robustness:* Misspellings and previously unknown tokens are not handled by any of the above methods in their strict forms; a workaround known as an *open lexicon strategy* allows unknown stems in analyses and reduces memory requirements, but can also harm accuracy.

References

J. Allen, S. Hunnicut, and D. Klatt. *From Text to Speech: the MITalk system.* Cambridge University Press, 1987.

T. Dutoit. *An Introduction to Text-to-Speech Synthesis.* Kluwer, Dordrecht, 1997.

M. J. Liberman and K. W. Church. Text analysis and word pronunciation in text-to-speech synthesis. In S. Furui and M. M. Sondhi, editors, *Advances in Speech Signal Processing.* Dekker, New York, 1992.