Introduction to Text Analytics with R – Part 2

Dave Langer

Representation

- Key question: How to represent text as a data frame?
- Answer: Words become columns!
- Take the following hypothetical document:
  - “If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck.”

Tokenization

- First step in representation is decomposing a text document into distinct pieces, or *tokens*.

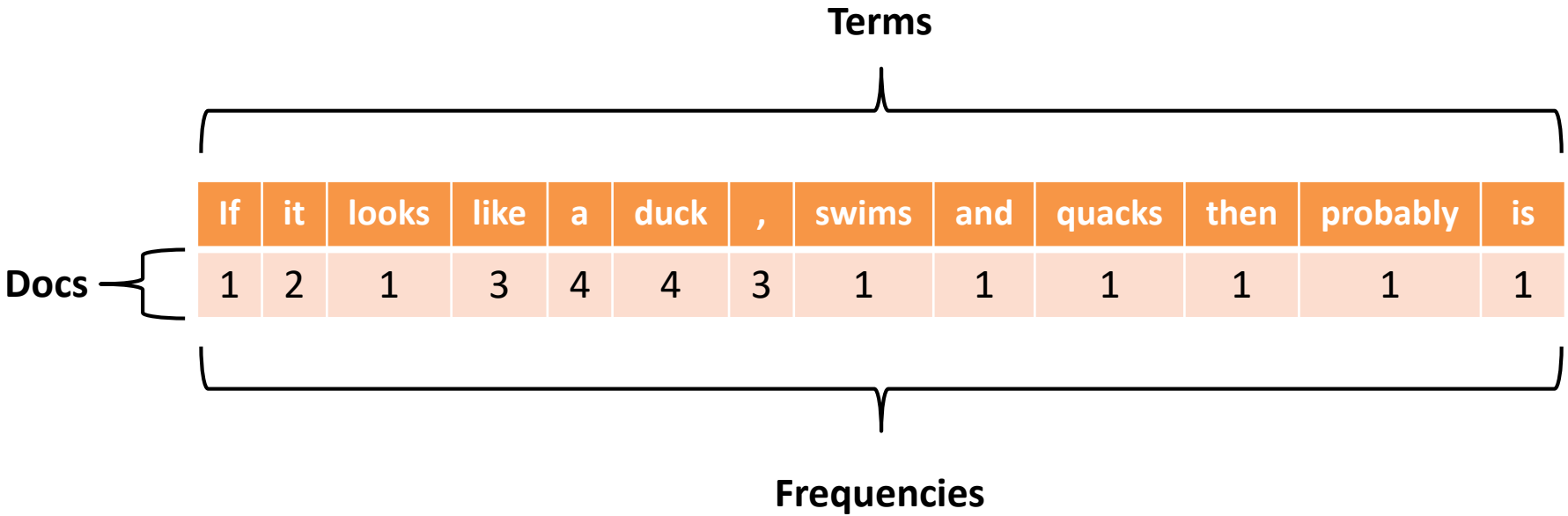
- Applying tokenization to our hypothetical document could produce the following tokens:
  - [If] [it] [looks] [like] [a] [duck] [,] [swims] [like] [a] [duck] [,] [and] [quacks] [like] [a] [duck] [,] [then] [it] [probably] [is] [a] [duck][.]

Document-Frequency Matrix

- With tokenization complete, it is possible to construct a data frame (i.e., a matrix) where:
  - Each row represents a document.
  - Each column represents a distinct token.
  - Each cell is a count of the token for a document.

- This representation is wildly useful!

Hypothetical DFM



- Word ordering is not preserved!
- This is known as the “bag-of-words” model.
- The BOW model is a very common representation.

Some Considerations

- Do we want all tokens to be terms in our DFM?
  - Casing (e.g., If vs if)?
  - Punctuation (e.g., “, ?, !, etc.)?
  - Numbers (e.g., 0, 56, 109, etc.)?
  - Every word (e.g., the, an, a, etc.)?
  - Symbols (e.g., <, @, #, etc.)?
  - What about similar words (e.g., ran, run, runs, running)?

- Pre-processing is a major part of text analytics!