University of Sheffield

School of Computer Science

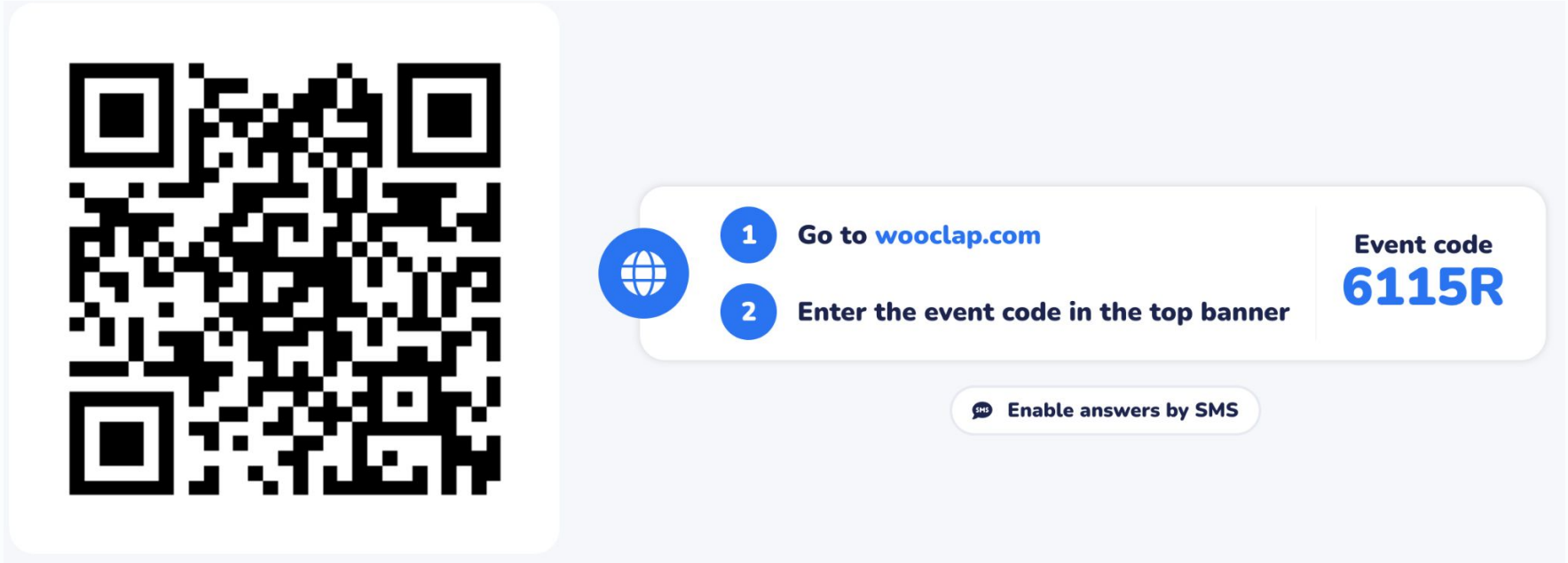
COM6115: Text Processing

Weeks 1-6 Revision

Varvara Papazoglou

Week 1
Intro to Text Processing

Wooclap – Text preprocessing tasks (Weeks 1 & 5)



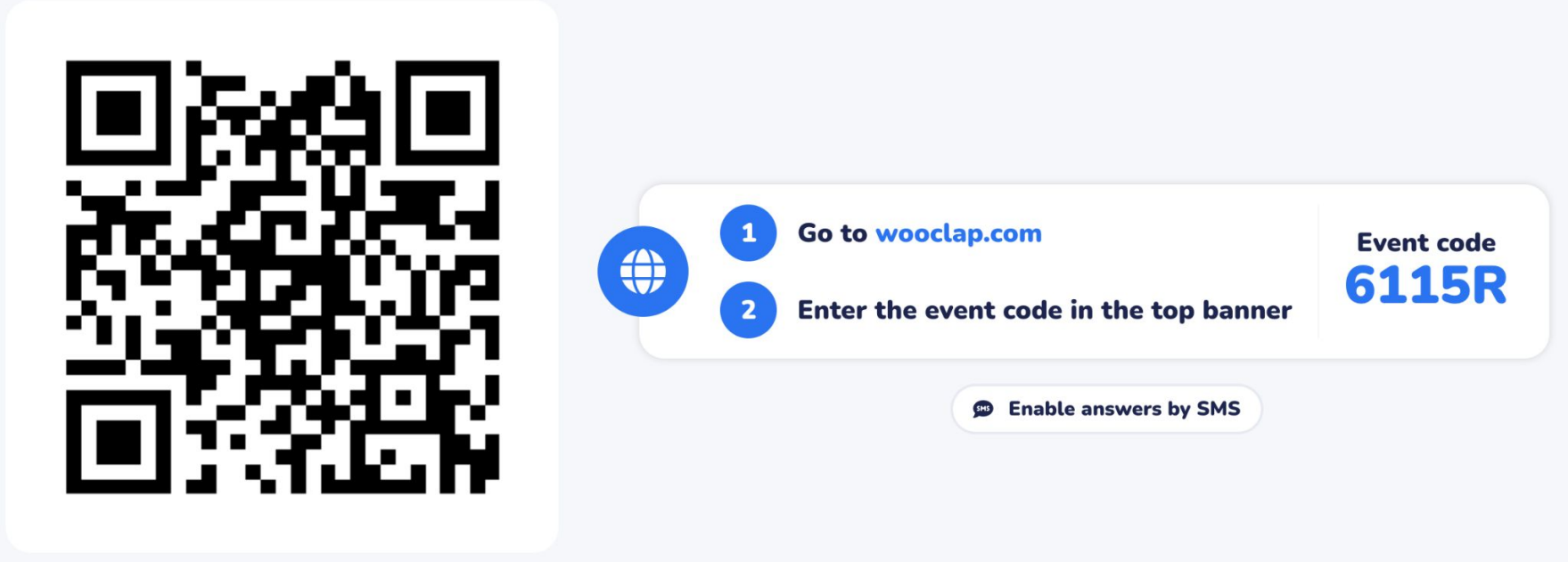
Weeks 2-3
Text Encoding & Text Compression

Character Encoding

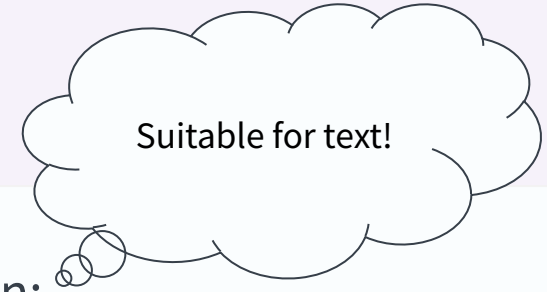
- **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (ASCII)
  - 7-bit code
  - Introduced in 1963, finalised in 1968
  - Not sufficient for several Western European languages
    - Additional accents and symbols

- **U**nicode **T**ransformation **F**ormat (UTF)
  - **UTF-32**: 32-bit (4-byte) code unit length
  - **UTF-16**: 16-bit (2-byte) code unit length
  - **UTF-8**: 1-4 bytes (variable code unit length)

Wooclap – Encodings

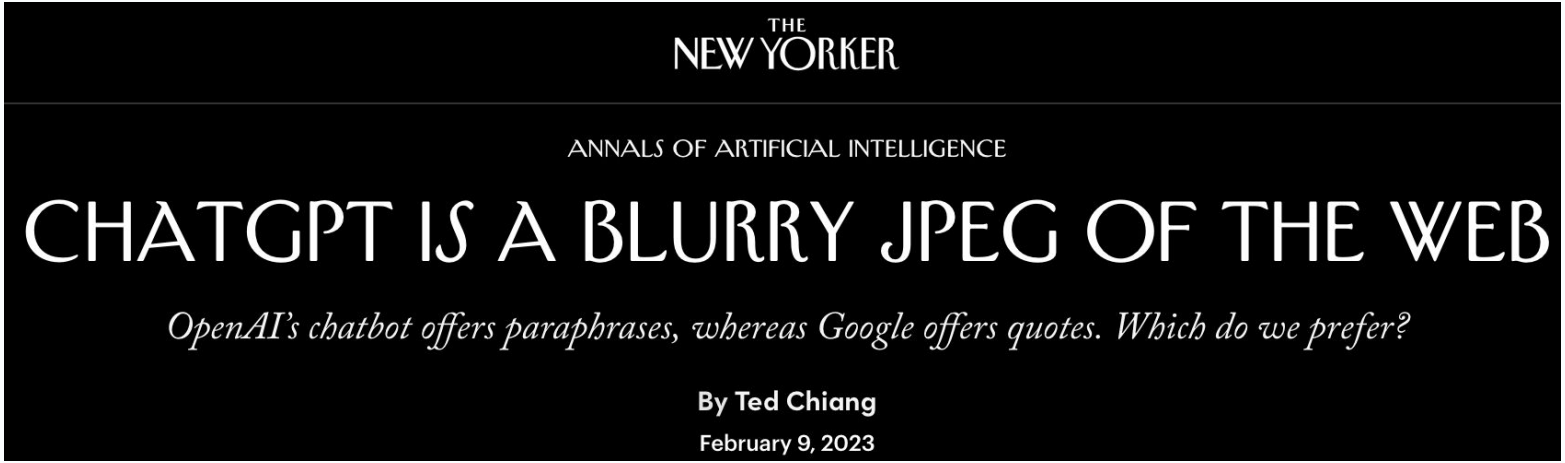


Compression

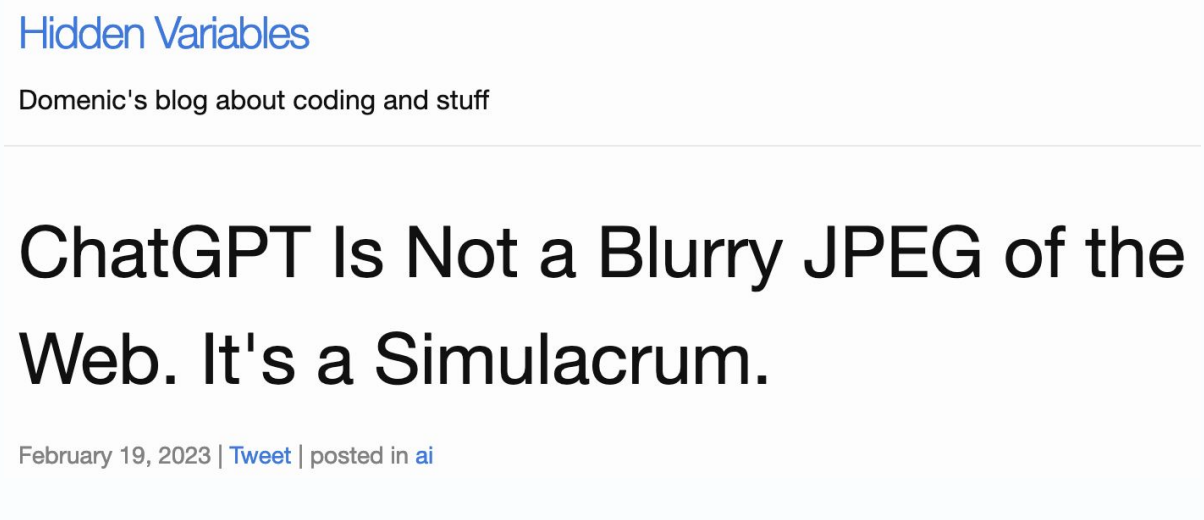


- **Lossless** Compression:
  - class of algorithms allowing original data to be **perfectly reconstructed** from compressed data

- **Lossy** Compression:
  - achieve data reduction by **discarding** (i.e. losing) information
  - suitable for **certain media types**, esp.: image / video / audio data
  - widely used in data **streaming** contexts



https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web



https://blog.domenic.me/chatgpt-simulacrum/

Additional Resources

- Language Modelling is Compression (Delétang et al., 2024)
- LLMZip: Lossless Text Compression using Large Language Models (Valmeekam et al., 2023)
- Semantic and Generative Models for Lossy Text Compression (Witten et al., 1994)

Text compression techniques

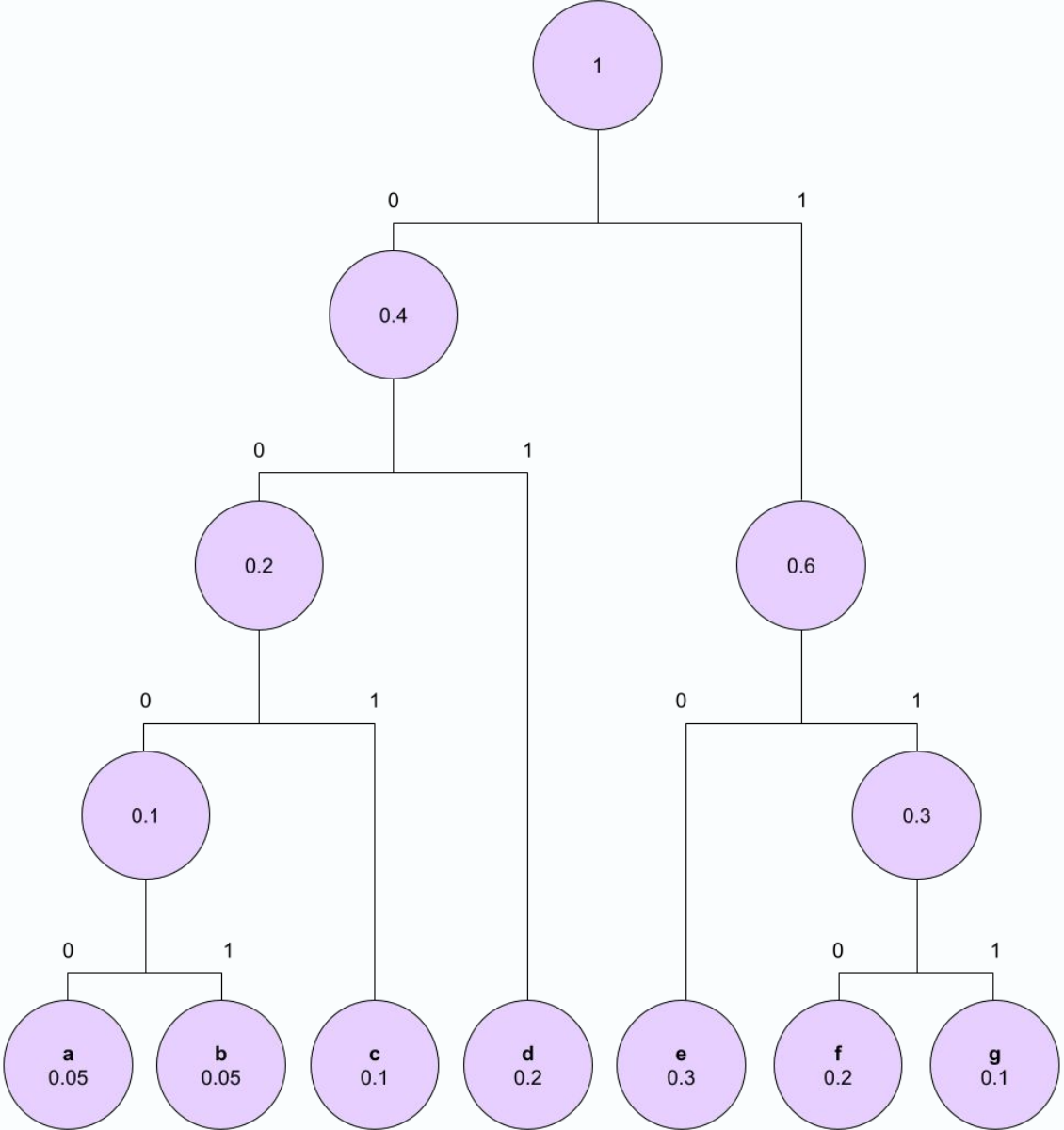
- **symbolwise** vs. **dictionary** methods
  - **Symbolwise**: estimate the probabilities of symbols (characters/words); code one symbol at a time using shorter codewords for the more likely symbols
  - **Dictionary**: replace substrings in a text with codewords/indices that identify the substring in a dictionary

- **static** vs. **adaptive** methods
  - **Static**: pre-defined model/dictionary for all texts
  - **Semi-static**: derives model/dictionary for the file in a 1^{st} pass; applies it in a 2^{nd} pass
  - **Adaptive**: builds model/dictionary adaptively in 1 pass

Huffman coding

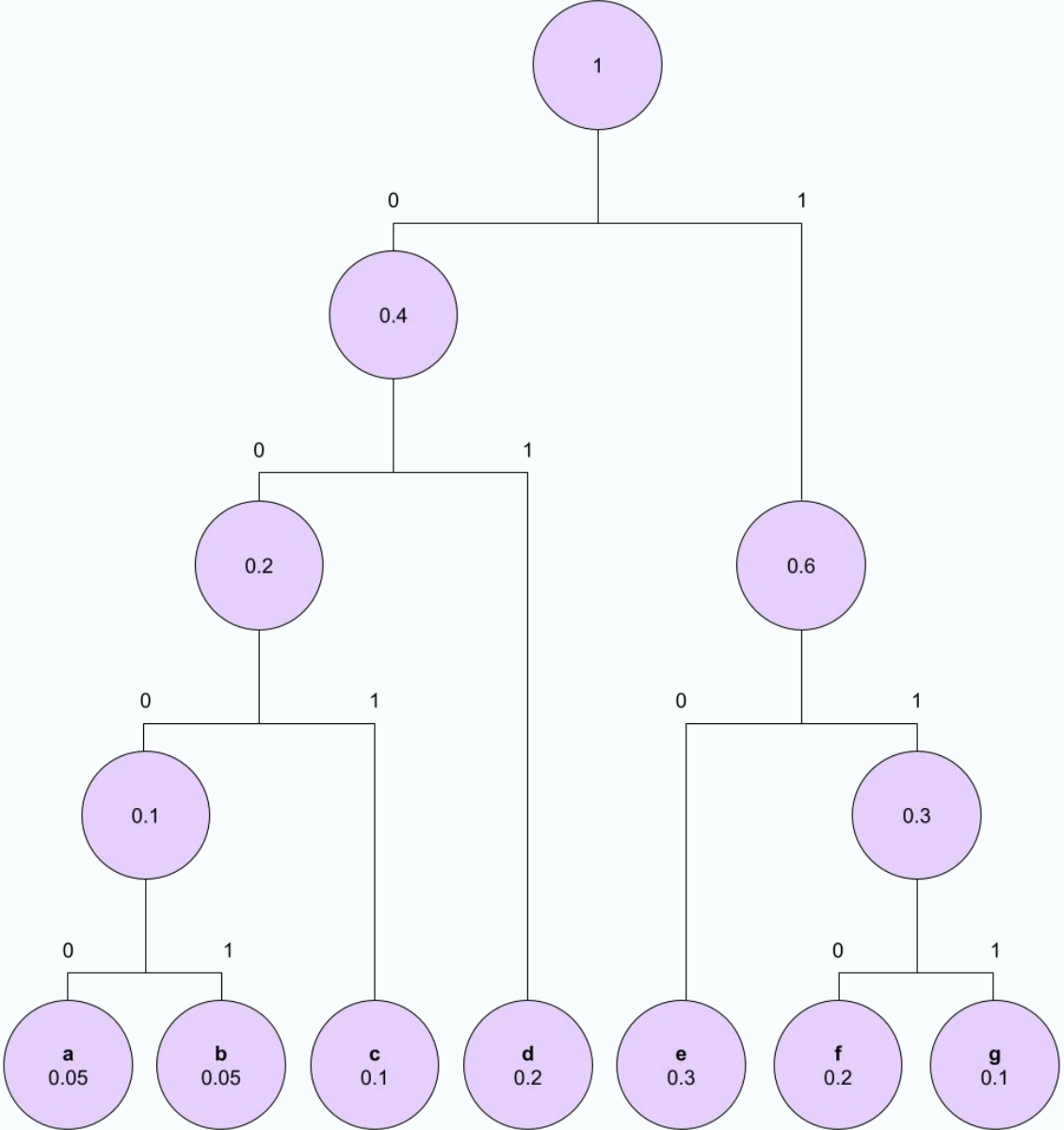
- Encoding:
  - Given the estimated probability of symbols, build the coding tree.
    - Start from the leave nodes.

- Decoding:
  - Given the coding tree (which is prefix-free), decode the sequence of integers into the corresponding symbols.
    - Start from the root to determine which codeword corresponds to each symbol.



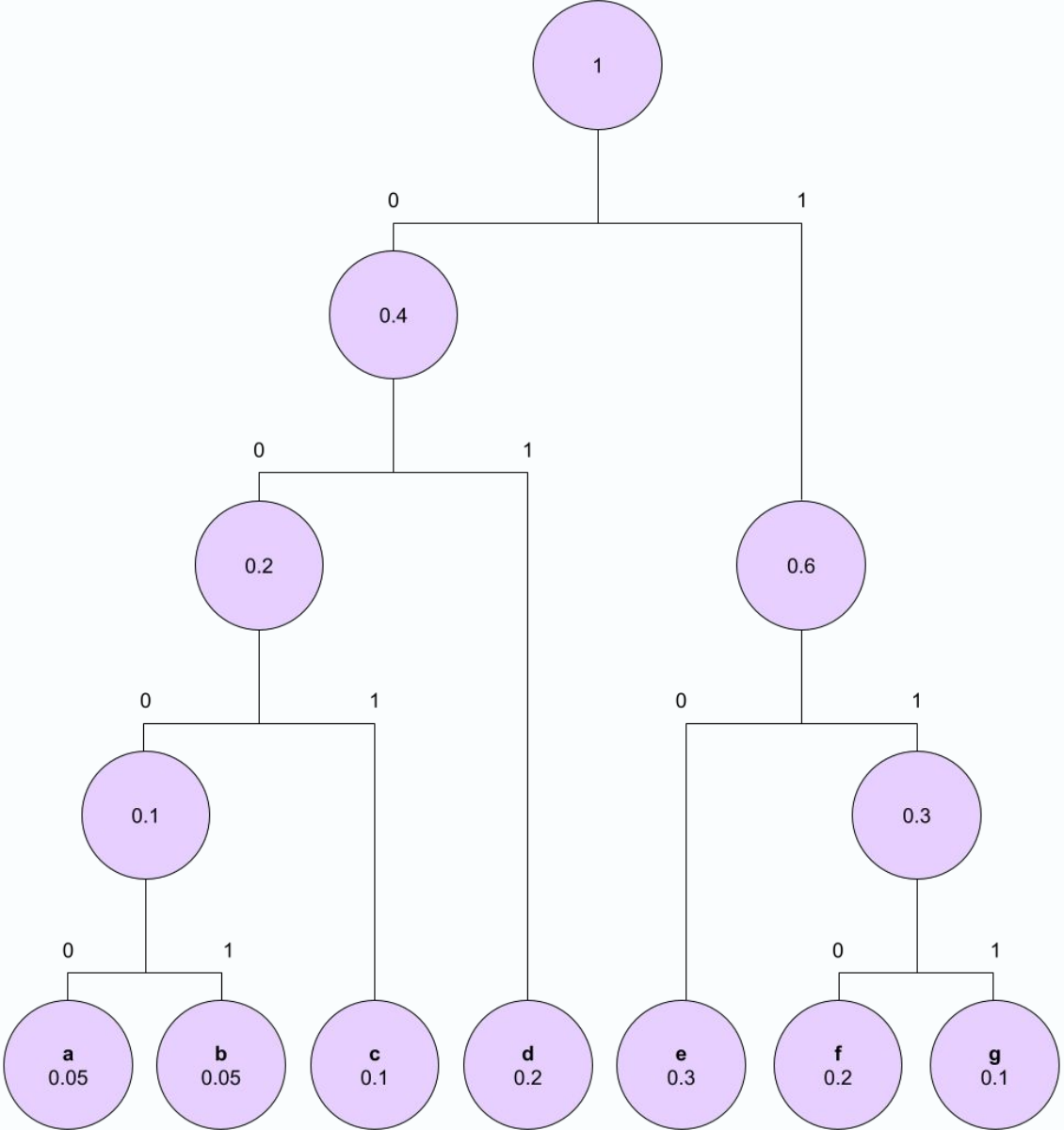
Test your understanding:

1. Encode
   a. “ddef”
   b. “cab”



Test your understanding:

1. Encode
   a. “ddef”
      010110110
   b. “cab”
      00100000001

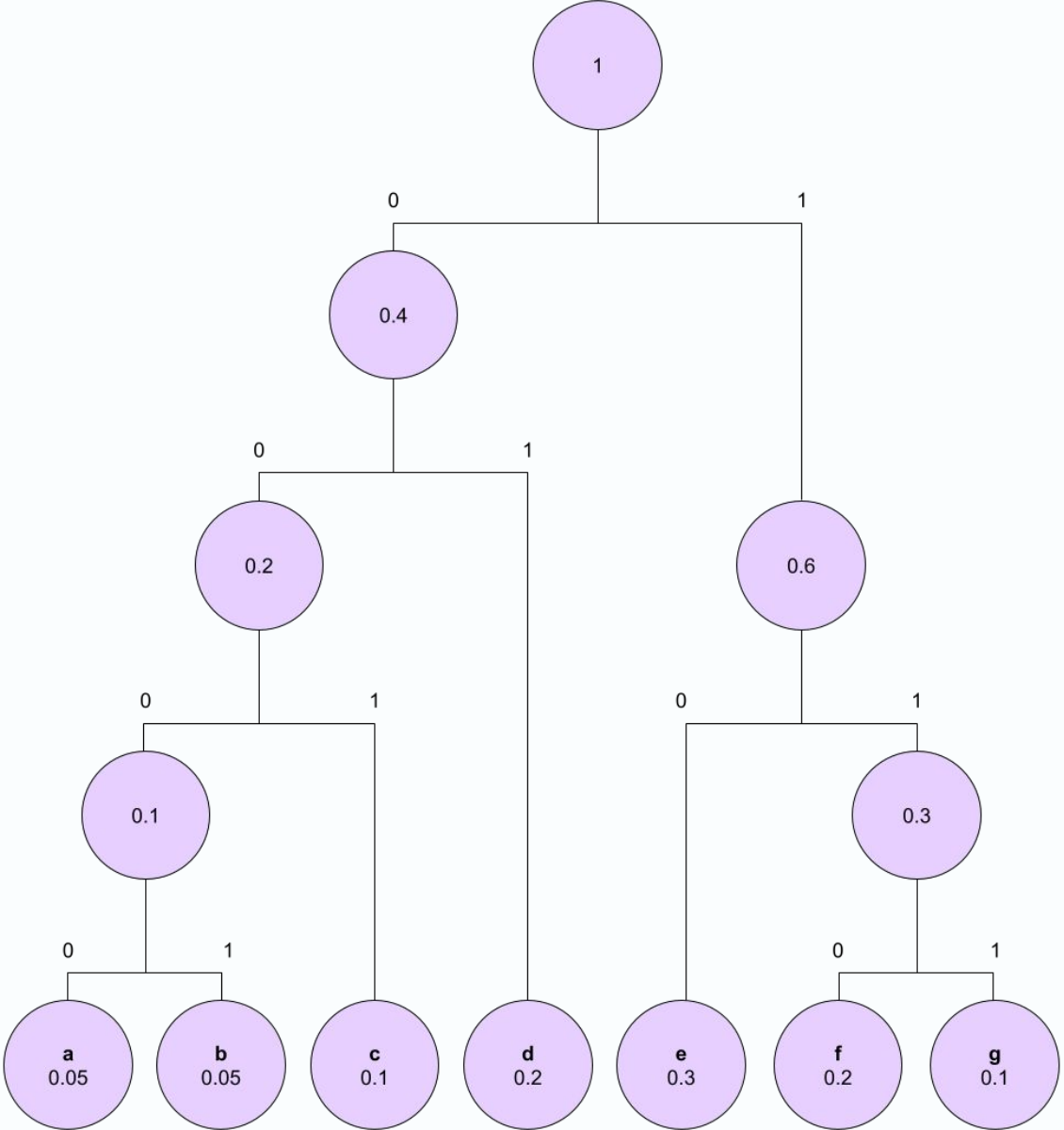


Test your understanding:

2. Decode

   a. 0001000001

   b. 1111101001



Test your understanding:

2. Decode

   a. 0001000001
      “bad”

   b. 1111101001
      “gfed”

Canonical Huffman Codes

- Encoding:
  - Given the length of the codewords, group symbols by their code-length
  - Assign codes

  Example:

[2]	d	e	[3]	c	f	g	[4]	a	b
	00	01		100	101	110		1110	1111

Canonical Huffman Codes

- Decoding:
  - Given the sequence of symbols & number of symbols at each code-length, form the symbol groups

  Example:

  d e c f g a b, (0, 2, 3, 2)

Canonical Huffman Codes

- Decoding:
  - Given the sequence of symbols & number of symbols at each code-length, form the symbol groups

  Example:

  d e c f g a b, (0, 2, 3, 2)

  [1] - [2] d e [3] c f g [4] a b

Canonical Huffman Codes

- Decoding:
  - Given the sequence of symbols & number of symbols at each code-length, form the symbol groups

  Example:

  d e c f g a b, (0, 2, 3, 2)

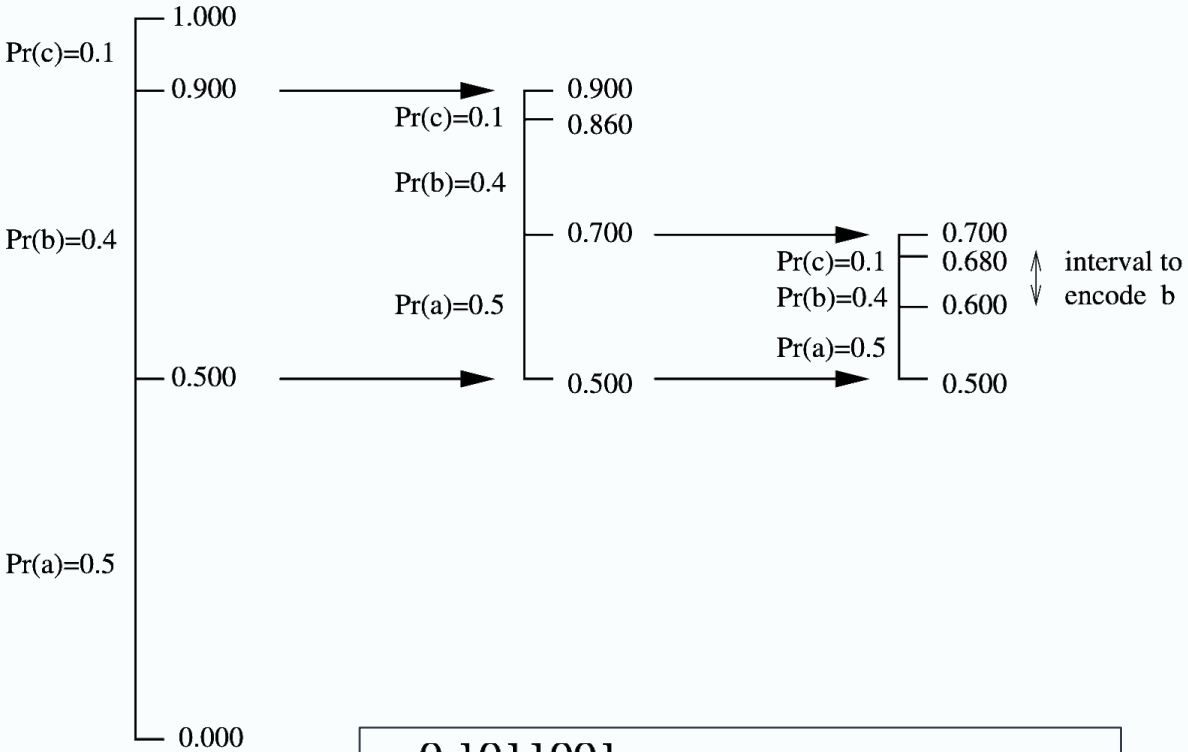
  [1] - [2] d e [3] c f g [4] a b

  [2] d e [3] c f g [4] a b
  00 01 100 101 110 1110 1111

Arithmetic Coding

- Given: the static model of probabilities and the string to be encoded
- Divide the intervals between 0-1
- Represent the codewords as a stream of bits, which is actually expressed as a fractional binary number between 0 and 1

Arithmetic Coding



0.1011001

$\frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{128} = \sim 0.695$

1/2 1/4 1/8 1/16 1/32 1/64 1/128

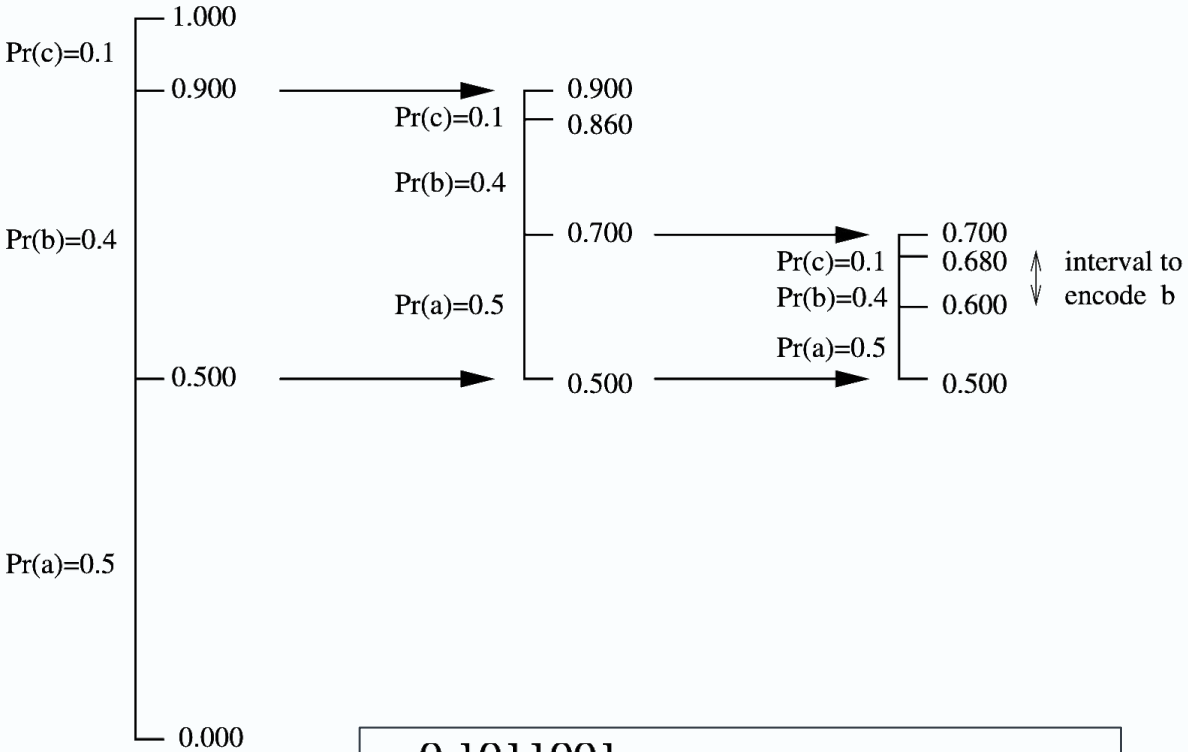
Test your understanding:

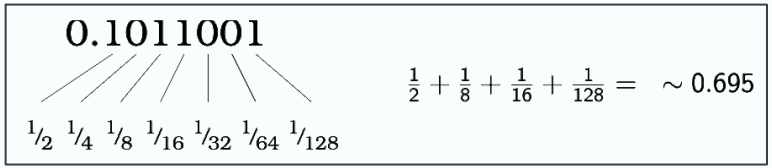
This is how you can encode "bab" → 101 (0.625), given
Pr(a) = 0.5 Pr(b) = 0.4 Pr(c) = 0.1

1. Why is "bb" → 11 ?

2. Why is 1011 → "bac" ?

Arithmetic Coding





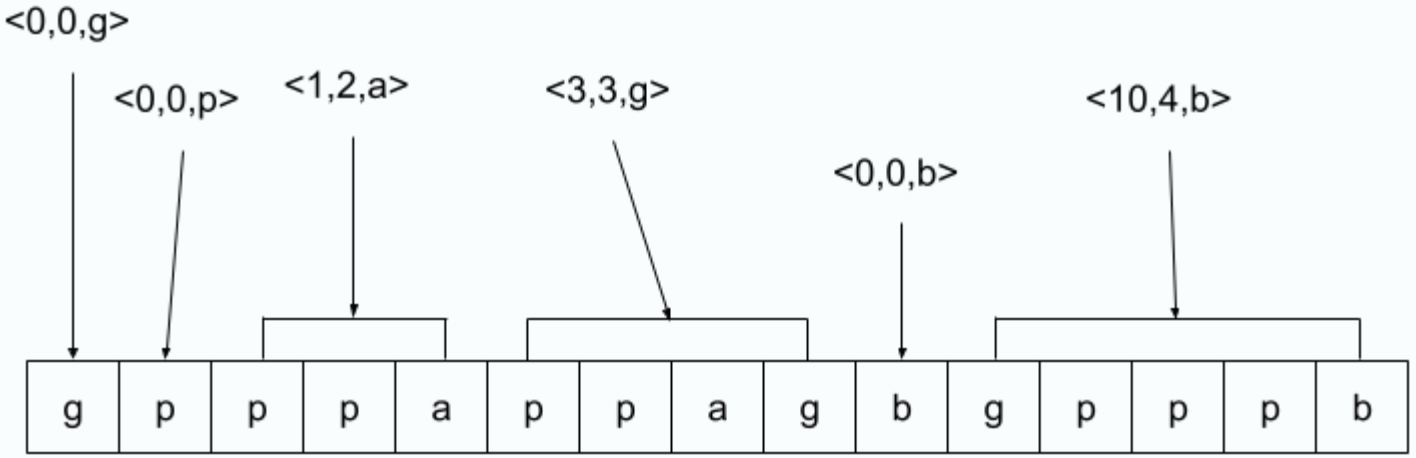
Test your understanding:

This is how you can encode
"bab" → 101 (0.625), given
Pr(a) = 0.5 Pr(b) = 0.4 Pr(c) = 0.1

1. Why is "bb" → 11 ?
   11 → 0.11 → ½ + ¼ = 0.75

2. Why is 1011 → "bac" ?
   ½ + ⅛ + 1/16 = 0.688

LZ77



Weeks 4-5
Information Retrieval

Retrieval Models

- the **Boolean Model**
  - each document in a collection is either relevant to a given query or not relevant to that query
  - queries are formulated using boolean operators (AND, OR, NOT) between terms

- the **Vector Space Model**
  - each document in a collection has some degree of relevance to query
  - documents and queries are both represented as vectors of the terms occurring within them
  - degree of similarity computed as cosine of angle between document and query vectors

- the **Probabilistic Model**
  - ranked retrieval model like the vector space model
  - relatedness of document to query measured as the probability that a document is relevant to a given query

Retrieval Models

- the **Boolean Model**
  - each document in a collection is either relevant to a given query or not relevant to that query
  - queries are formulated using boolean operators (AND, OR, NOT) between terms

- the **Vector Space Model**
  - each document in a collection has some degree of relevance to query
  - documents and queries are both represented as vectors of the terms occurring within them
  - degree of similarity computed as cosine of angle between document and query vectors

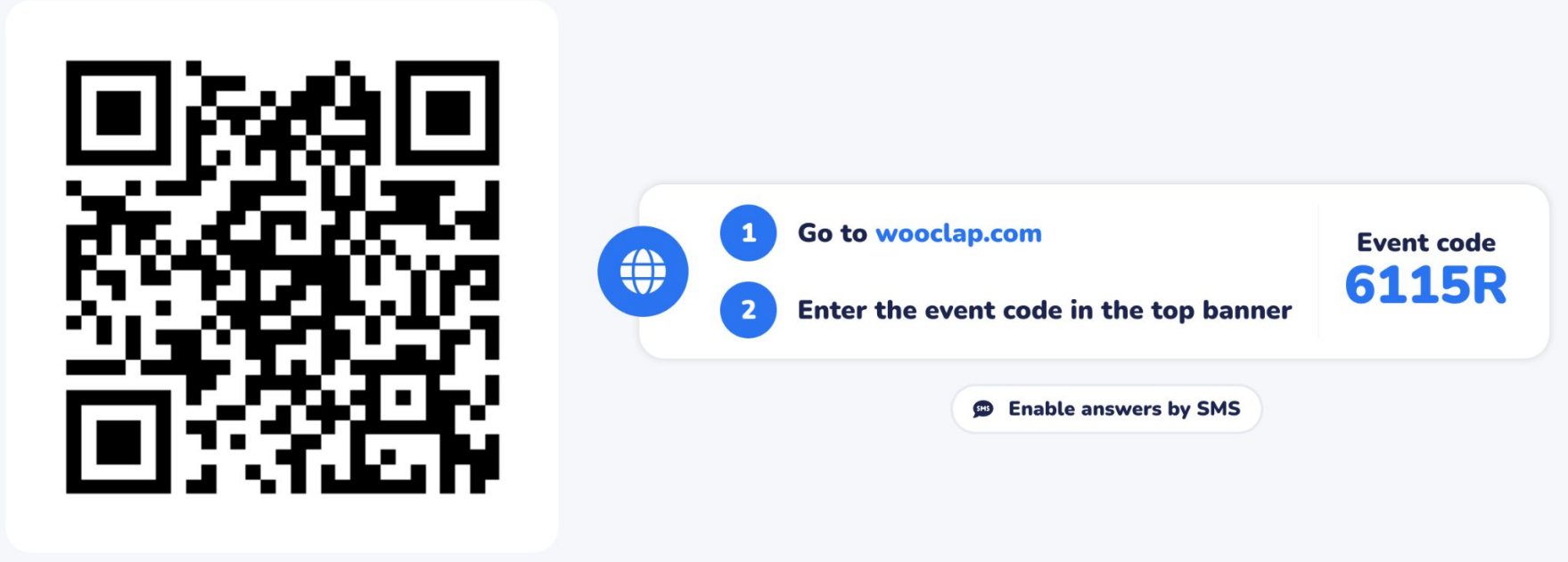
- the **Probabilistic Model**
  - ranked retrieval model like the vector space model
  - relatedness of document to query measured as the probability that a document is relevant to a given query

Term weighting

- Binary weighting
- tf
- tf.idf

→ Why is each of the above increasingly ‘better’ for IR?

Wooclap – IR


1
Go to wooclap.com
2
Enter the event code in the top banner
Event code
6115R
Enable answers by SMS