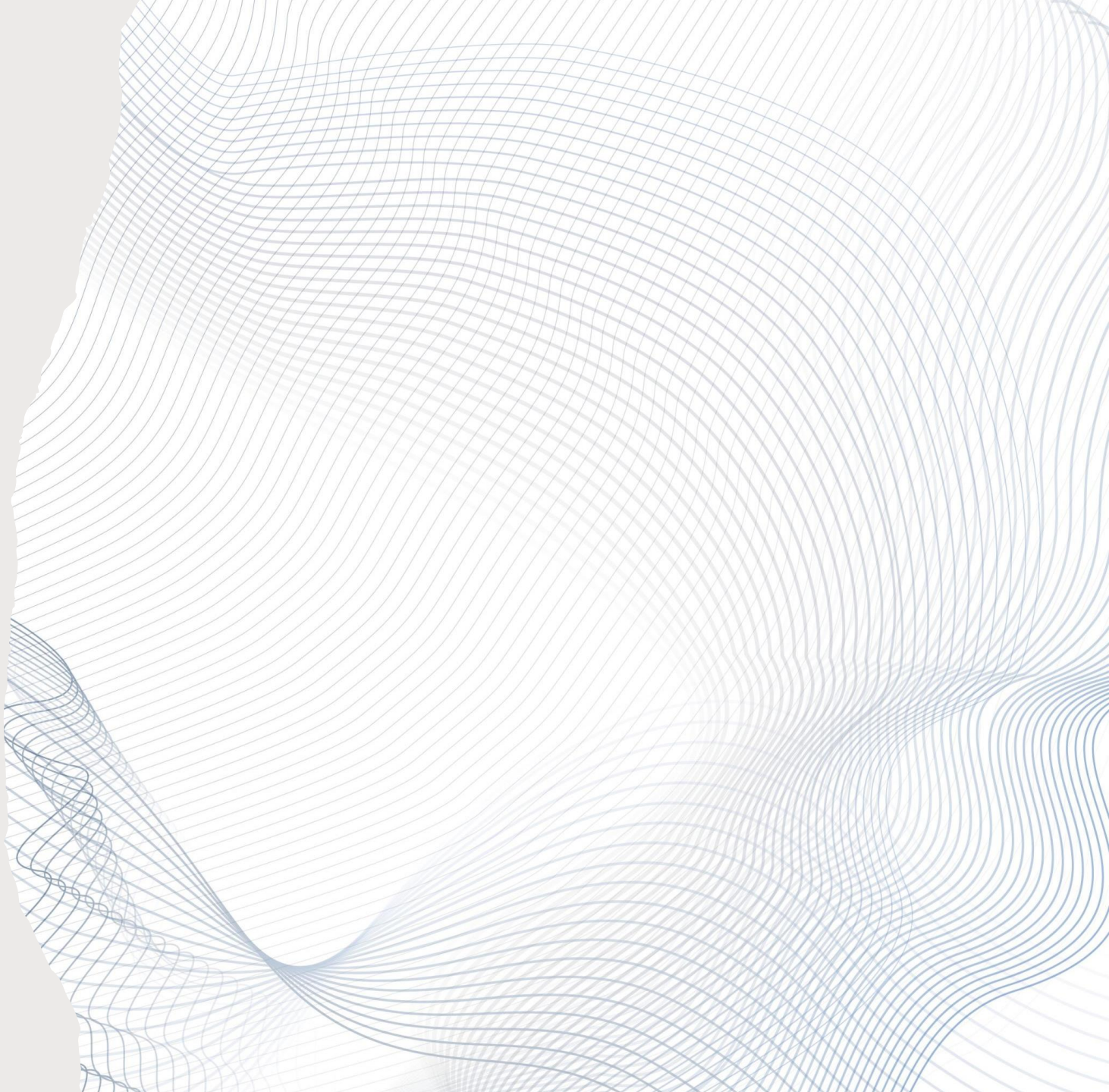
# BYTE PAIR ENCODING

# TERMINOLOGIES

**Word Frequency:** The number of times a word coming in the text corpus

**Char Frequency:** The number of time a char appears in the text corpus



# STEPS INVOLVED

For a given text corpus ‘C’, perform the following step to implement BPE

- **STEP 1:** Add end of the word char ‘<\w’ at the end of each word. Add spaces between the characters of each word.
- **STEP 2 :** Find word frequencies, character frequencies (in the form of dictionary)
- **STEP 3:** For a predefined no. of iterations X, perform the following:
  - **STEP 3a:** Find the pair of most frequent consecutive characters $P_{freq}$ .
  - **STEP 3b:** Merge the pair of characters in $P_{freq}$ , and save it as rule no. ‘i’.
  - **STEP 3c:** Update word frequency, and character frequency dictionaries.

# E X A M P L E

- Corpus = ‘I am Adam’, Let X = 5
- **Step 1:**

| | | |
|---|---|---|
| I <\w | a m <\w | A d a m <\w |

- **Step 2: Word frequency →**

| | | |
|---|---|---|
| I <\w: 1 | a m <\w: 1 | A d a m <\w: 1 |

**Character Frequency →**

| | | | | | |
|---|---|---|---|---|---|
| I :1 | <\w: 1 | a: 2 | m: 2 | A: 1 | d: 1 |

- **Step 3a:** All possible consecutive characters possible( using unique characters) with their counts are {(I, <\w): 1, (a, m): 2, (m, <\w): 2, (A, d): 1, (d, a): 1} **(Note: ignore pairs with count=0)**
- **Step 3b:** Most frequent pair $\mathbf{P_{freq}}$ = (a, m) ; **Rule 1: a+m → ‘am’ (sub word token)**

| | | |
|---|---|---|
| I <\w: 1 | **am** <\w: 1 | A d **am** <\w: 1 |

- **Step 3c: update word frequency →**

**update character frequency →**

| | | | | |
|---|---|---|---|---|
| I :1 | <\w: 1 | **am**: 2 | A: 1 | d: 1 |

**For iteration 2:**

**Step 3a:** All possible consecutive characters possible( using updated characters) with their counts are {(I, <\w): 1, (am, <\w): 2, (A, d): 1, (d, am): 1}

**Step 3b:** Most frequent pair $\mathbf{P_{freq}}$ = (am, </w) ; **Rule 2: am + </w → 'am</w'**

**Step 3c: update word frequency →**

| | | |
|---|---|---|
| I <\w: 1 | **am<\w**: 1 | A d **am<\w**: 1 |

**update characters and frequency →**

| | | | | |
|---|---|---|---|---|
| I :1 | <\w: 1 | **am</w**: 2 | A: 1 | d: 1 |

**For iteration 3:**

**Step 3a:** All possible consecutive characters possible( using updated characters) with their counts are {(I, <\w): 1, (A, d): 1, (d, am</w): 1}

**Step 3b:** Most frequent pair $\mathbf{P_{freq}}$ = (I, </w) ; **Rule 3: I + </w → 'I</w'**

**Step 3c: update word frequency →**

| | | |
|---|---|---|
| **I<\w**: 1 | am<\w: 1 | A d am<\w: 1 |

**update characters and frequency→**

| | | | |
|---|---|---|---|
| **I</w**:1 | am</w: 2 | A: 1 | d: 1 |

**For iteration 4:**

**Step 3a:** All possible consecutive characters possible( using updated characters) with their counts are {(A, d): 1, (d, am</w): 1}

**Step 3b:** $P_{freq}$ = (A, d) ; **Rule 4: A + d→ 'Ad'**

**Step 3c: update word frequency →**

| | | |
|---|---|---|
| l<\w: 1 | am<\w: 1 | **Ad** am<\w: 1 |

**update characters and frequency→**

| | | |
|---|---|---|
| l<\w: 1 | am<\w: 1 | **Ad**: 1 |

**For iteration 5:**

**Step 3a:** All possible consecutive characters possible( using updated characters) with their counts are {(Ad, am): 1}

**Step 3b:** $P_{freq}$ = (Ad, am) ; **Rule 5: Ad+ am→ 'Adam'**

**Step 3c: update word frequency →**

| | | |
|---|---|---|
| l<\w: 1 | am<\w: 1 | **Adam**<\w: 1 |

**update characters and frequency→**

| | | |
|---|---|---|
| l<\w: 1 | am<\w: 1 | **Adam**: 1 |

**Generate sub word for test sentence**

**Test = Adam Madam**

| | |
|---|---|
| **A, d, a, m, </w** | **M, a, d, a, m, </w** |

Using rules order wise:

**Using Rule 1:** {A, d, am, </w, M, a, d, am, </w}
**Using Rule 2:** {A, d, am</w, M, a, d, am</w}
**Using Rule 3:** Not applicable
**Using Rule 4:** {Ad, am</w, M, a, d, am</w}
**Using Rule 5:** {Adam</w, M, a d, am</w}. ***Final list of sub word tokens***