```
from typing import List, Set

key_to_dot = {
    'D': 1, 'W': 2, 'Q': 3, 'K': 4, 'O': 5, 'P': 6
}

braille_to_char = {
    "1": 'a', "1-2": 'b', "1-4": 'c', "1-4-5": 'd', "1-5": 'e',
    "1-2-4": 'f', "1-2-4-5": 'g', "1-2-5": 'h', "2-4": 'i', "2-4-5": 'j',
    "1-3": 'k', "1-2-3": 'l', "1-3-4": 'm', "1-3-4-5": 'n', "1-3-5": 'o',
    "1-2-3-4": 'p', "1-2-3-4-5": 'q', "1-2-3-5": 'r', "2-3-4": 's', "2-3-4-5": 't',
    "1-3-6": 'u', "1-2-3-6": 'v', "2-4-5-6": 'w', "1-3-4-6": 'x',
    "1-3-4-5-6": 'y', "1-3-5-6": 'z'
}

dictionary = ["cat", "bat", "rat", "can", "man", "cap", "map", "mat", "cot", "cop"]

def convert_braille_to_text(braille_input: List[Set[str]]) -> str:
    result = ""
    for cell in braille_input:
        dots = sorted([key_to_dot[ch] for ch in cell if ch in key_to_dot])
        key = '-'.join(str(dot) for dot in dots)
        result += braille_to_char.get(key, '?')
    return result

def levenshtein_distance(a: str, b: str) -> int:
    dp = [[0] * (len(b) + 1) for _ in range(len(a) + 1)]
    for i in range(len(a) + 1):
        for j in range(len(b) + 1):
            if i == 0:
                dp[i][j] = j
            elif j == 0:
                dp[i][j] = i
            elif a[i - 1] == b[j - 1]:
                dp[i][j] = dp[i - 1][j - 1]
            else:
                dp[i][j] = 1 + min(dp[i - 1][j - 1],
                                   dp[i - 1][j], dp[i][j - 1])
    return dp[-1][-1]

def suggest_word(word: str) -> str:
    closest_word = min(dictionary, key=lambda x: levenshtein_distance(word, x))
    return closest_word

def main():
    braille_input = []
    print("Enter Braille characters one by one.")
    print("For each character, type the keys (D W Q K O P) space-separated.")
    print("Press Enter on an empty line when done.")
    while True:
        line = input("Enter keys for a Braille character: ").strip().upper()
        if not line:
            break
```

```
        keys = line.split()
        cell = set()
        valid = True
        for key in keys:
            if key not in key_to_dot:
                print(f"Invalid key: {key}")
                valid = False
                break
            cell.add(key)
        if valid:
            braille_input.append(cell)

    if not braille_input:
        print("No input provided.")
        return

    typed_word = convert_braille_to_text(braille_input)
    print(f"Typed word: {typed_word}")
    print(f"Suggested word: {suggest_word(typed_word)}")

if __name__ == "__main__":
    main()
```