7A-7C. A subset of the candidate words are selected based on predefined criteria (**210**) and the selected subset is presented to the user (**212**). In some embodiments, the selected candidate words are presented to the user as a horizontal listing of words.

**[0049]** Attention is now directed to FIG. **3**, a flow diagram illustrating a process of scoring candidate words in accordance with some embodiments. Process flow **300** describes a process of scoring a candidate word. The scoring helps determine which word(s) in the dictionary is/are the best potential replacement(s) for the input sequence of characters.

**[0050]** Each character in a candidate word is compared to the character in the corresponding position in the input sequence (**302**). Thus, the first character in the candidate word is compared to the first character in the input sequence, the second character in the candidate word is compared to the second character in the input sequence, and so forth. If either the candidate word or the input sequence is longer than the other, then the additional characters beyond the shorter length of the two are ignored in the comparison. In some embodiments, further comparison of the candidate word with the input sequence may be made. For example, the further comparison may include determining the number of character differences between the candidate words and the input sequence, and determining if any character differences are a result of transposed characters. A score is calculated for the candidate word based on the comparison described above (**304**). Each character comparison yields a value, and the values are added to yield the score for the candidate word.

**[0051]** In some embodiments, the score value given for a character comparison is based on the actual characters as opposed to merely whether the characters match. More particularly, the value may be based on whether the character in the candidate word matches the corresponding character in the input sequence exactly and/or whether the character in the candidate word is a keyboard layout neighbor of the corresponding character in the input sequence.

**[0052]** Optionally, a first "bonus" may be added to the score of the candidate word if the candidate word and the input sequence are different in only one character (**306**). Similarly, an optional second "bonus" may be added to the score of the candidate word if the candidate word and the input sequence are different in only a pair of transposed adjacent characters (**308**). Further details regarding candidate word scoring is described below, in relation to FIGS. **7A-7C**.

**[0053]** Attention is now directed to FIG. **4**, a flow diagram illustrating a process of selecting and presenting candidate words in accordance with some embodiments. Process flow **400** describes in further details blocks **210** and **212** (FIG. **2**), which involves selection and presentation of candidate words.

**[0054]** The candidate words are split into two groups based on their usage frequency rankings within the dictionary (**402**). A first group includes the candidate words whose usage frequency rankings exceeds a predefined threshold. The second group includes the candidate words whose usage frequency rankings does not exceed the threshold. With each of the two groups, the candidate words are sorted by their candidate word scores.

**[0055]** There may be candidate words in the second group whose scores are very high because, for example, they match the input sequence exactly or almost exactly. In some embodiments, these high-scoring words may be removed from the second group and added to the first group if their scores exceed the score of the highest scoring candidate word in the first group by a predefined margin (**404**). In some embodiments, the predefined margin is that the score of the candidate word in the second group must be at least two times the highest candidate word score in the first group.

**[0056]** One or more of the highest scoring candidate words in the first group are presented to the user (**406**). It should be appreciated that if candidate words from the second group were moved to the first group as described above, then the candidate words that are presented will include at least one candidate word that was originally in the second group since that candidate word has a higher score than any of the original candidate words in the first group.

**[0057]** In some embodiments, if block **404** is not performed, either because no candidate word in the second group satisfies the score margin threshold or because the moving of candidate words is not performed at all, the highest scoring candidate word in the second group may nevertheless be presented along with the candidate words from the first group (**408**). Furthermore, in some embodiments, the input sequence as entered by the user may be presented as a matter of course (**410**). The user may choose any one of the presented candidate words to replace the input sequence, including choosing the input sequence as entered if the user is satisfied with it.

**[0058]** Attention is now directed to FIGS. **5**A and **5**B, which are exemplary layouts of letter keys on a keyboard in accordance with some embodiments. As described above, the prefix strings, based on which candidate words are identified, are generated based on characters in the input sequence and their corresponding neighbor characters on a keyboard layout. Keyboard layouts **502** and **504** are exemplary keyboard layouts. A keyboard layout defines the positions of each key on the keyboard and the alignment of the keys relative to each other. For ease of description, only the letter keys of the layouts **502** and **504** are shown. It should be appreciated, however, that a keyboard layout may also include keys for numbers, punctuation, symbols, and functional keys. In some embodiments, some keys may be overloaded, that is, a key may correspond to multiple characters and/or functions.

**[0059]** Layouts **502** and **504** are layouts that follow the well-known QWERTY layout. However, the key alignment in layout **502** is different from the key alignment in layout **504**. In layout **502**, the keys are aligned in rows but not in columns; a key in one row may straddle two keys in an adjacent row. For example, key "T" straddles keys "F" and "G" in layout **502**. In layout **504**, the keys are aligned in columns as well as in rows. The definition of which keys are the neighbors of a key may be different depending on how the keys are aligned. In layout **502**, the neighbors of a particular key may be defined as the keys that are directly adjacent to the particular key or whose peripheries "touch" a periphery of the particular key. For example, the neighbors of key "G" in layout **502** are keys "T," "Y," "F," "H," "V," and "B;" and the neighbors of key "W" are keys "Q," "E," "A," and "S," In layout **504**, the neighbors of a particular key may be defined as the keys that are immediately above, below, to the side of, and diagonal of the particular key. For example, the neighbors of key "G" in layout **504** are keys "R," "T," "Y," "F," "H," "C," "V," and "B;" and the neighbors of key "W" are keys "Q," "E," "A," "S," and "D."

**[0060]** It should be appreciated, however, that layouts **502** and **504** are merely exemplary, and that other layouts and key alignments are possible and the same key may have different neighbors in different layouts.