[0061] Attention is now directed to FIG. **6**, an exemplary derivation of candidate words based on a text input in accordance with some embodiments. FIG. **6** illustrates an example of the identification of candidate words from an input sequence.

[0062] In FIG. **6**, the input sequence **602** is "rheatre." For prefix strings of three characters in length, the first three characters and their corresponding neighbors **604** are identified. Here, the first character is "r" and its neighbors, in accordance with the layout **502**, are "e," "d," "f," and "t." The second character is "h," and its neighbors are "y," "u," "g," "j," "b," and "n." The third character is "e," and its neighbors are "w," and "s," "d," and "r."

[0063] From the input characters and corresponding neighbors, the character permutations **606** are determined. Each permutation is a character combination where the first character is the first input character or a neighbor thereof, the second character is the second input character or a neighbor thereof, and the third character is the third input character or a neighbor thereof. From these permutations, prefix strings are generated and compared to words in the dictionary. Examples of three-character permutations based on the input sequence **602** include "the," "rus," "rye," and "due." Words in the dictionary that have one of these strings as a prefix are identified as candidate words **608**. Examples of candidate words include "theater," "rye," "rusty," "due," "the," and "there." In other embodiments, the character permutations may include four, five, or more characters, rather than three characters.

[0064] Attention is now directed to FIGS. **7**A-**7**C, which are examples of scoring of candidate words in accordance with some embodiments. FIG. **7**A shows an input sequence and three possible candidate words that may be identified from permutations of the first three characters of the input sequence. The candidate words are compared to the input sequence character-by-character and scores for the candidate words are tallied.

[0065] In some embodiments, a score tally of a candidate word involves assigning a value for each character comparison and adding the values together. The value that is assigned for a character comparison is based on the result of the comparison. Particularly, the value is based on whether the character in the candidate word, compared to the character in the corresponding position in the input sequence, is an exact match, a neighbor on the keyboard layout, or neither. In some embodiments, the value assigned for an exact match is a predefined value N. If the characters are not an exact match but are neighbors, then the value assigned is a value αN, where α is a constant and $\alpha<1$. In some embodiments, α is 0.5. In other words, the value assigned for a neighbor match is a reduction of the value for an exact match.

[0066] In some embodiments, if the character in the candidate word is neither an exact match or a neighbor of the corresponding character in the input sequence, then the assigned value is βN, where β is a constant and $\beta<\alpha<1$. For example, β may be 0.25. In some other embodiments, β may be a function of the "distance" between the characters on the keyboard layout. That is, β may be a smaller number if the candidate word character is farther away on the keyboard layout from the input sequence character than if the candidate word character is closer on the keyboard layout from the input sequence character without being a neighbor.

[0067] More generally, the value assigned for a character comparison is γN, where N is a predefined value, $\gamma=1$ for an exact match, and γ may vary based on some function of the "distance" on the layout between the character in the candidate word and the corresponding character in the input sequence. For example, γ may be 1 for an exact match, 0.5 for a neighbor, and 0 otherwise. As another example, γ may be 0.5 for a neighbor (a 1-key radius), 0.25 for keys that are two keys away (a 2-key radius), and 0 for keys that are three or more keys away. In some embodiments, N is equal to 1.

[0068] If the candidate word has a length that is longer than the input sequence, or vice versa, then the character positions that are beyond the lesser of the two lengths are ignored or assigned a value of 0.

[0069] The first candidate word shown in FIG. **7**A is "theater." Compared to the input sequence of "rheatre," there are exact matches in the second thru fifth positions. The characters in the first, sixth, and seventh positions of the candidate word are keyboard layout neighbors of input sequence characters in the corresponding positions. Thus, the score for "theater" in this case is 0.5N+N+N+N+N+0.5N+0.5N=5.5N.

[0070] The second candidate word is "threats." Compared to the input sequence of "rheatre," there is an exact match in the second position. The characters in the first, third, sixth, and seventh positions of the candidate word are keyboard layout neighbors of the input sequence characters in the corresponding positions, and the characters in the fourth and fifth positions of the candidate word are neither exact matches nor neighbors of the input sequence characters in the corresponding positions. Thus, the score for "threats" in this case is 0.5N+N+0.5N+0.25N+0.25N+0.5N+0.5N=3.5N.

[0071] The third candidate word is "there." Compared to the input sequence of "rheatre," there is an exact match in the second and third positions. The character in the first position of the candidate word is a keyboard layout neighbor of the input sequence character in the corresponding position, and the characters in the fourth and fifth positions of the candidate word are neither exact matches nor neighbors of the input sequence characters in the corresponding positions. Furthermore, because the input sequence is two characters longer than the candidate word, the last two characters in the input sequence are ignored in the comparison and are assigned score values of 0. Thus, the score for "there" in this case is 0.5N+N+N+0.25N+0.25N=3N.

[0072] Some candidate words, when compared to the input sequence, may merit a score bonus, examples of which are shown in FIGS. **7**B and **7**C. In FIG. **7**B, the input sequence is "thaeter" and the candidate word is "theater." The score based on the character comparisons alone is 5.5N. However, the only difference between "thaeter" and "theater" is a pair of transposed or swapped characters, namely "ae" in "thaeter" vs. "ea" in "theater." In some embodiments, a first bonus P is added to the score for this fact. In FIG. **7**C, the input sequence is "thester" and the candidate word is "theater." The score based on the character comparisons alone is 6.5N. However, the only difference between "thester" and "theater" is a single character, namely "s" in "thester" vs. "a" in "theater." In some embodiments, a second bonus Q is added to the score for this fact. In some embodiments, both P and Q are equal to 0.75.

[0073] It should be appreciated that, in some other embodiments, alternative candidate word scoring and selection schemes other than the ones described above may be used.

[0074] For example, one alternative scheme may include, instead of dividing the candidate words into the first and second groups based on usage frequency rankings, the usage frequency rankings may instead be used as a weighting to be