Optilearn.word.short_word_treatment

08 November 2024 19:54

short_word_treatment

The short_word_treatment function is designed to expand abbreviations, acronyms, and other commonly used short words in text to their full forms. This function is flexible and customizable, allowing users to add or remove short words as needed or clear the default short words entirely to rely on a custom set. This makes it ideal for applications where text normalization is important, such as in chat processing, text analysis, or document cleaning.

Parameters

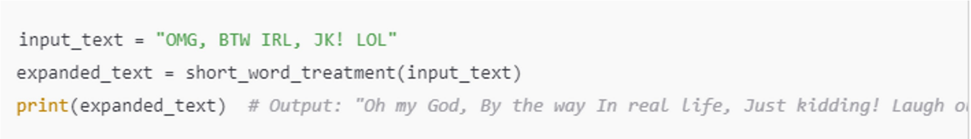
1. **text** (str):
   - The input text string to be processed.
2. **add_words** (dict, optional):
   - A dictionary where keys are short words and values are their corresponding expanded forms. These custom mappings will be added to the predefined short word dictionary.
   - **Default**: None.
3. **remove_words** (list, optional):
   - A list of short words to be removed from the predefined dictionary. This allows selective removal of default expansions.
   - **Default**: None.
4. **remove_all** (bool, optional):
   - If set to True, clears the predefined dictionary of short words, so only user-defined short words in add_words are used.
   - **Default**: False.
5. **text_case** (str, optional):
   - Controls the case of the output text:
     - 'same': Retains the original case of the input text (default).
     - 'lower': Converts the output to lowercase.
     - 'upper': Converts the output to uppercase.
   - **Default**: 'same'.

Returns

- **str**: The processed text where any recognized short words have been expanded based on the rules specified.

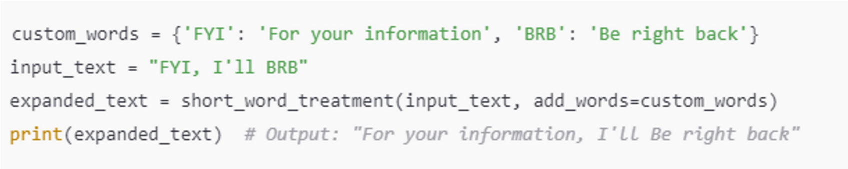
Example Usage

Basic Expansion



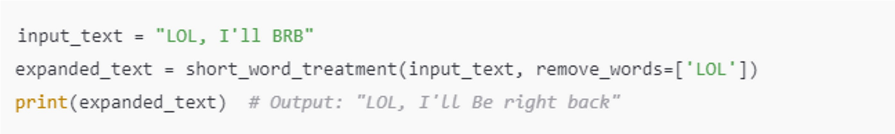
```
input_text = "OMG, BTW IRL, JK! LOL"
expanded_text = short_word_treatment(input_text)
print(expanded_text)  # Output: "Oh my God, By the way In real life, Just kidding! Laugh o
```

Adding Custom Words



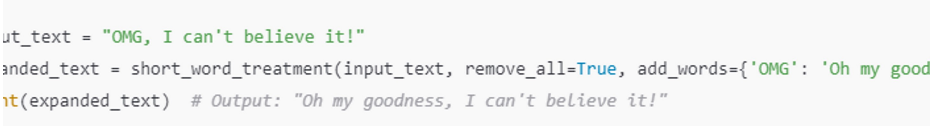
```
custom_words = {'FYI': 'For your information', 'BRB': 'Be right back'}
input_text = "FYI, I'll BRB"
expanded_text = short_word_treatment(input_text, add_words=custom_words)
print(expanded_text)  # Output: "For your information, I'll Be right back"
```

Removing Specific Words



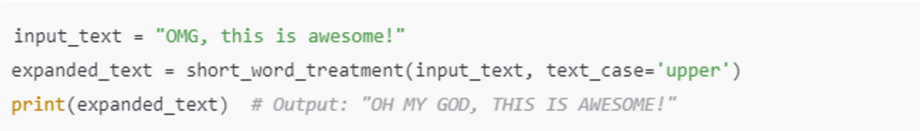
```
input_text = "LOL, I'll BRB"
expanded_text = short_word_treatment(input_text, remove_words=['LOL'])
print(expanded_text)  # Output: "LOL, I'll Be right back"
```

Clearing Predefined Words and Using Custom Words



```
ut_text = "OMG, I can't believe it!"
anded_text = short_word_treatment(input_text, remove_all=True, add_words={'OMG': 'Oh my good
nt(expanded_text)  # Output: "Oh my goodness, I can't believe it!"
```

Setting Case to Upper



```
input_text = "OMG, this is awesome!"
expanded_text = short_word_treatment(input_text, text_case='upper')
print(expanded_text)  # Output: "OH MY GOD, THIS IS AWESOME!"
```

Notes

- **Selective Word Retention**: To keep only specific words in the vocabulary, set remove_all=True and specify only those words in add_words.
- **Handles Punctuation**: The function ensures that punctuation and special characters are correctly managed, expanding only the specified short words.
- **Error Handling**: Designed to handle cases where inputs might be missing or incorrect, making it robust in various scenarios.

Benefits

This function provides a customizable way to expand short forms in text, ensuring that text is easily readable and standardized for downstream applications. The flexibility to modify the dictionary of short words, combined with options for case control and handling punctuation, makes this function highly adaptable for many text processing workflows.