**Output Sorting Guidelines**

The sorting of outputs is divided into two main parts: The first part involves organizing the results of the model output (in txt format, containing words and labels, conventional metaphors will be labelled with “:1”), and the second part involves generating a csv file from the organized results of the first part, using the words and labels for score calculation.

1. Sorting of Raw Text Outputs

The objective here is to ensure the text is in the correct format (list words and labels) for csv file generation. The encountered issues and their solutions include:

* **Word Formatting Issues**:
  + **Problem**: Words are not on separate lines but are placed on the same line, separated by spaces or commas.
  + **Solution**: Use code to format them into one word per line.
* **Self-Created Word Groups and Labels**:
  + **Problem**: The model creates its own word groups and assigns labels to the entire group. For example,

*Standard: Letter*

*Policy*

*Output: Letter Policy*

* + **Solution**: Require the model to output again.
* **Marks Before Words**:
  + **Problem**: There are marks (e.g. ’’’) before words.
  + **Solution**: Remove the marks.
* **Retaining Original Sentences and Word Lists**:
  + **Problem**: Both the original sentences and word lists are retained. For example,

*Standard: Time*

*flies:1*

*Output: Time flies*

*Time*

*flies:1*

* + **Solution**: Delete the original sentences.
* **Separate Word Lists and Labels**:
  + **Problem**: Word lists and labels are separate. For example,

*Output: Time*

*flies*

*Here are the conventional metaphorical words:*

*flies:1*

* + **Solution**: Merge the labels with the word lists.
* **Including Additional Content**:
  + **Problem**: Includes explanations of words, opening or closing sentences, or symbols. For example,

*Output:*

*Step 1: Analyze every word in each given sentence to identify conventional metaphors.*

*\*\*Conventional metaphors identified in the sentences:\*\**

*1. (No conventional metaphor identified)*

*2. "set+up" (could be interpreted as a metaphor for establishing), "ended" (metaphor for cessation), "futile" (metaphor for lack of success), "infighting" (metaphor for conflict)*

*……*

*Step 2: List every word in the given sentence sequentially*

* + **Solution**: Delete non-list content, retaining only the word list.
* **Partial Word Lists**:
  + **Problem**: Only a part of the word list is enumerated. For example,

*Output:*

*Since there are 21 sentences and the task is quite lengthy, I will demonstrate the process using the first three sentences. Here's how the lists would start:*

*……*

* + **Solution**: Require the model to output the complete list.
* **All Words Marked as 1**:
  + **Problem**: Every word in a sentence is marked as 1. For example,

*Standard: Time*

*flies:1*

*Output: Time:1*

*flies:1*

* + **Solution**: Retain this judgment as the model's output may vary.
* **Words and labels are not on the same row**:
  + **Problem:** Words and labels are on different rows. For example,

*Standard: Time*

*flies:1*

*Output: Time*

*flies*

*:1*

* + **Solution:** Put the labels back to the same rows with the words.

2. Sorting of Raw CSV (Raw csv) Outputs

After sorting the word lists in the txt files to right format (word:label, one word per row), the sorted outputs are then transformed into two columns in a csv file. These are compared with the manual word and label columns. The process involves handling misalignments between the machine output word list and the manual word list, mainly in two categories:

* **Mismatch in Word Tokenization**:
  + **Problem**: Words with different tokenization from the manual version.
* Words are split differently than expected. For example,

*Manual: Runner+beans*

*Output: Runner*

*Beans*

* Words are combined differently than expected. For example,

*Manual: Letter*

*Policy*

*Output: Letter Policy*

* Words are partially split differently than expected. For example,

*Manual: set+up*

*Output: up*

* + **Solution**: Since there is no clear pattern for when and how the word tokenization problems occur, manually check and adjust are used in this step:
* For problem one and three, if a part of the word pair is labelled as 1, then the whole word pair will be labelled as 1.
* For problem two, if the word pair is labelled as 1, then every separate words will be labelled as 1.
* **Incomplete word:label Lists**:
  + **Problem**: The output word list is incomplete.
  + **Solution**: Align using Python, with missing labels defaulted to 0.