```
def process_full_text(full_text):
    if 'Unprocessed:' in full_text: # Need to split the instruction and response
        pattern = r'([*]*Please write an essay about [^*]*[*]*)'
        x = re.search(pattern, full_text)
        if x:
            splits = re.split(pattern, full_text)
            text = splits[-1].strip()
            # print(f"=== Split text ===\n {text}")
            return text
        else:
            print(f"### UnProcess text###\n {full_text}")
            return None
    else:
        return full_text.strip() # Remove the space to the left

def tokenize_with_spacy(text, tokenizer=en_tokenizer):
    tokenized_text = tokenizer(text)
    tokens = [token.text for token in tokenized_text]
    trailing_whitespace = [bool(token.whitespace_) for token in tokenized_text]
    return tokens, trailing_whitespace


full_text_check = process_full_text(original_text['text'][0])
if full_text_check:
        # Tokenize the text using spacy tokenizer
        tokens_check, trailing_whitespace_check = tokenize_with_spacy(full_text_check)
        row2_check = essay.iloc[0]
        #labels_check = assign_labels(row2_check, tokens_check)
```