SOFTWARE

LLM WORD CHECKER

4.07.2025

The recent Science Advance paper by Kobak et al. studied

> vocabulary changes in more than 15 million biomedical abstracts from 2010 to 2024 indexed by PubMed and show how the appearance of LLMs led to an abrupt increase in the frequency of certain style words. This excess word analysis suggests that at least 13.5% of 2024 abstracts were processed with LLMs.

Although they say that the analysis was performed on the corpus level and cannot identify individual texts that may have been processed by a LLM, we can of course check the proportion of LLM words in a text.

Unfortunately their online list contains stop words that I am eliminating here. But then we can run the following script!

```
# based on https://github.com/berenslab/llm-excess-vocab/tree/main

import csv
import re
import os
from collections import Counter
from striprtf.striprtf import rtf_to_text
from nltk.corpus import stopwords
import nltk
import chardet

# Ensure stopwords are available
nltk.download('stopwords')

# Paths
```

```
rtfd_folder_path = '/Users/x/Desktop/mss_image.rtfd' # RTFD is a directory
rtf_file_path = os.path.join(rtfd_folder_path, 'TXT.rtf') # or 'index.rtf'
csv_file_path = '/Users/x/Desktop/excess_words.csv'

# Read and decode the RTF file
with open(rtf_file_path, 'rb') as f:
raw_data = f.read()

# Try decoding automatically
encoding = chardet.detect(raw_data)['encoding']
rtf_content = raw_data.decode(encoding)
plain_text = rtf_to_text(rtf_content)

# Normalize and tokenize text
words_in_text = re.findall(r'\b\w+\b', plain_text.lower())

# Remove stopwords
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words_in_text if word not in stop_words]

# Load excess words from CSV
with open(csv_file_path, 'r', encoding='utf-8') as csv_file:
reader = csv.reader(csv_file)
excess_words = {row[0].strip().lower() for row in reader if row}

# Count excess words in filtered text
excess_word_counts = Counter(word for word in filtered_words if word in excess_words)

# Calculate proportion
total_words = len(filtered_words)
total_excess = sum(excess_word_counts.values())
proportion = total_excess / total_words if total_words &gt; 0 else 0

# Output
print("\nExcess Words Found (Sorted by Frequency):")
for word, count in excess_word_counts.most_common():
print(f"{word}: {count}")
print(f"\nTotal words (without stopwords): {total_words}")
print(f"Total excess words: {total_excess}")
print(f"Proportion of excess words: {proportion:.4f}")
```

7 Aug 2025

The long ’em dash’ — U+2014 instead of the standard minus – seems to be a characteristic sign of chatGPT 4 even when asked not use it.