**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.

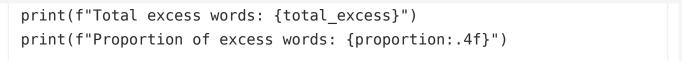
Unfortunately their online list contains stop words that I am eliminating here. But then we can run the following script!

```
# 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 > 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}")
```



## 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.

CC-BY-NC Science Surf 4.07.2025, access 18.10.2025 ☐