Computational Psycholinguistics

Assignment 3

Nidhi Vaidya
2023114005

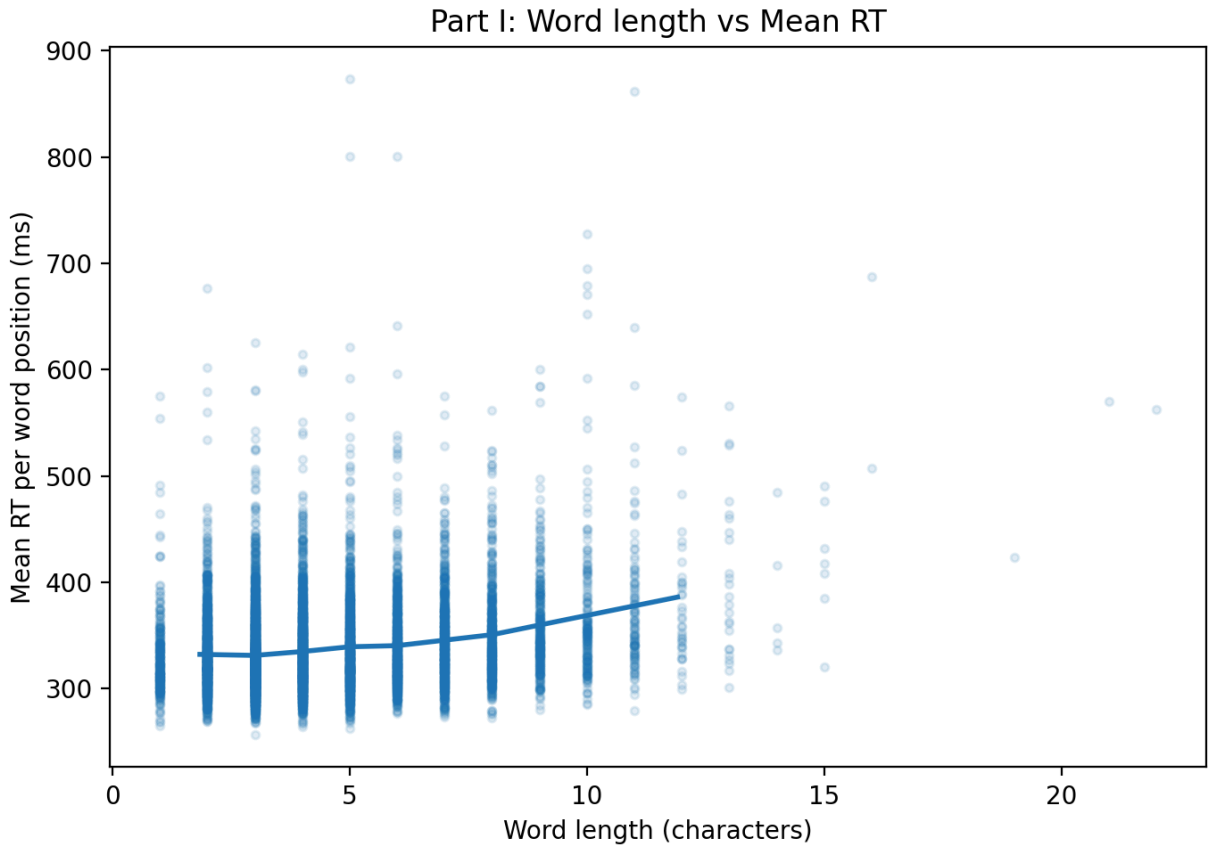
GitHub repository link: `https://github.com/nidhi-00/Word-Processing.git`

Part I: Preliminary Data Analysis

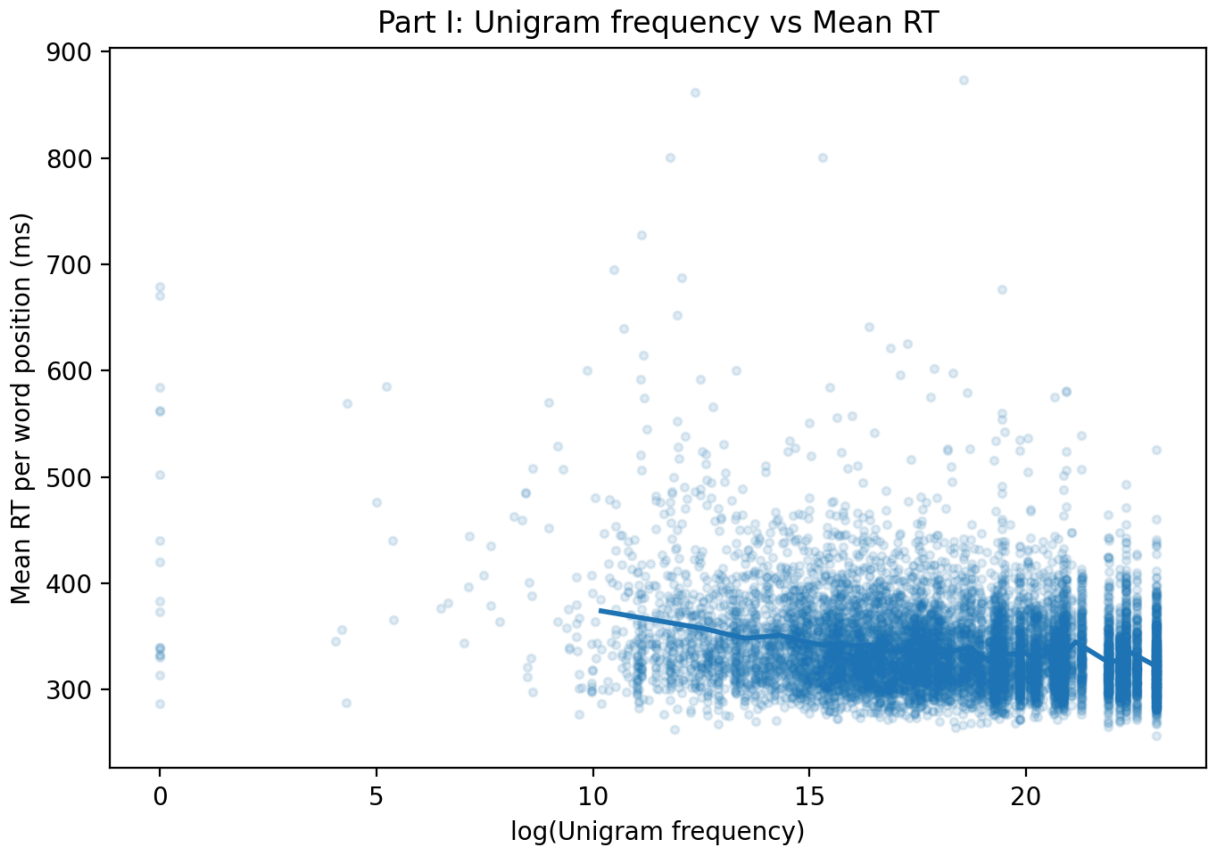
Mean RT per word

For each word position in `processed_RTs.tsv`, I computed the mean **RT per word** by averaging RT across all subjects for the same (item, zone, word).

- **Word length (characters) vs mean RT**:



- **Word frequency vs mean RT** (using log unigram frequency for visualization stability):



Pearson correlations

- **r(length, log freq)** = **−0.693729** (p ≈ 0)
- **r(length, mean RT)** = **0.232792** (p = 2.951e−126)
- **r(log freq, mean RT)** = **−0.224874** (p = 9.999e−118)

Short note

- **Longer words are less frequent** (strong negative length–frequency correlation).
- **Longer words are read more slowly** (positive length–RT correlation).
- **More frequent words are read faster** (negative frequency–RT correlation).

Part II: Hypothesis Testing

Hypothesis 1 (10 marks)

Hypothesis: LM probabilities are better predictors of reading time than word frequency.

I fit the required regressions (controlling for word length):

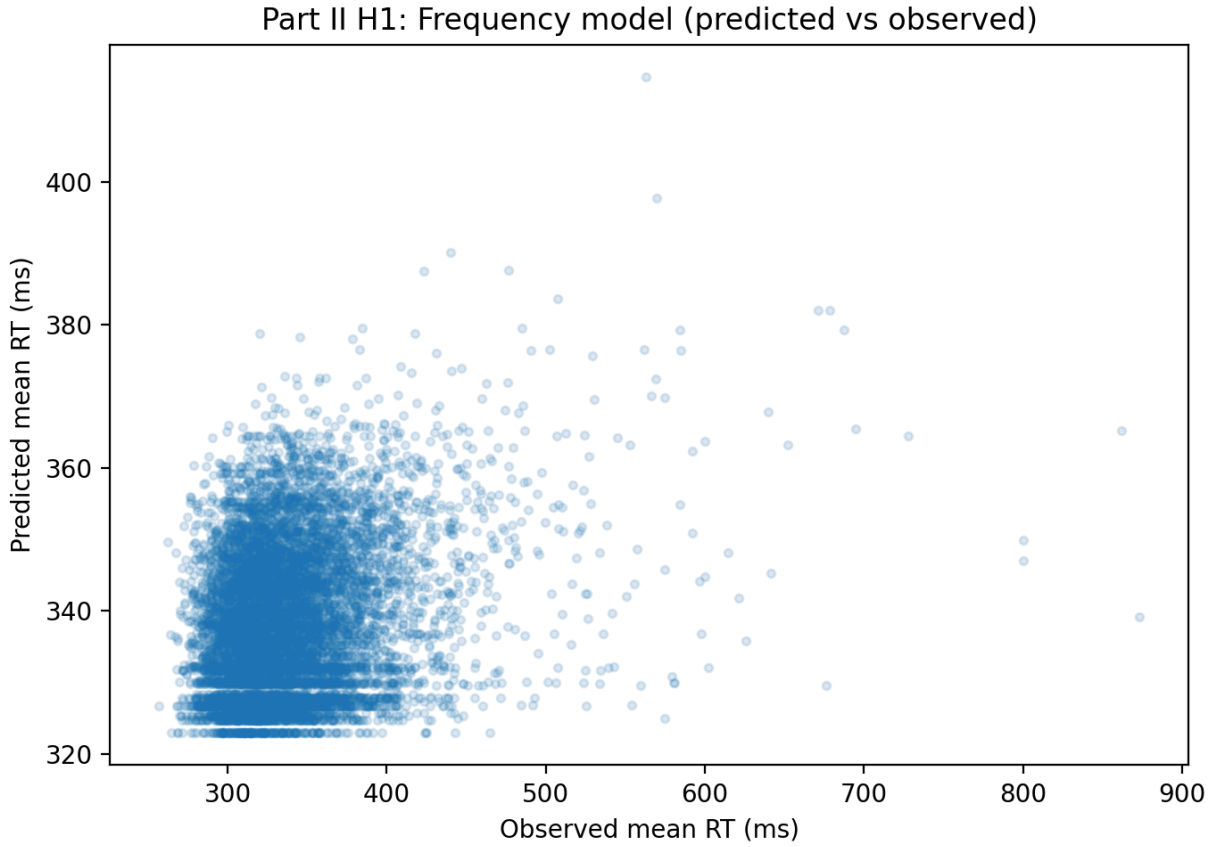
- **Model 1:** mean RT ~ word freq + word length
- **Model 2:** mean RT ~ −log(GPT-3 probability) + word length

Model fit comparison:

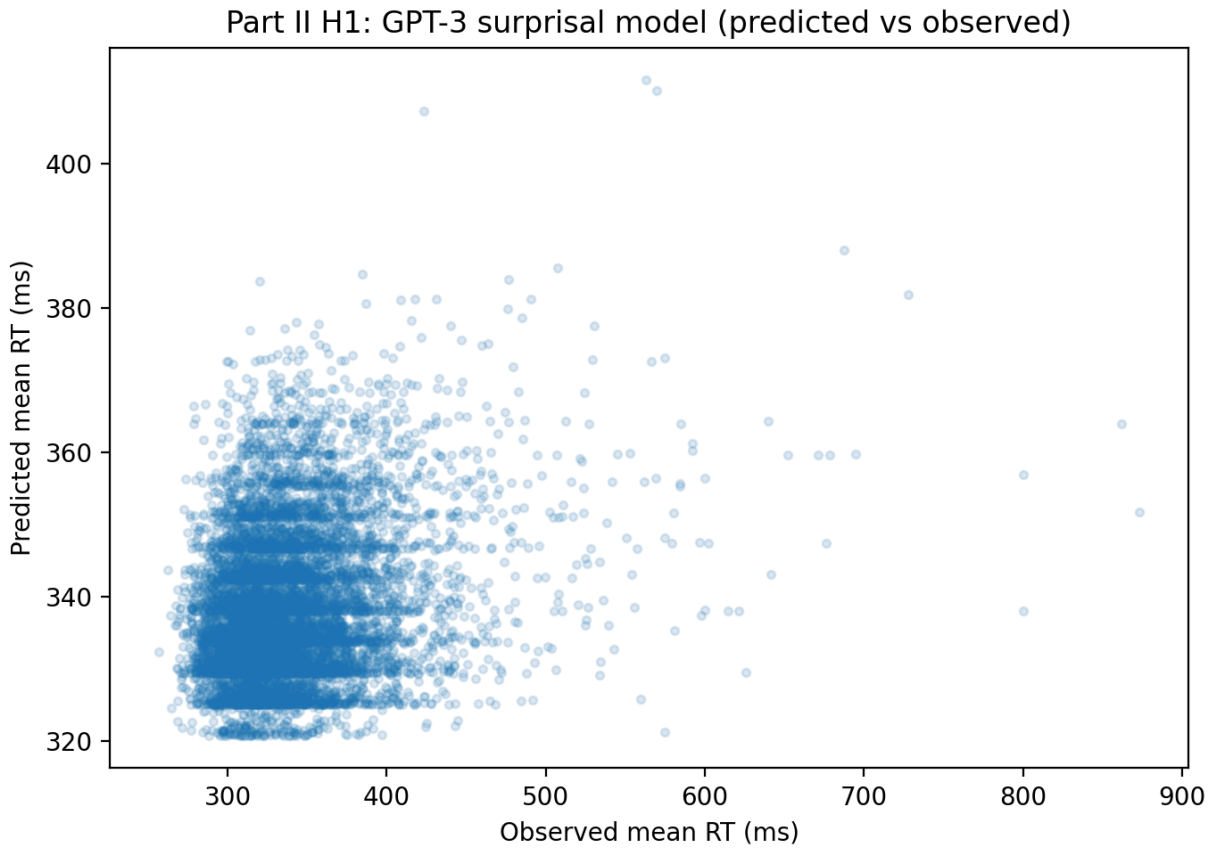
- Model A (freq): **Adj R^2 = 0.061753**, AIC = 105142.67
- Model B (GPT-3 surprisal): **Adj R^2 = 0.058313**, AIC = 105180.20

Visualizations:

- Predicted vs observed (freq):



- Predicted vs observed (GPT-3):



Brief conclusion:
Both predictors help explain mean RT in the expected direction when controlling for length, but **word frequency fits slightly better than GPT-3 surprisal** here (higher Adj R^2 and lower AIC).

Hypothesis 2 (15 marks)

Hypothesis: Content words are processed differently than function words.

I split tokens into **content** vs **function** and fit the four required regressions:

(Content words = nouns/verbs/adjectives/adverbs (POS-tagged); function words = all others)

- Content: (1) freq+len, (2) −log(GPT-3 prob)+len
- Function: (3) freq+len, (4) −log(GPT-3 prob)+len

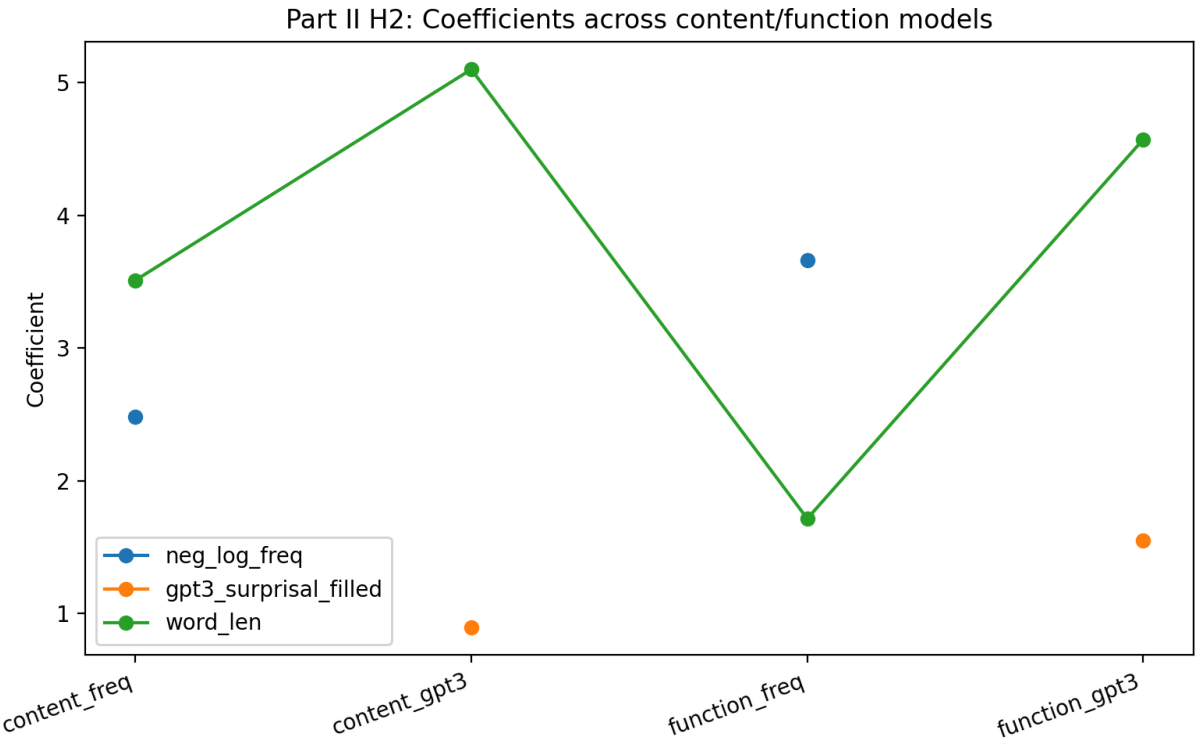
Model fit comparison:

- **Content tokens**: 5882
  - freq model: **Adj R^2 = 0.082196**
  - GPT-3 model: **Adj R^2 = 0.068617**

- **Function tokens**: 4374
    - freq model: **Adj R^2 = 0.048024**
    - GPT-3 model: **Adj R^2 = 0.031753**

Visualization (required):

- Coefficient comparison across the four models:



Brief conclusion:
Content vs function show different overall fit levels (content models fit better than function models here), but **within each category, frequency outperforms GPT-3 surprisal** in this dataset/run (higher Adj R^2 for freq in both content and function).

Part III: Frequency Ordered Bin Search (FOBS)

FOBS setup (5 marks)

I used a **standard lemmatizer/lexicon** (WordNet-based lemmatization with POS tagging) to obtain **roots/lemmas**, and I constructed lemma families and their frequencies by aggregating surface-form frequencies within each lemma family, enabling a lemma-frequency ordering consistent with the FOBS idea.

(Generated lemma table: `outputs/tables/partIII_H1_lemma_table.csv`.)

Hypothesis 1 (10 marks)

Hypothesis: Root frequency predicts reading times better than surface frequency.

I compared the required models:

- **Model 1:** mean RT ~ word freq + word length
- **Model 2:** mean RT ~ lemma freq + lemma length

Model fit comparison:

- Surface model: **Adj R^2 = 0.061753**, AIC = 105142.67
- Lemma model: **Adj R^2 = 0.058613**, AIC = 105176.93

Short note:
In this run, **surface frequency predicts mean RT slightly better than lemma frequency** (higher Adj R^2 and lower AIC for the surface model).

Hypothesis 2 (5 marks)

Hypothesis: Pseudo-affixed words like “finger” take more processing time than regularly affixed words like “driver.”

Task: Take **5 words** of approximately the same word length and frequency containing **real and pseudo affixes**, and test the hypothesis comprehensively.

Selected words (matched set)

Using the script’s matching procedure (matched on length and similar frequency), the following **5 pseudo-affix** and **5 real-affix** words were selected (see `outputs/tables/partIII_H2_matched_pairs.csv`):

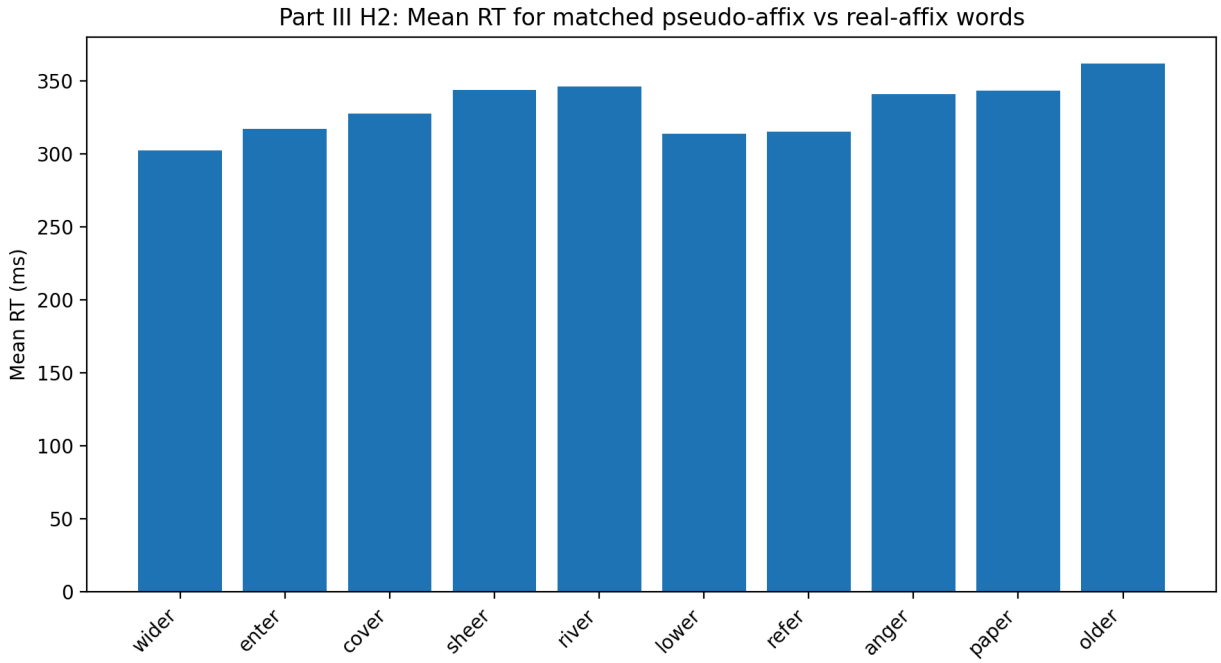
- **Pseudo-affix (5):** sheer, wider, enter, river, cover
- **Real-affix (5):** anger, refer, older, paper, lower

(All selected words are matched to length = 5 characters in this run, and frequencies are in the saved matched-pairs table.)

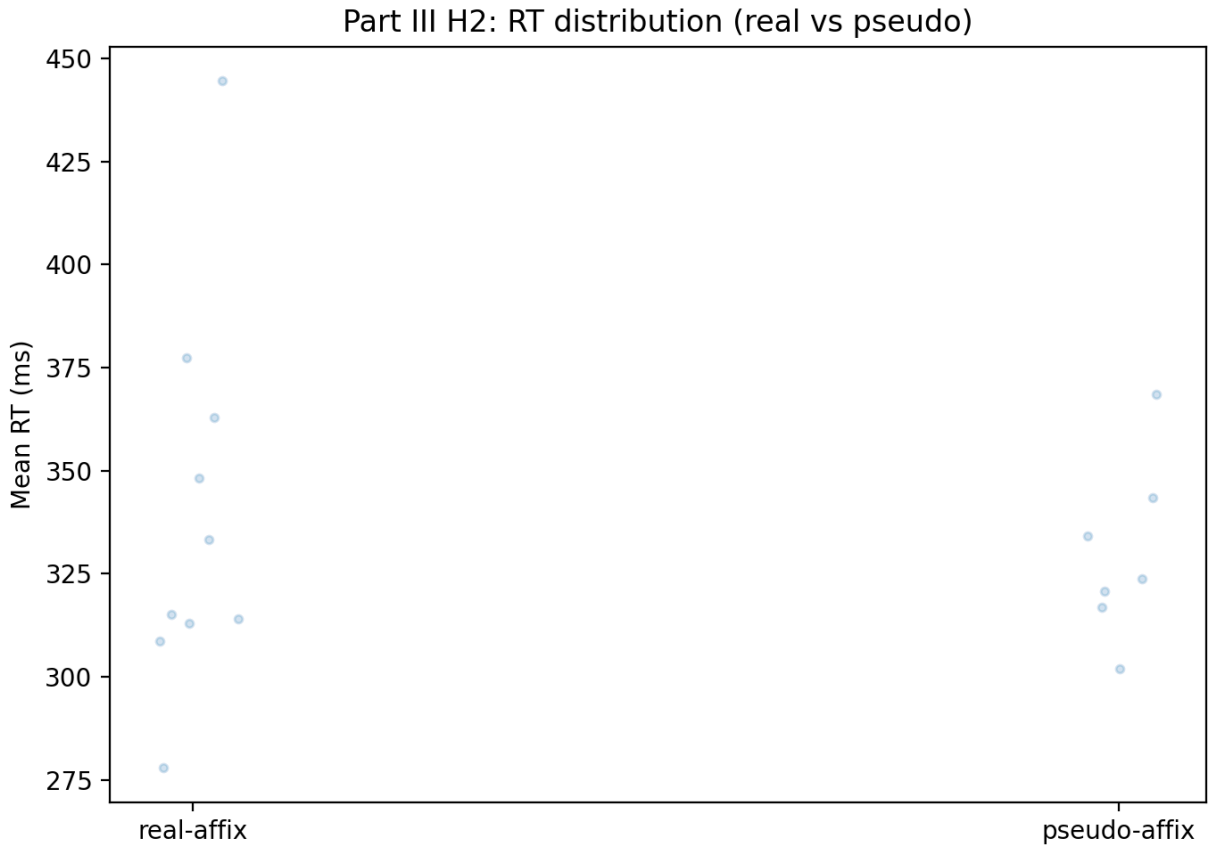
Tests and visualizations

Visualizations:

- Mean RT per selected word:



- RT distribution by class (real vs pseudo):



Statistical tests:

- Welch t-test (pseudo vs real): **t = −0.568623**, **p = 0.5790**
- Regression controlling for frequency and length:
  mean RT ~ is_pseudo + −log(freq) + word_len
  is_pseudo coef = −11.265244, **p = 0.6119**

Short note

With these matched sets, **pseudo-affixed words were not significantly slower** than real-affixed words in this run (non-significant Welch t-test and non-significant is_pseudo effect in the regression).