# Spell Checker

Ting-Wei Shen
Data Science for Linguistics
April 11, 2019

# Outline

- Spell checker foundation
  - the basic four elements
- Data processing
  - ELI dataset – raw data without precleaning
  - the fair use of data
- Linguistic analysis
  - writing quality assessment
  - check out spell checker on our dataset

# The main idea- Some Probability Theory

- Most likely spelling correction c for w

$$\text{argmax}_{c \in candidates} P(c|w)$$

- By *Bayes' Theorem*, this is equivalent to:

$$\text{argmax}_{c \in candidates} P(c)\, P(w|c) / P(w)$$

- Since P(w) is the same for every possible candidate c, we can factor it out, giving:

$$\text{argmax}_{c \in candidates} P(c)\, P(w|c)$$

# Divided into four main parts

- **Selection Mechanism**: *argmax*
  - Choose the candidate with the highest combined probability.
- **Candidate Model**: $c \in candidates$
  - This tells us which candidate corrections, *c*, to consider.
- **Language Model**: *P(c)*
  - The probability that *c* appears as a word of English text.
  - For example, occurrences of "the" make up about 7% of English text, so we should have $P(the) = 0.07$.
- **Error Model**: $P(w \mid c)$
  - The probability that *w* would be typed in a text when the author meant *c*.
  - For example, $P(teh \mid the)$ is relatively high, but $P(theeexyz \mid the)$ would be very low.

# Some assumptions on Error Model - candidates(word)

1. The original word, if it is known; otherwise
2. The list of known words at edit distance one away, if there are any; otherwise
3. The list of known words at edit distance two away, if there are any; otherwise
4. The original word, even though it is not known.

```
def correction(word):
    "Most probable spelling correction for word."
    return max(candidates(word), key=P)

def candidates(word):
    "Generate possible spelling corrections for word."
    return (known([word]) or known(edits1(word)) or known(edits2(word)) or [word])
```

# How does ELI data look like?

- Original: 46239 files
  - student_id, question_id, text, gender, native_language, question
- After data processing: 492 files
  - student_id, question_id, text, gender, native_language, question, tokens, token_count, types, type_count, TTR

```
df_eli_revised.head()
```

| | student_id | question_id | text | gender | native_language | question | tokens | token_count | types | type_count | TTR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **43439** | gu2 | 5796 | They see the online shopping do pollution but ... | NaN | NaN | Chapter 5: "A Cleaner Way to Shop?" \n\nChoos... | [They, see, the, online, shopping, do, polluti... | 109 | {multiple, smaller, heard, do, opposed, goods,... | 79 | 0.724771 |
| **753** | cq4 | 37 | 1-Mismanagement\nSentence: So if they were nom... | Female | Arabic | For each of the following words, write the sen... | [1-Mismanagement, Sentence, :, So, if, they, w... | 198 | {persuade, method, speech, The, 2, someone, wa... | 116 | 0.585859 |
| **23677** | dp5 | 3279 | What I do everyday is boring. I wake up 8:45... | Male | Korean | Be sure to write a paragraph with at least 7 s... | [What, I, do, everyday, is, boring, ., I, wake... | 109 | {homework, shower, smoke, do, same, for, of, c... | 60 | 0.550459 |
| **39150** | gw1 | 5238 | Before making a decision about what I will stu... | Male | Arabic | Acknowledge Ambiguous Crucial Compensat... | [Before, making, a, decision, about, what, I, ... | 274 | {Aziz, understand, chance, thinking, The, me, ... | 133 | 0.485401 |
| **23987** | ce2 | 3288 | While I was living in Germany, I met my f... | Male | Arabic | Your topic sentence can be: "I met my friend ... | [While, I, was, living, in, Germany, ,, I, met... | 79 | {living, same, friend, second, of, me, things,... | 47 | 0.594937 |

# Loading chosen data

1. answer.csv
   - student_id, question_id, text
2. student_information.csv
   - student_id, gender, native_language
3. question.csv
   - question_id, question

# Sample data

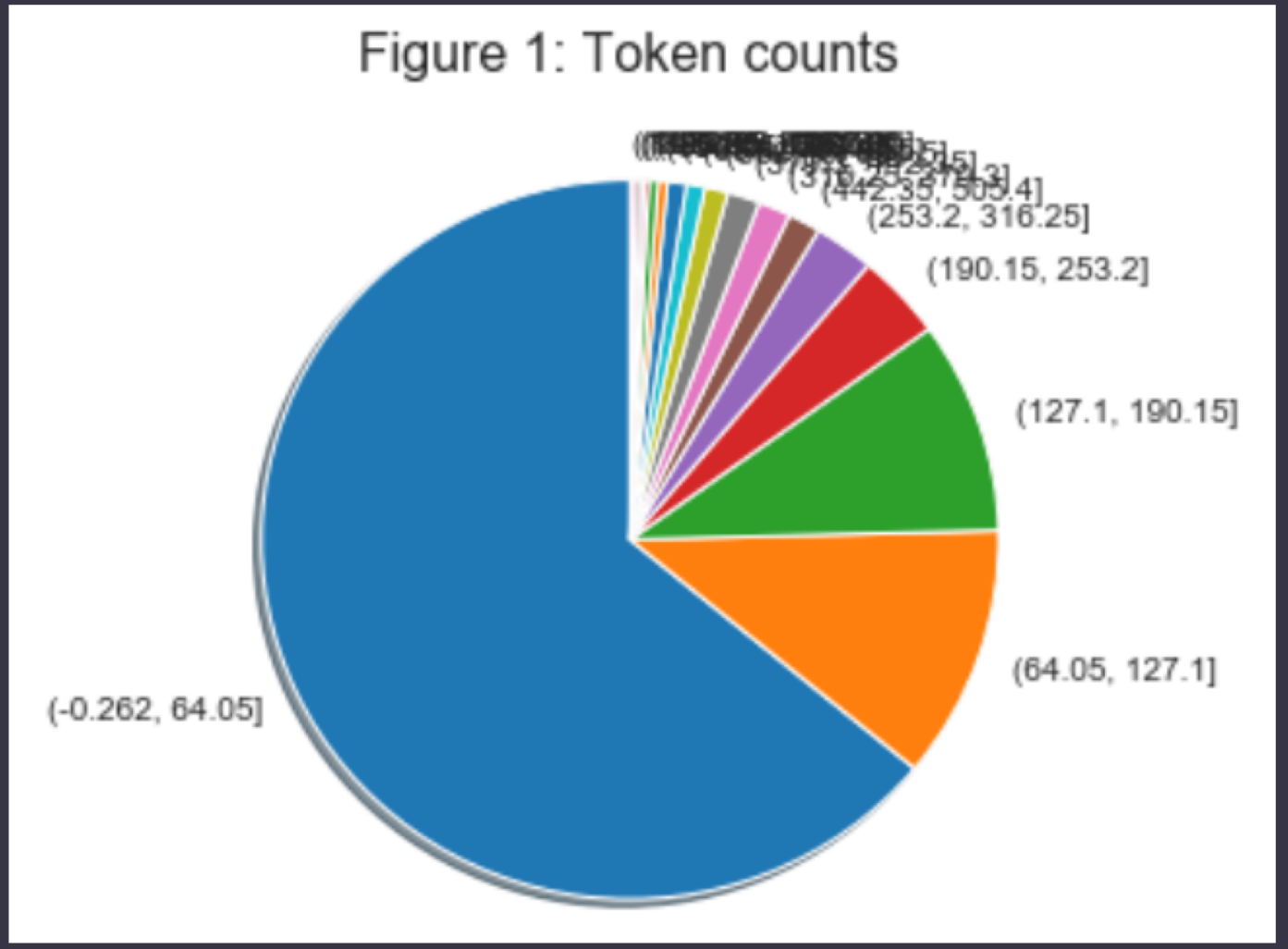
1. Consider the fair use – 500 texts to represent
2. Sample data to avoid data bias
   - random_state = 1 to fix the data
3. Eliminate personal information
4. Output as a new csv file – 492 texts left

# Challenge I met - Text length as token count

df_tokens

| | |
|---|---|
| (-0.262, 64.05] | 315 |
| (64.05, 127.1] | 56 |
| (127.1, 190.15] | 47 |
| (190.15, 253.2] | 19 |
| (253.2, 316.25] | 13 |
| (442.35, 505.4] | 7 |
| (316.25, 379.3] | 7 |
| (379.3, 442.35] | 7 |
| (568.45, 631.5] | 5 |
| (631.5, 694.55] | 4 |
| (505.4, 568.45] | 4 |
| (694.55, 757.6] | 2 |
| (1009.8, 1072.85] | 2 |
| (820.65, 883.7] | 1 |
| (946.75, 1009.8] | 1 |
| (1072.85, 1135.9] | 1 |
| (1198.95, 1262.0] | 1 |

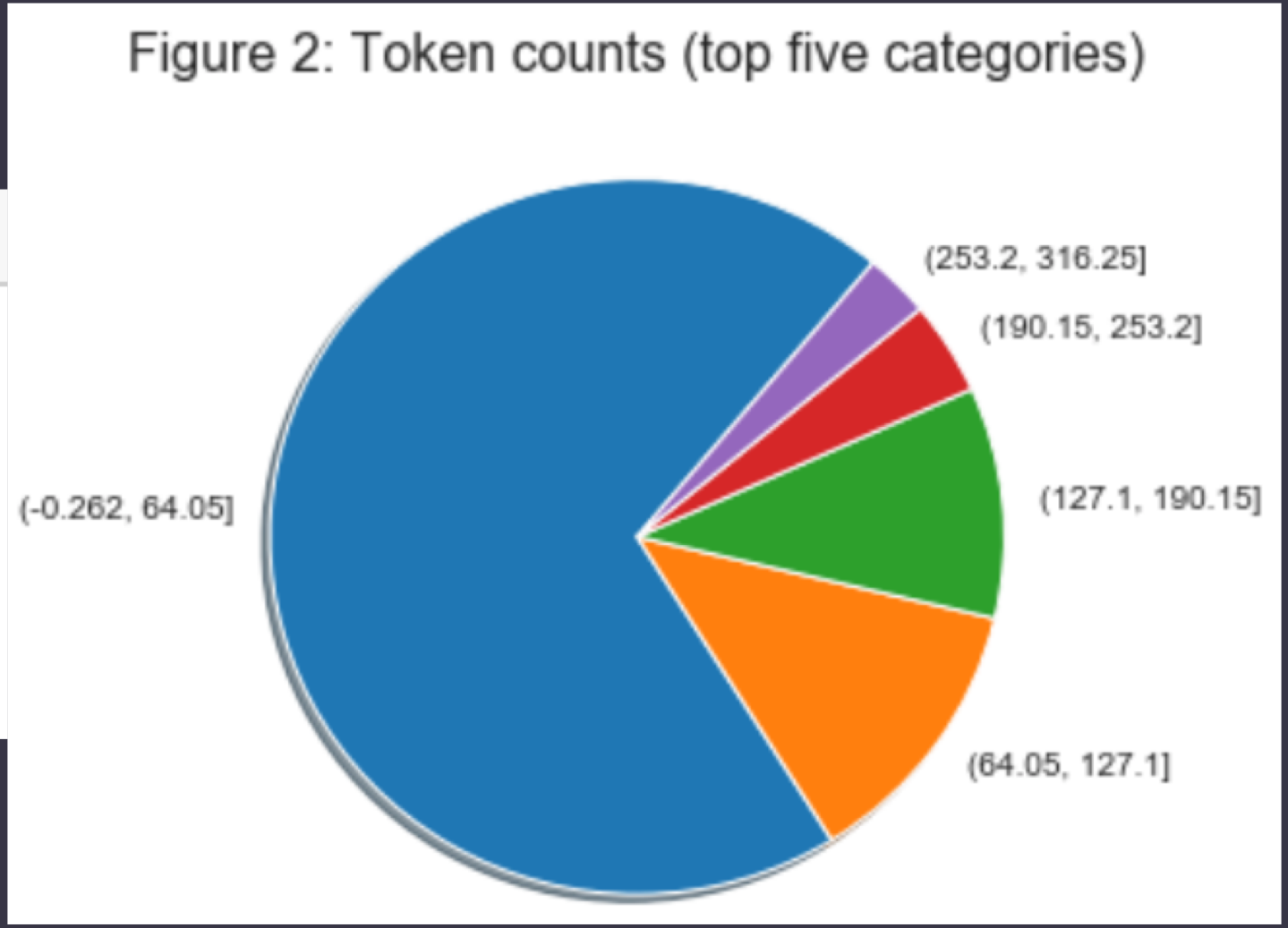
Name: , dtype: int64



# Token counts (top five categories)

```
df_tokens_5
```

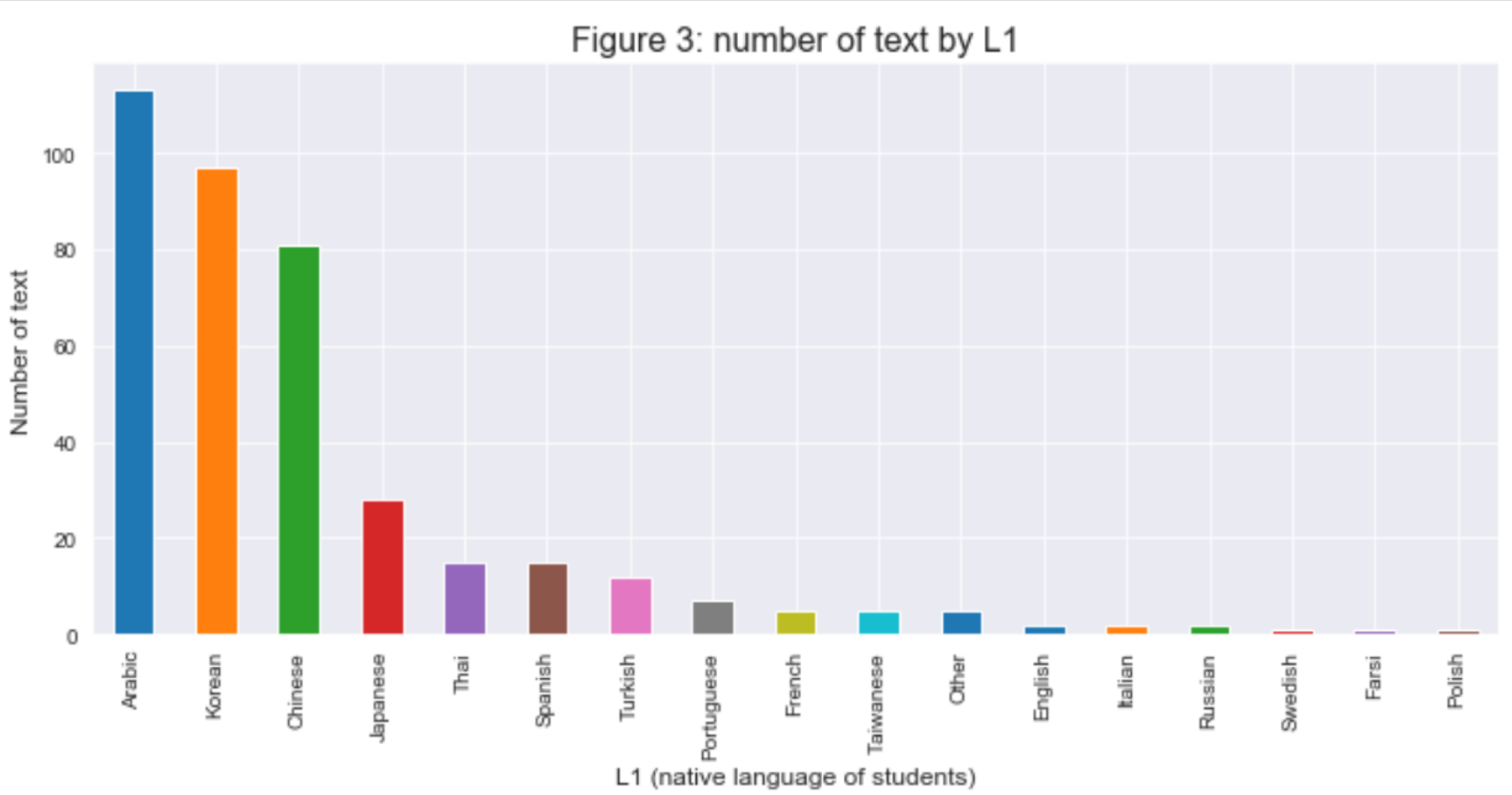
```
(-0.262, 64.05]     315
(64.05, 127.1]       56
(127.1, 190.15]      47
(190.15, 253.2]      19
(253.2, 316.25]      13
Name: , dtype: int64
```



# Number of text by native languages

df_native

| | |
|---|---|
| Arabic | 113 |
| Korean | 97 |
| Chinese | 81 |
| Japanese | 28 |
| Thai | 15 |
| Spanish | 15 |
| Turkish | 12 |
| Portuguese | 7 |
| French | 5 |
| Taiwanese | 5 |
| Other | 5 |
| English | 2 |
| Italian | 2 |
| Russian | 2 |
| Swedish | 1 |
| Farsi | 1 |
| Polish | 1 |



# Challenge from data part

- There are 49 text files that only contains one word. These are true/false, fill in the blank and multiple choice questions.
- 315/492 = 64% of total files are under 64.05 tokens.
- Arabic, Korean and Chinese are the three main categories of native languages.

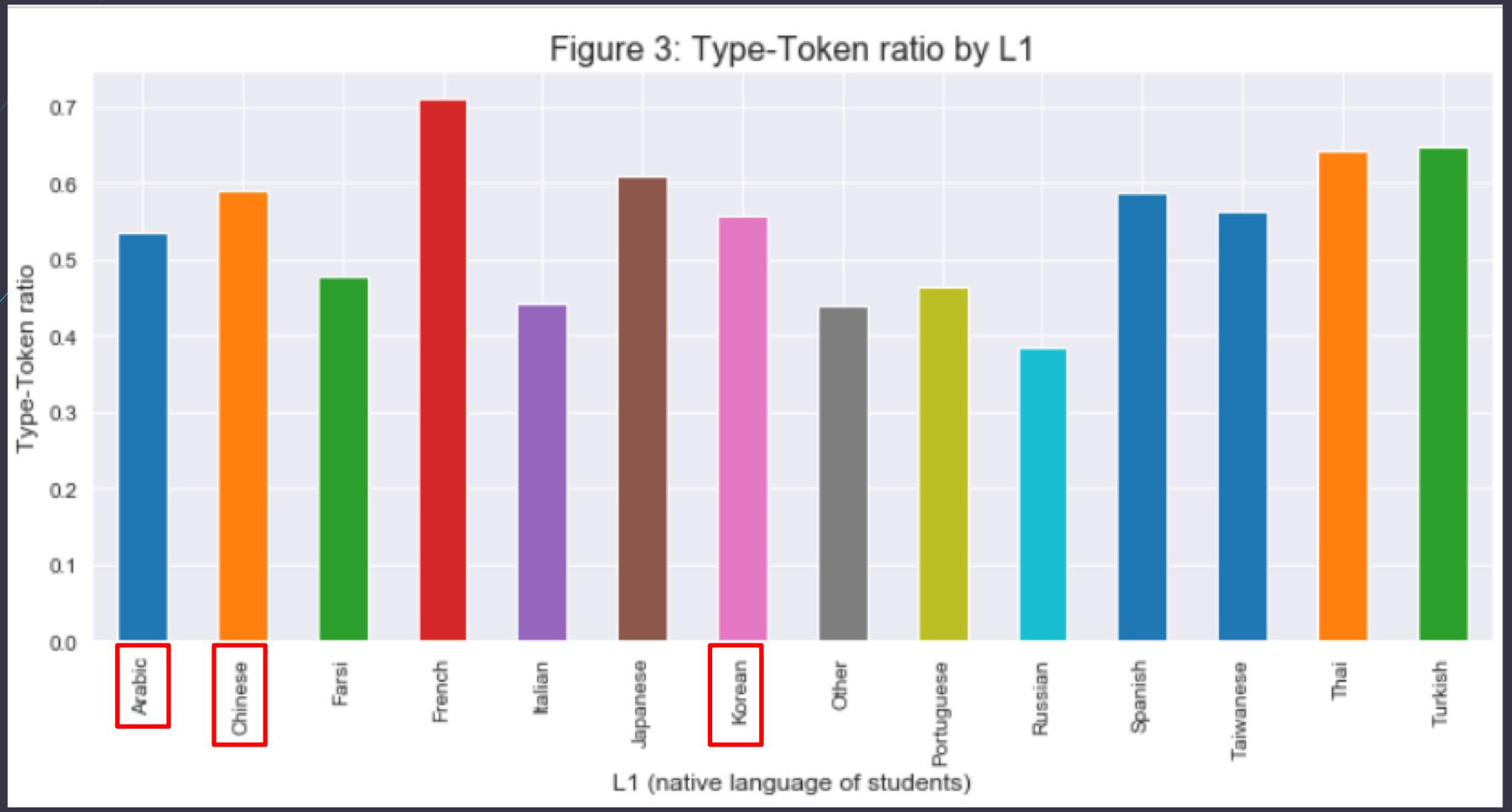
# Linguistic analysis

- By selecting essays between 50 to 600 tokens, there are 172 essays left.
- On the other hand, there are many NaN existed in the data. So the number of total available files is 132.

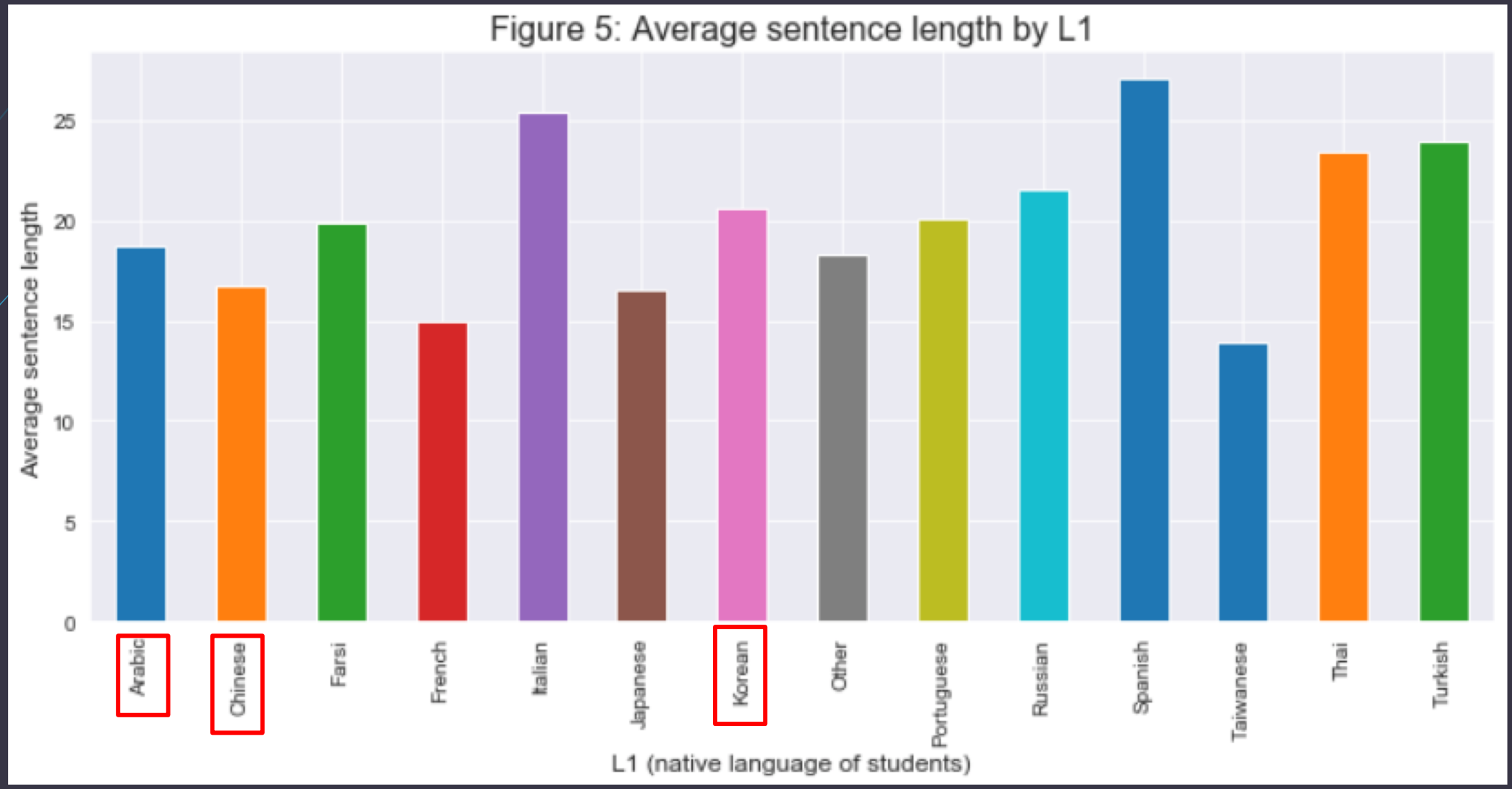
# Linguistic analysis - Assessing writing quality

1. Lexical diversity
   - Type-token ratio (TTR)
2. Syntactic complexity
   - Average sentence length
3. Vocabulary level
   - Average word length, % of word tokens in top 1K, 2K, 3K most common English words

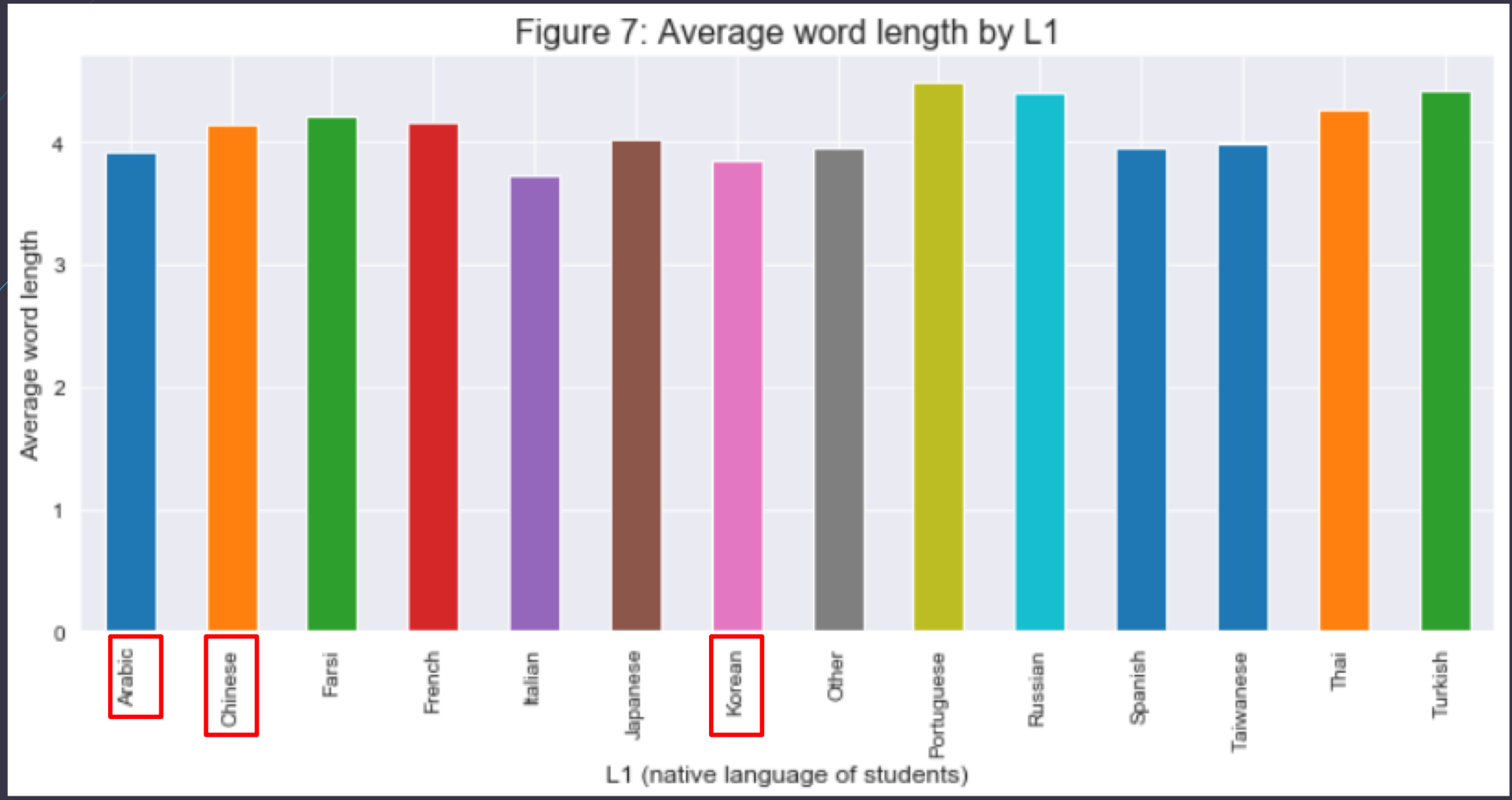
# 1. Lexical diversity



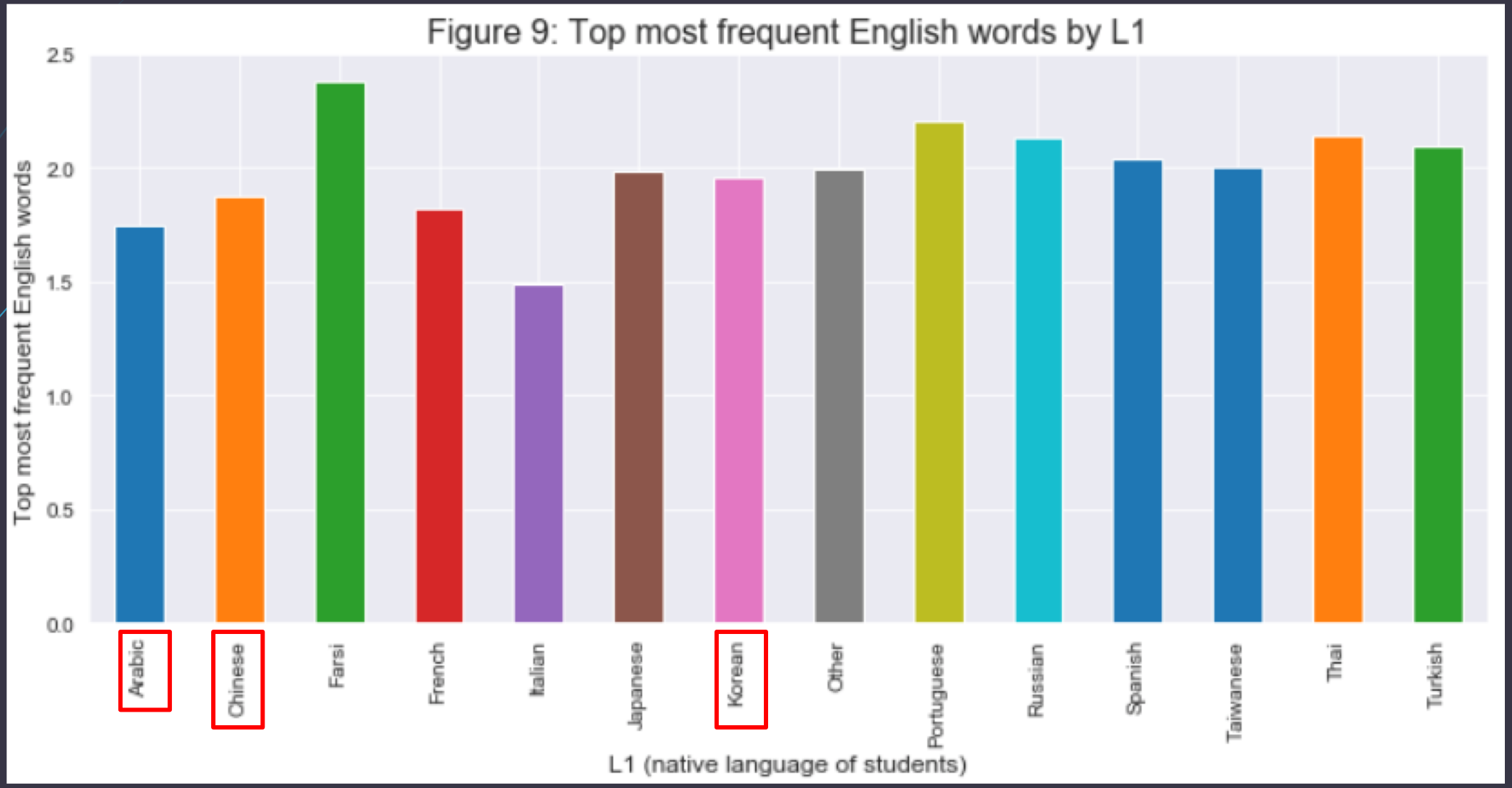
# 2. Syntactic complexity



# 3a. Vocabulary level - Average word length



# 3b. Top most frequent English words



# Check out the results on our dataset

- Narrow down the scope of the text to essays only.
- Select essays between 250 to 400 tokens. There are 22 essays left.

```
misspelling words:  've
corrected words:  eve

candidate words:  {'ive', 'eve', 'rve', 'ave', 've', 'ove'}
------------------------------
misspelling words:  'm
corrected words:  mm

candidate words:  {"c'm", 'sm', 'mm', 'lm', 'pm', 'cm', 'qm', 'bm', 'wm', 'vm', "m'", 'hm', 'x
m', 'am', 'em', 'om', 'dm', 'tm', 'rm', "'", 'fm', 'jm', 'km', 'im', 'nm', 'gm', 'm', 'um'}
------------------------------
misspelling words:  ''
corrected words:  d'

candidate words:  {"n'", "d'", "e'", "a'", "y'", "j'", "p'", "t'", "s'", "r'", "m'", "'", "f'"
, "c'", "o'", "i'", "q'"}
------------------------------
misspelling words:  hight
corrected words:  right

candidate words:  {'light', 'tight', 'wight', 'height', 'might', 'right', 'night', 'high-', 'h
ighs', 'sight', 'high', 'eight', 'fight', 'bight'}
```

```
misspelling words:  nowruz
corrected words:  nowruz

candidate words:  {'nowruz'}
--------------------------------
misspelling words:  quran
corrected words:  duran

candidate words:  {"qur'an", 'qumran', 'duran'}
--------------------------------
misspelling words:  baklava
corrected words:  baklava

candidate words:  {'baklava'}
--------------------------------
```

# Conclusion

- Students may have less error in writing material.
- The spell checker performance will be influenced by the dictionary and tokenized words.
- This model may be more suitable to the speaking data which are written as the texts.
- It may not be proper to replace the data we have now with the words that this spell checker model recommend.

# Thank you!

Reference:

- How to Write a Spelling Corrector (Peter Norvig)
  - http://norvig.com/spell-correct.html
- Bayes' Theorem (Na-Rae Han slides)
  - http://www.pitt.edu/~naraehan/ling1330/Lecture11.pdf

# Q&A

# Bayes' Theorem

- *P(A)*: the probability of A occurring
- *P(A | B)*: Conditional probability
  - the probability of A occurring, given that B has occurred
- *P(A, B)*: Joint probability
  - the probability of A occurring and B occurring
  - Same as P(B, A).
  - If A and B are independent events, same as P(A)*P(B).
  - If not, same as P(A | B)*P(B) and also P(B | A)*P(A).

# Bayes' Theorem

❶ $$P(B \mid A) = \frac{P(B, A)}{P(A)} = \frac{P(A \mid B) * P(B)}{P(A)}$$

- B: Pitt closing, A: snowing
- P(B | A): probability of Pitt closing, given snowy weather
- P(B, A): probability of Pitt closing and snowing

- 1 : the probability of Pitt closing given it's snowing is equal to the probability of Pitt closing and snowing, divided by the probability of snowing.