

START OF QUIZ

Student ID:

**89702757,MacFar-
lane,Jarrett**

Question 1

Topic: Lecture 4

Source: Lecture 4

In French, negation is often indicated by "ne ... pas" (ie, "je ne parle pas" - "I am not speaking"; "tu ne conduis pas" - "You are not driving", etc.). However, in speech, one of the two is often dropped: "je ne parle." or "tu conduis pas.". Using this information, how would you determine whether a corpus was composed of written or spoken French? You don't need to write the code, but explain the logic that you would use to come to this conclusion. (2)

Question 2

Topic: Lecture 3

Source: Lecture 3

Imagine that we have a parallel corpus (ie, a corpus containing sentences in two languages), and we want to extract a bilingual lexicon. What are some simple steps we could do to identify words that could be translations of each other? (2)

Question 3

Topic: Lecture 2

Source: Lecture 2

Would a language with with lots of inflection have a higher or lower TTR than one with little inflection? Briefly explain. (1)

Question 4

Topic: Lecture 3

Source: Lecture 3

How does "get" differ from a default dictionary (2 ways)? (1)

Question 5

Topic: Lecture 1

Source: Lecture 1

What are two ways to check if a word is all capital letters (neither should require more than one function call)? (1)

Question 6

Topic: Lecture 4

Source: Lecture 4

In class, we removed stopwords by using a lexicon. Can you think of another way that we could remove all closed class words? (1)

Question 7

Topic: Lecture 1

Source: Lecture 1

Vowels are often used as a proxy for syllables in words (it's not a perfect correspondence, but it's not bad). Write a function that counts the vowels in a word, without using a loop, using only the tools we went over in Lecture 1 (list comprehension counts as a loop). (2)

Question 8

Topic: Lecture 2

Source: Lecture 2

As we expand the size of our corpus, we increase the number of Hapaxes. Do you think this is also true of stopwords? Briefly explain. (1)

Question 9

Topic: Coding

Source: Coding

Grimm's law is a linguistic phenomenon that describes how sounds in language (mostly related to the Germanic languages like English, Dutch, German, Norwegian, Icelandic, etc.) changed over time (specifically from some progenitor thousands of years old - Germanic languages observed the change, while Romance languages did not). For example, the /p/ sound in Latin evolved into the /f/ sound in English across many words - compare "piscus" with "fish"; "pater" with "father"; "pedus" with "foot" (there are a couple other changes in there, too - see if you can spot them!). If you had a time machine, and could bring a computational toolkit to help Jacob Grimm formulate his law, what would you need, in terms of lexicons, keeping in mind that the /p/ -> /f/ change is only one of a handful of sound changes, and that the changes occurred over dozens of languages? Explain (with pseudocode, if necessary), how you would start to identify trends in the data (assuming that your computer still works in the 19th century)? (3)

END OF QUIZ