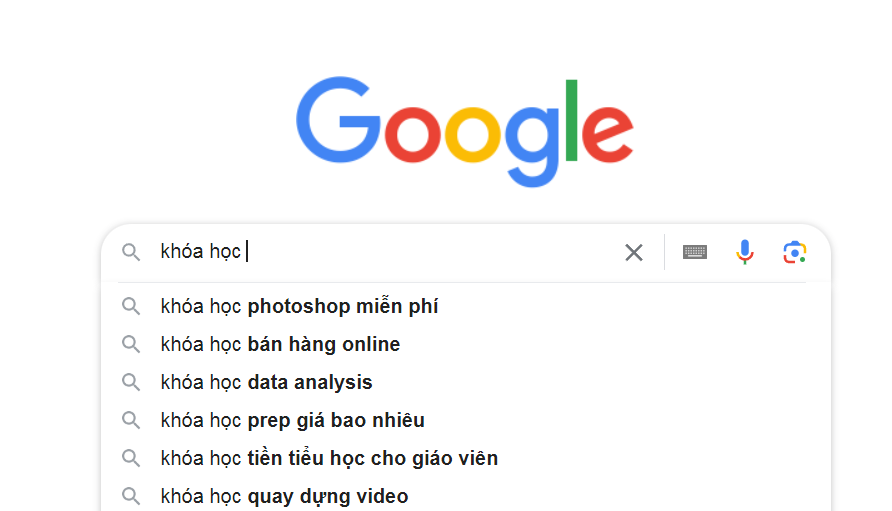
# Language Models: Auto-Complete

Auto-complete là một công nghệ trí tuệ nhân tạo tự động tự động gợi í từ tiếp theo khi người dùng dự định nhập một văn bản nào đó, một ví dụ cũng như ứng dụng phổ biến nhất của công nghệ này là gợi ý từ khóa tìm kiếm trong ô Search của Google, hoặc nhập liệu trên bàn phím điện thoại,..

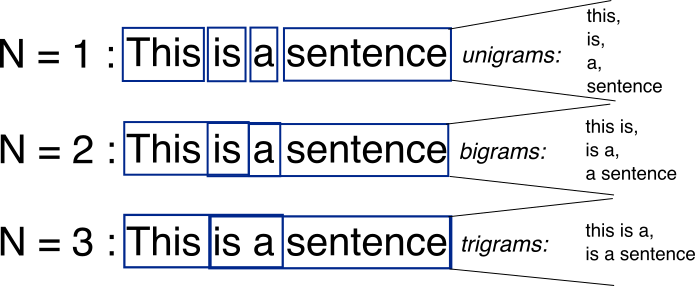


Có rất nhiều công có thể hiện đại hơn, nhưng ở đây tôi đề cập đến một công nghệ cổ điển, nền tảng của những công nghệ mới ngày nay. (được đề cập trong khóa học coursera: **Natural Language Processing**)

Công nghệ: N – Gram

Nguồn Coursera: <https://www.coursera.org/learn/probabilistic-models-in-nlp?specialization=natural-language-processing>

1 Mô hình N-gram:



N-gram là một chuỗi liên tiếp của n từ trong một đoạn văn bản. Ví dụ:

Câu: I love programming

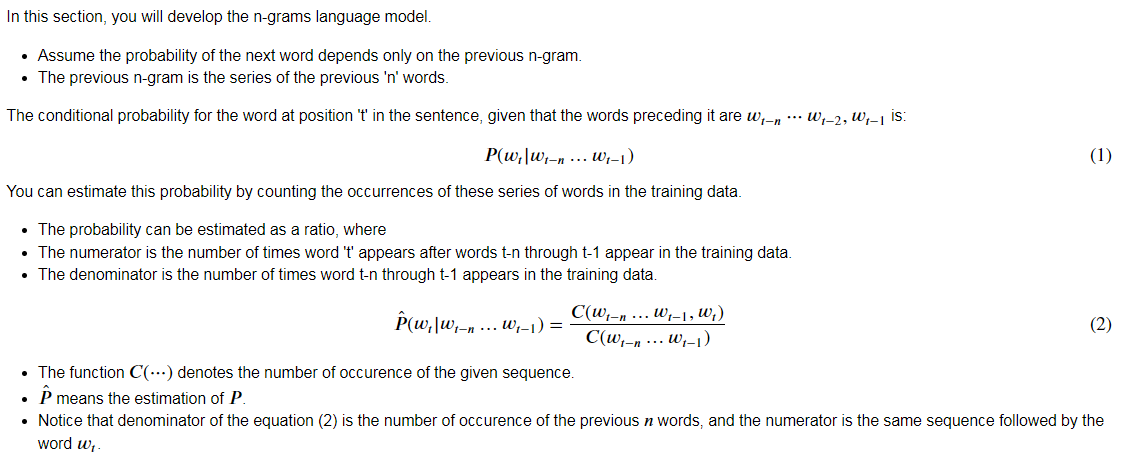
1-gram (unigram): "I", "love", "programming"

2-gram (bigram): "I love", "love programming"

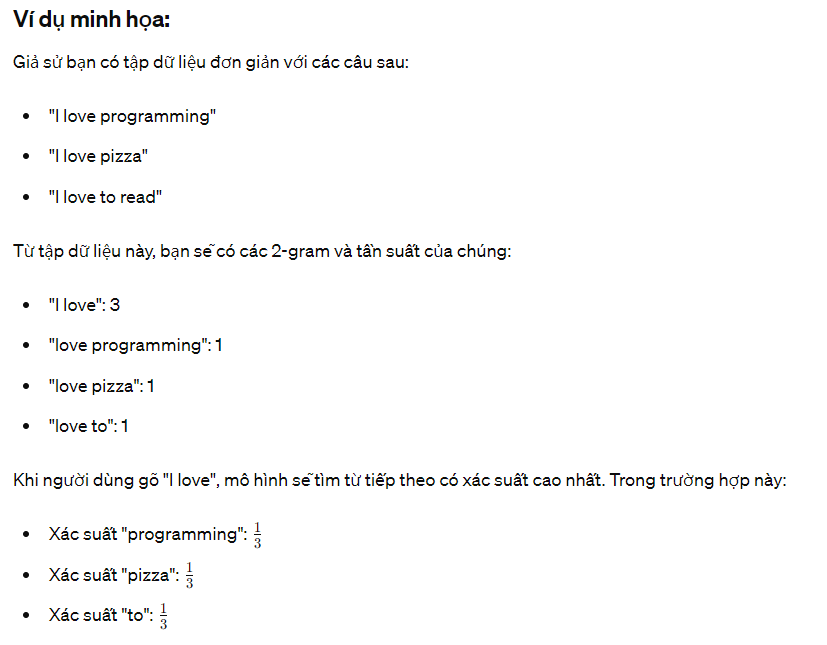
3-gram (trigram): "I love programming"

Sau khi có được bộ data N – gram (N: 1, 2, 3, 4,.. tùy vào cách thiết lập và mong muốn và thử nghiệm của người code). Chúng ta sẽ tính “xác suất” xuất hiện của từ “tiếp theo” với công thức:

# Coursera:



Công thức có vẻ nhìn rắc rối, nhưng tô có một ví dụ minh họa để dễ tiếp cận hơn:

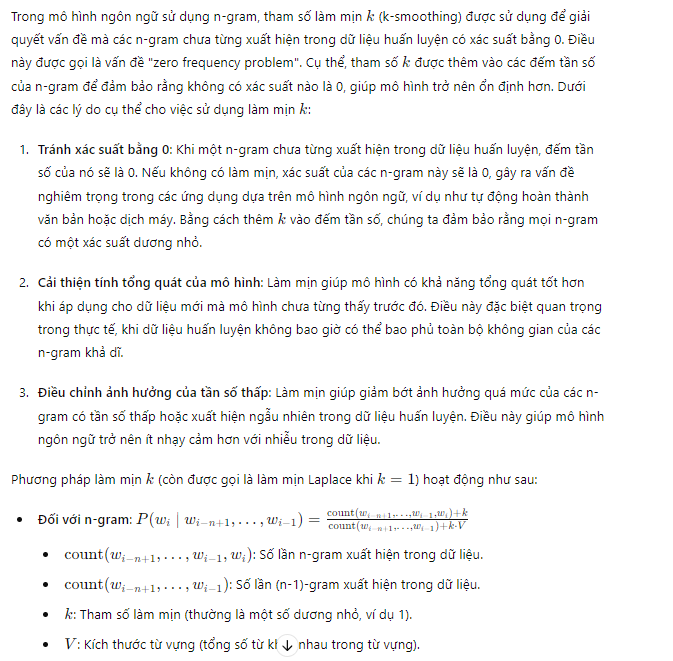


Xác suất hiện tại ở đây đều bằng nhau, nghĩa là bất kì từ nào trong: programming, pizza, hoặc to đều có thể là từ tiếp theo của cụm 2 - gram “I love”. Nếu trong data của bạn có bất kì một từ nào có xác suất cao nhất thì đó là lựa chọn tốt nhất phù hợp để trở thành từ tiếp theo của cụm “I love” (hệ thống gợi ý).

Nội dung trên là tổng quát về N-gram được ứng dụng trong Auto-complete.

Chi tiết về hệ thống, tôi sẽ trình bày trong code. Bên cạnh đó một minh họa có đầy đủ giao diện, và cách thức kết nối giữa UI và UX cơ bản để sử dụng Auto-complete.

Trong thực tế người ta sẽ dùng các biến thể khác của công thức tính xác suất ở trên, ở đây tôi sử dụng công thức Laplace với một tham số làm mịn k



<https://github.com/Hoanha2101>