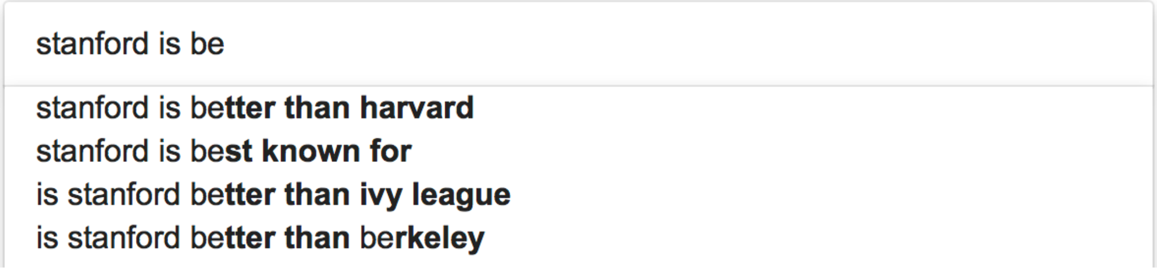
Homework 3
Language Models: Auto-complete

You will develop a prototype of an auto-complete system. Auto-complete system is something you may see every day

- When you google something, you often have suggestions to help you complete your search.
- When you are writing an email, you get suggestions telling you possible endings to your sentence.



stanford is be

stanford is **better than harvard**
stanford is **best known for**
is stanford **better than ivy league**
is stanford **better than** berkeley

A key building block for an auto-complete system is a language model. A language model assigns the probability to a sequence of words so that more "likely" sequences receive higher scores. For example,

> "I have a pen" is expected to have a higher probability than "I am a pen" since the first one seems to be a more natural sentence in the real world.

You can take advantage of this probability calculation to develop an auto-complete system. Suppose the user typed

> "I eat scrambled" Then you can find a word x such that "I eat scrambled x" receives the highest probability. If x = "eggs", the sentence would be "I eat scrambled eggs"

While a variety of language models have been developed, this assignment uses **N-grams**, a simple but powerful method for language modeling.

- N-grams are also used in machine translation and speech recognition.

Here are the steps for this homework:

1. Load and preprocess data
    - Load and tokenize data.
    - Split the sentences into train and test sets.
    - Replace words with a low frequency by an unknown marker <unk>.
2. Develop N-gram based language models
    - Compute the count of n-grams from a given data set.
    - Estimate the conditional probability of a next word with k-smoothing.
3. Evaluate the N-gram models by computing the perplexity score.
4. Use your own model to suggest an upcoming word given your sentence.