a) N-grams are contiguous sequences of n words from a given text or corpus. For example, "I love cats" can be broken down into unigrams (single words) such as "I," "love," and "cats," bigrams (two-word sequences) such as "I love" and "love cats," and trigrams (three-word sequences) such as "I love cats." N-grams can be used to build language models, which estimate the probability of a word given its preceding words. This is important in natural language processing tasks such as speech recognition, machine translation, and text generation. To build a language model using n-grams, we can count the frequency of each n-gram in a corpus and divide it by the frequency of its preceding n-1 gram. This gives us the probability of a word given its context. This can be extended to, literally, any N-gram sequence of words, and can include punctuation.

b) N-grams can be used in various applications, including spell checking, text classification, text generation, and language identification. They can also be used to generate auto-complete suggestions and to analyze the sentiment of a text based on word combinations.

c) To calculate the probabilities of unigrams and bigrams, we count the frequency of each word and bigram in the corpus, respectively, and divide by the total number of words or bigrams. However, this approach often results in zero probabilities for unseen n-grams, which can affect the accuracy of the language model as it's reflected as missing data.

d) The source text is crucial in building a language model, as the model is only as good as the corpus it is trained on. It is important to use a diverse and representative corpus to ensure that the language model can handle a wide range of inputs.

e) Smoothing is an important technique used to adjust the probability estimates for unseen n-grams. One simple approach is to add a small constant value to the numerator and denominator of the probability estimate. This is called Laplace smoothing and helps to avoid zero probabilities.

f) Language models can be used for text generation by sampling the probabilities of the next word given the previous words. However, this approach has limitations as it may result in grammatically incorrect, nonsensical, or downright strange sentences.

g) Language models can be evaluated by measuring their perplexity, which is the inverse of the geometric mean of the probabilities assigned by the model to each word in a test set. A lower perplexity indicates better performance. They can also be evaluated based on the sheer size of the data they're trained on, and on how "holistically" or "contextually" they predict text.

h) Google's n-gram viewer is a tool that allows users to explore the frequency of n-grams in a large corpus of text. An example of its use is to compare the usage of two different words over time, such as "climate change" and "global warming." The tool can also show the frequency of specific n-grams in different languages and genres. https://books.google.com/ngrams/graph?content=Brexit%2C+EU&year_start=1800&year_end=2019&corpus=en-2019&smoothing=3

You can also select specific corpus' to use to further narrow and predict a term.