**Person 2: N-gram Models and Their Limitations\***

**\*Topic: \*Classical Approaches: N-grams and Fixed Context Windows**

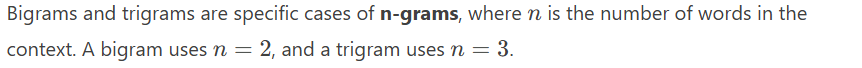
1. **What mathematical framework underpins bigram and trigram models?**

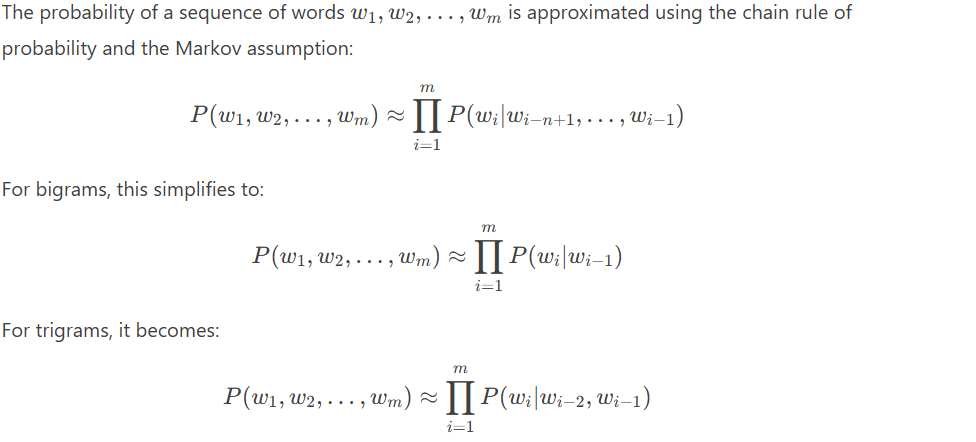
**Probability Theory:**

● Models such as bigrams and trigrams derive from two or more words appearing together and use conditional probability to assess how frequent this melding is occurring. To demonstrate, they would estimate the probability of a single word occurring given a single word or two consecutive words preceding it – bigram and trigram specifications, respectively.

**Markov Assumption:**

• These frameworks use the Markov assumption, which makes calculations easier by claiming that the chance of a word occurring relies on a certain set of context words that came before it.   
• Bigram models make the Markov assumption of stating that the chances of a word being said depends solely on the word said prior to it.  
  
• For trigram models, the assumption is made that the two previous words aid in increasing the probability of a word’s occurrence.

**N-grams**:



1. **What are the primary strengths of n-gram models in real-world applications (e.g., Google’s auto-complete)**

The n-gram models, especially the bigram and trigram models, have advantages which make them useful in a variety of domains such as Google’s auto-completion features, speech recognition, machine translation, and text generation. Following are the main advantages:

**a. Simplicity and Efficiency:**

• The implementation and computation of n-gram models is easy and efficient since everything is based on counting word sequences (n-grams) within a corpus. The implementation method is particularly straightforward as it revolves around estimating probabilities.   
• Their simplicity makes them appropriate for real-world scenarios, such as real-time autocomplete \ applications, in which minimal delays are very important.

**b. Contextual Prediction:**

* N-gram models capture local dependencies between words, which is useful for predicting the next word in a sequence. For example, in auto-complete, the model can suggest likely words based on the previous one or two words typed by the user.
* This contextual prediction is particularly effective for short-range dependencies, which are common in natural language.

**c. Robustness with Large Corpora:**

* When trained on large corpora, n-gram models can generalize well to a wide variety of language patterns and contexts. For example, Google’s auto-complete benefits from training on massive datasets, enabling it to handle diverse user inputs and languages.

**d. Interpretability:**

* N-gram models are interpretable because they rely on explicit counts of word sequences. This makes it easier to debug and understand why certain predictions are made, which is valuable for improving systems like auto-complete or spell-checking.

**Example: Google’s Auto-Complete**

In Google’s auto-complete, n-gram models are particularly effective because:

* They predict likely search queries based on partial input, leveraging the frequency of n-grams in past searches.
* They adapt to user behavior and trending topics by updating n-gram counts in real time.
* They provide fast and accurate suggestions, even for rare or long-tail queries, by combining n-grams with other techniques like personalization and ranking.

**Limitations (for Context):**

While n-gram models have many strengths, they also have limitations, such as:

* Difficulty capturing long-range dependencies (e.g., dependencies between words far apart in a sentence).
* Struggling with rare or unseen n-grams (though smoothing techniques help mitigate this).
* Lack of semantic understanding (e.g., they don’t understand the meaning of words, only their co-occurrence patterns).

1. **How do fixed context windows limit the accuracy of n-gram predictions?**

Fixed context windows in n-gram models, such as bigrams (context of 1 word) or trigrams (context of 2 words), limit the accuracy of predictions in several ways. These limitations arise because n-gram models rely on a fixed number of preceding words to predict the next word, which can fail to capture important linguistic phenomena. Here are the key ways fixed context windows limit accuracy:

**Inability to Capture Long-Range Dependencies:**

* Natural language often involves dependencies between words that are far apart in a sentence. For example:
  + **Pronoun Reference**: "The cat, which was sitting on the mat, **it** was sleeping."
  + **Verb Agreement**: "The books on the shelf **are** old."
* N-gram models with fixed context windows (e.g., bigrams or trigrams) cannot capture these long-range dependencies because they only consider a limited number of preceding words.