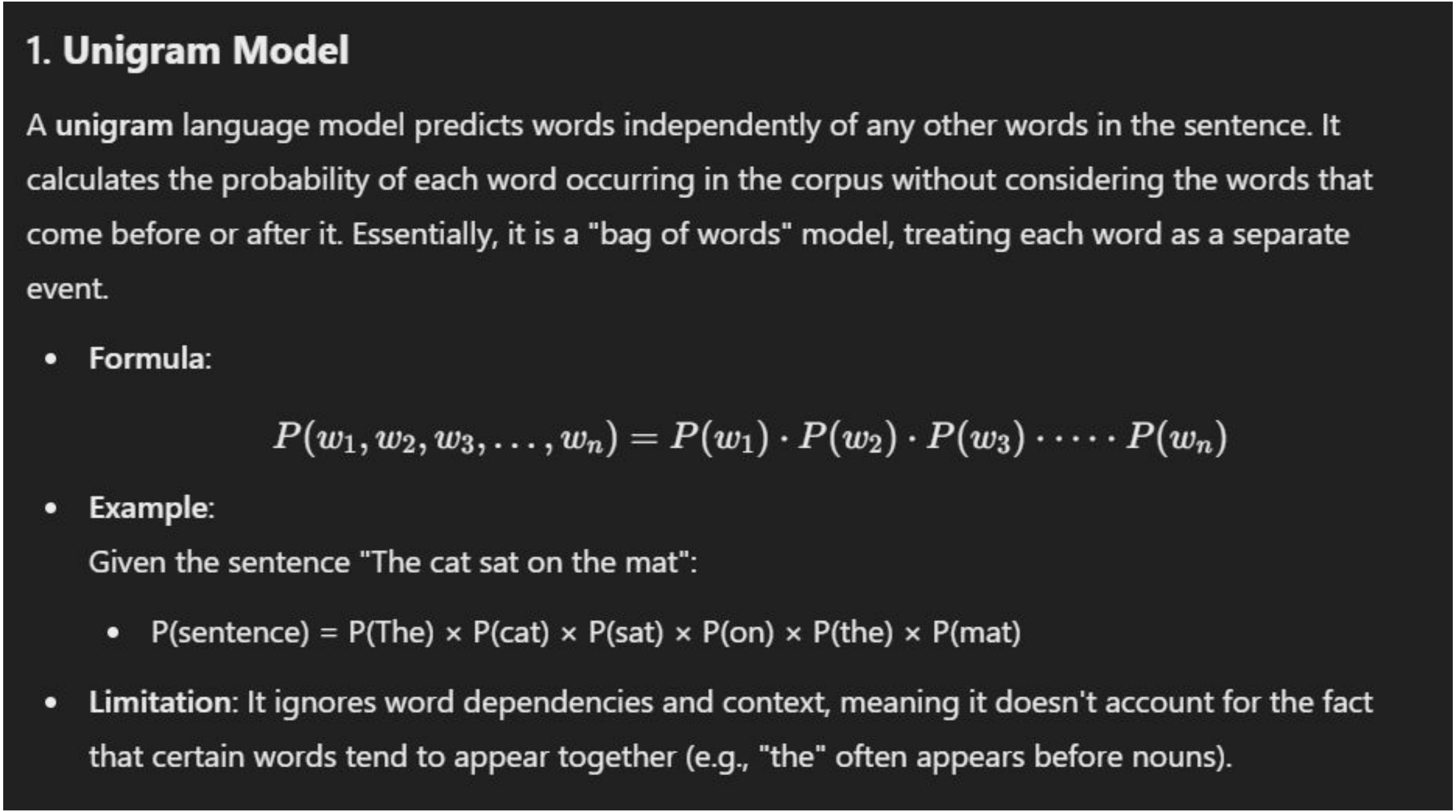
Language Model

Basic



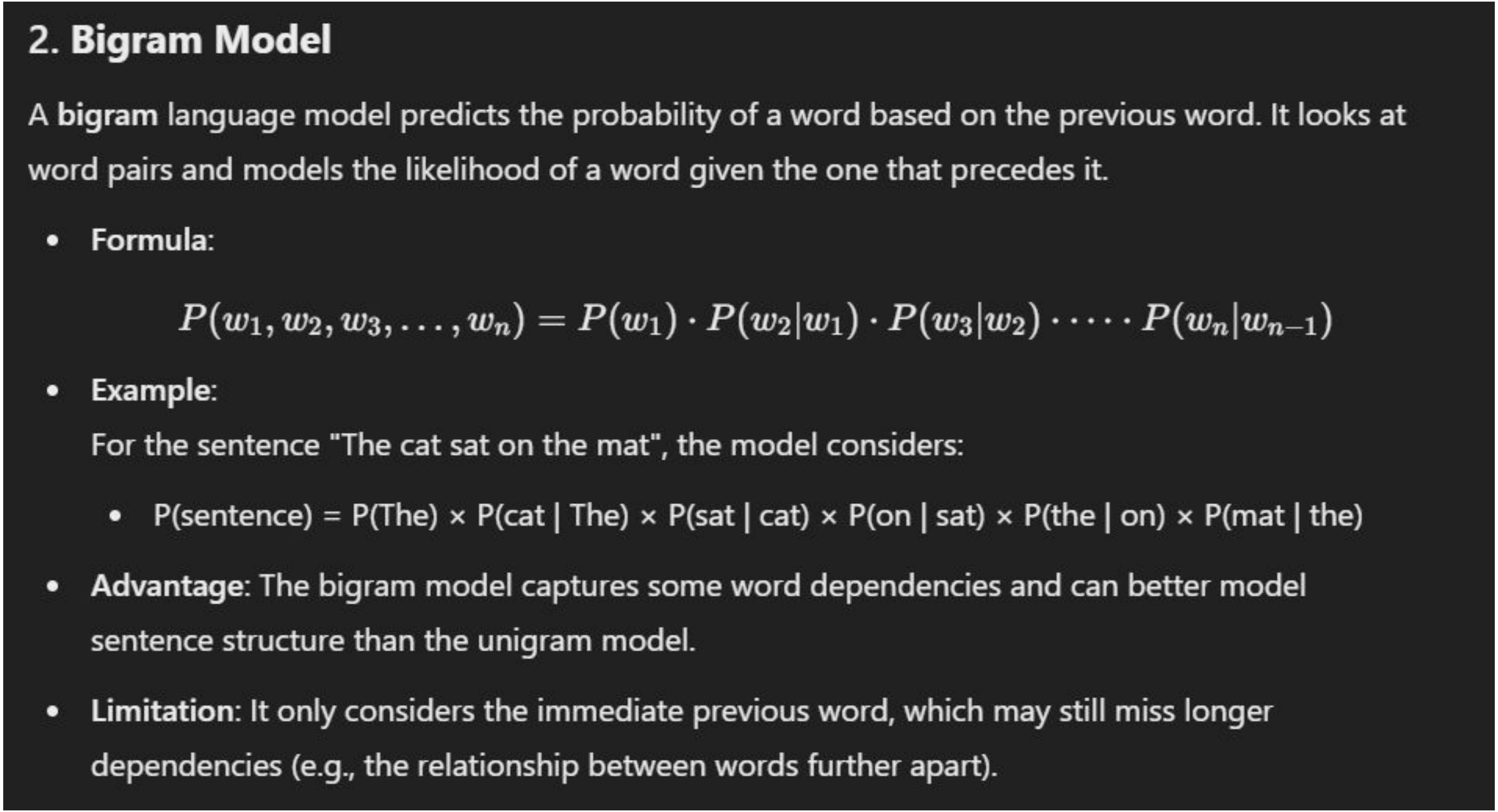
1. Unigram Model

A **unigram** language model predicts words independently of any other words in the sentence. It calculates the probability of each word occurring in the corpus without considering the words that come before or after it. Essentially, it is a "bag of words" model, treating each word as a separate event.

- **Formula**:

$$P(w_1, w_2, w_3, \ldots, w_n) = P(w_1) \cdot P(w_2) \cdot P(w_3) \cdot \cdots \cdot P(w_n)$$

- **Example**:
  Given the sentence "The cat sat on the mat":
  - P(sentence) = P(The) × P(cat) × P(sat) × P(on) × P(the) × P(mat)
- **Limitation**: It ignores word dependencies and context, meaning it doesn't account for the fact that certain words tend to appear together (e.g., "the" often appears before nouns).



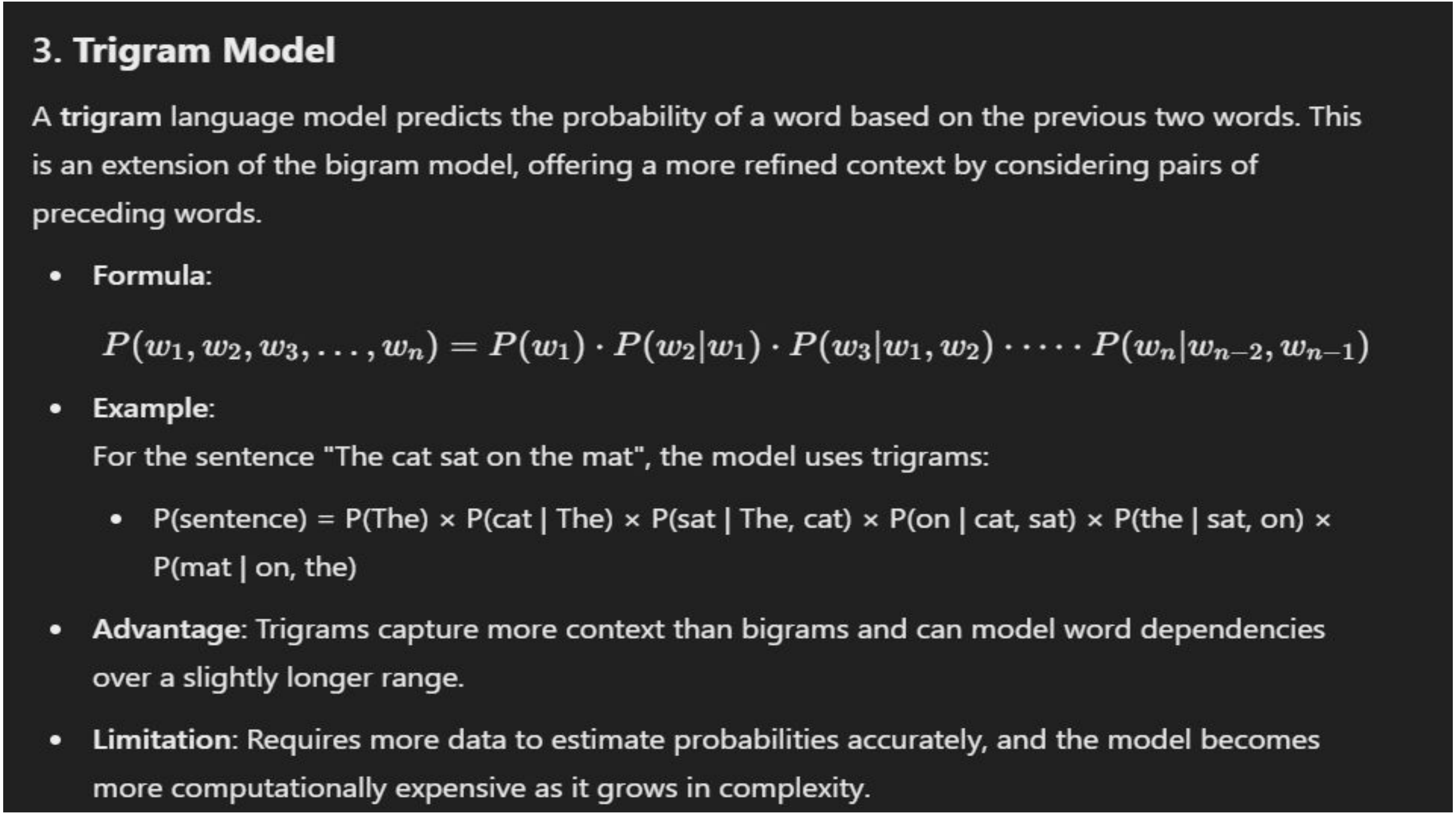
2. Bigram Model

A **bigram** language model predicts the probability of a word based on the previous word. It looks at word pairs and models the likelihood of a word given the one that precedes it.

- **Formula**:

$$P(w_1, w_2, w_3, \ldots, w_n) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_2) \cdot \cdots \cdot P(w_n|w_{n-1})$$

- **Example**:
  For the sentence "The cat sat on the mat", the model considers:
  - P(sentence) = P(The) × P(cat | The) × P(sat | cat) × P(on | sat) × P(the | on) × P(mat | the)
- **Advantage**: The bigram model captures some word dependencies and can better model sentence structure than the unigram model.
- **Limitation**: It only considers the immediate previous word, which may still miss longer dependencies (e.g., the relationship between words further apart).



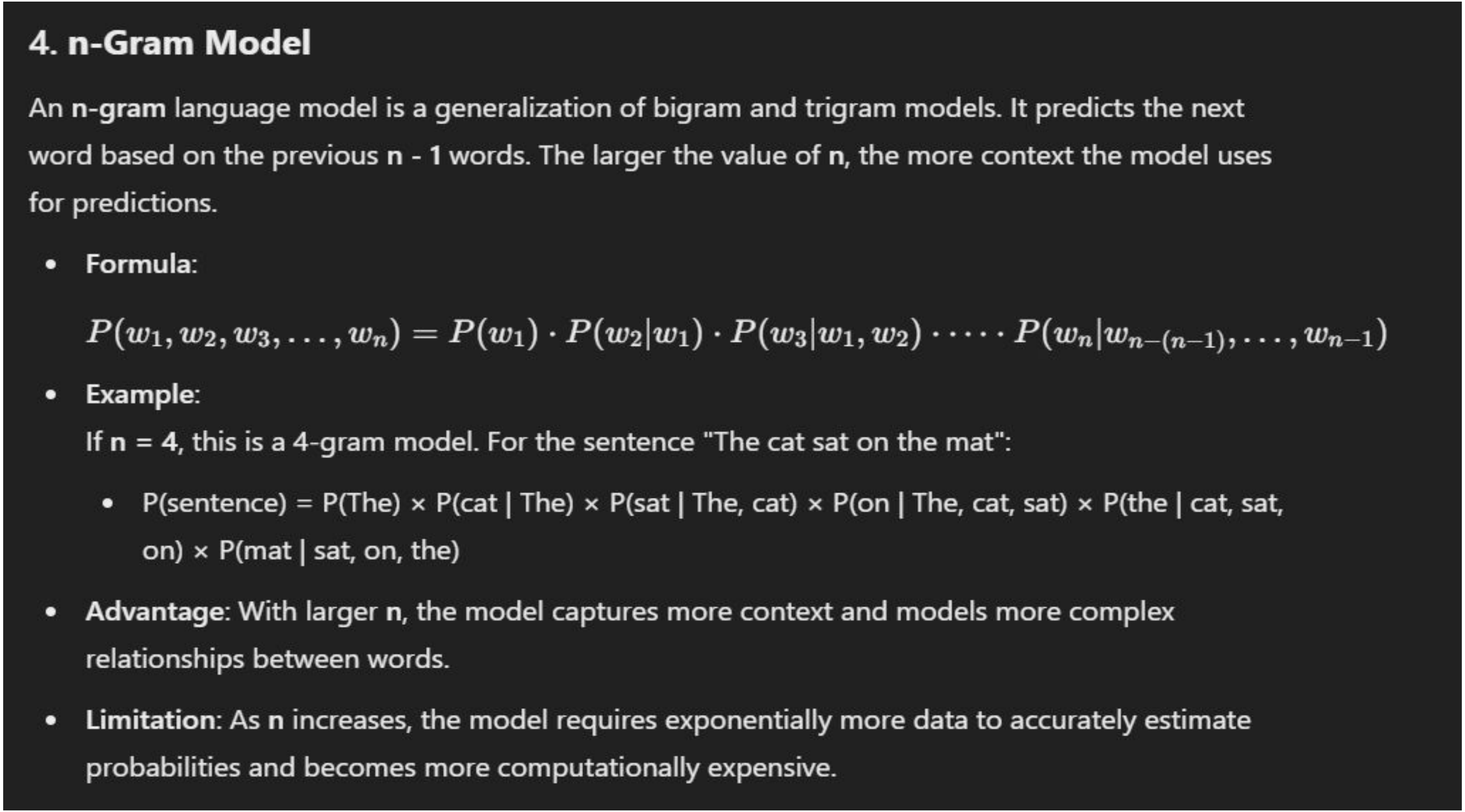
3. Trigram Model

A **trigram** language model predicts the probability of a word based on the previous two words. This is an extension of the bigram model, offering a more refined context by considering pairs of preceding words.

- **Formula**:

  $$P(w_1, w_2, w_3, \ldots, w_n) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \cdots \cdot P(w_n|w_{n-2}, w_{n-1})$$

- **Example**:
  For the sentence "The cat sat on the mat", the model uses trigrams:
  - P(sentence) = P(The) × P(cat | The) × P(sat | The, cat) × P(on | cat, sat) × P(the | sat, on) × P(mat | on, the)
- **Advantage**: Trigrams capture more context than bigrams and can model word dependencies over a slightly longer range.
- **Limitation**: Requires more data to estimate probabilities accurately, and the model becomes more computationally expensive as it grows in complexity.



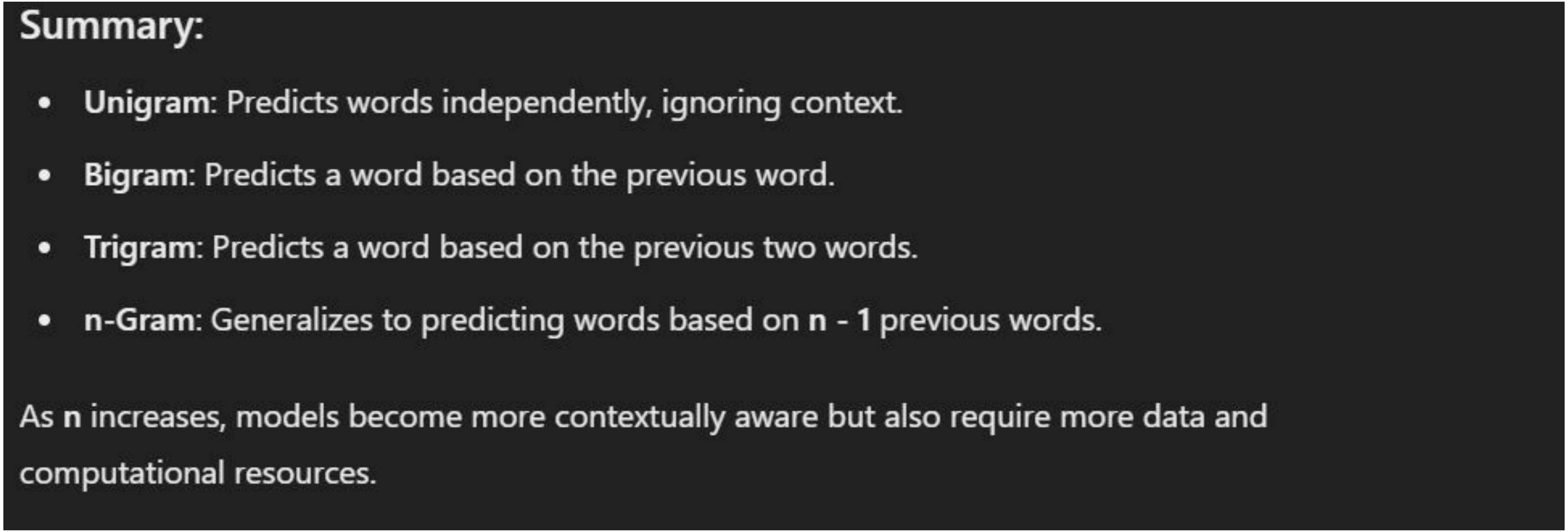
4. n-Gram Model

An **n-gram** language model is a generalization of bigram and trigram models. It predicts the next word based on the previous **n - 1** words. The larger the value of **n**, the more context the model uses for predictions.

- **Formula**:

  $$P(w_1, w_2, w_3, \ldots, w_n) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \cdots \cdot P(w_n|w_{n-(n-1)}, \ldots, w_{n-1})$$

- **Example**:
  If **n = 4**, this is a 4-gram model. For the sentence "The cat sat on the mat":
  - P(sentence) = P(The) × P(cat | The) × P(sat | The, cat) × P(on | The, cat, sat) × P(the | cat, sat, on) × P(mat | sat, on, the)
- **Advantage**: With larger **n**, the model captures more context and models more complex relationships between words.
- **Limitation**: As **n** increases, the model requires exponentially more data to accurately estimate probabilities and becomes more computationally expensive.



Summary:

- **Unigram**: Predicts words independently, ignoring context.
- **Bigram**: Predicts a word based on the previous word.
- **Trigram**: Predicts a word based on the previous two words.
- **n-Gram**: Generalizes to predicting words based on **n - 1** previous words.

As **n** increases, models become more contextually aware but also require more data and computational resources.