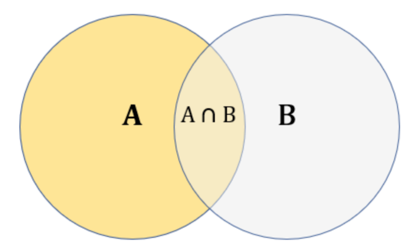
$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$



P(B|A): The probability of event B occurring given that event A has already occurred.
A: The condition (the event that has already happened).
B: The event we want to know the probability of.

Conditional Probability

1. Definition of Conditional Probability:

$P(B|A) = \frac{P(A \cap B)}{P(A)}$

* **Meaning:** The probability of event B occurring given that event A has already occurred is equal to the probability of both events A and B occurring divided by the probability of event A occurring.
* **When to Use:** Use this when you know the probability of both events A and B occurring, $P(A \cap B)$, and the probability of event A occurring, $P(A)$, and you want to find the probability of event B occurring given that event A has occurred, $P(B|A)$.
* **Important:** $P(A) > 0$ must be true. That is, the probability of event A occurring (the condition) must be greater than 0.

2. Expression of Joint Probability (Using Conditional Probability):

$P(A \cap B) = P(B|A) * P(A)$

* **Meaning:** The probability of both events A and B occurring is equal to the product of the probability of event A occurring and the conditional probability of event B occurring given that event A has occurred.
* **When to Use:** Use this when you know the probability of event A occurring, $P(A)$, and the probability of event B occurring given that event A has occurred, $P(B|A)$, and you want to find the probability of both events A and B occurring, $P(A \cap B)$.
* **Alternative Expression:** It can also be expressed as $P(A \cap B) = P(A|B) * P(B)$. In this case, you can use the probability of event B occurring and the conditional probability of event A occurring given that event B has occurred to find the probability of both events A and B occurring.

Key Summary:

* $P(B|A)$: Probability of B occurring given that A has occurred (Conditional Probability)
* $P(A \cap B)$: Probability of both A and B occurring (Joint Probability)
* $P(A)$: Probability of A occurring

These two formulas are very important in probability theory and are used to solve various problems. You need to choose the appropriate formula depending on what values you know and what values you need to find.

Sentence Probability

What is Sentence Probability?

Sentence probability is a measure of how naturally a particular sentence appears in a natural language. In other words, given a sentence, it's the probability, expressed as a number, of how likely that sentence is to actually be used. For example, the sentence "I ate rice" is much more natural and likely to be used than the sentence "I ate the refrigerator." Therefore, the sentence "I ate rice" would have a higher probability than the sentence "I ate the refrigerator."

Why Calculate Sentence Probability?

Sentence probability has various applications in the field of Natural Language Processing (NLP). For instance, in machine translation systems, it's used to evaluate the naturalness of translated sentences and select the most appropriate translation result. Also, in text generation models, it's used to train the model to generate sentences that are grammatically correct and semantically coherent by maximizing sentence probability. Sentence probability plays an important role in various NLP tasks such as spam mail filtering, speech recognition, and text summarization.

How is Sentence Probability Calculated?

The most common method for calculating sentence probability is to use conditional probability. The words that make up a sentence don't appear independently; they appear influenced by the preceding words. That is, the probability of a particular word appearing is conditioned on the words that appeared before it. Therefore, the probability of the entire sentence can be expressed as the product of the probabilities of each word conditioned on the preceding words.

Expressed as a formula: The probability of a sentence $w_1, w_2, ..., w_n$ is calculated as follows:

$P(w_1, w_2, ..., w_n) = P(w_1) \times P(w_2 | w_1) \times P(w_3 | w_1, w_2) \times ... \times P(w_n | w_1, w_2, ..., w_{n-1})$

Here, $P(w_i | w_1, w_2, ..., w_{i-1})$ represents the conditional probability that the word w_i will appear after the words $w_1, w_2, ..., w_{i-1}$ have appeared.

For example, the probability of the sentence "I ate rice" can be calculated as follows:

$P(\text{I ate rice}) = P(\text{I}) \times P(\text{ate} | \text{I}) \times P(\text{rice} | \text{I ate})$

Challenges in Calculating Actual Sentence Probability

In practice, calculating sentence probability has some difficulties. First, estimating the probabilities for all possible word combinations is computationally very expensive. Second, it's difficult to estimate the probabilities for word combinations that don't appear in the training data (rare word combinations). To solve these problems, various smoothing techniques and language modeling techniques are used.

Conclusion

Sentence probability is a core concept in the field of Natural Language Processing and is used in various applications. Understanding the basic idea of calculating sentence probability using conditional probability is very important for developing and understanding Natural Language Processing systems.

Sparcity Problem

What is the Sparsity Problem?
The sparsity problem refers to the phenomenon where certain events or combinations appear very rarely in a dataset. In other words, a dataset is considered "sparse" when most of its values are 0 or empty.

Sparsity Problem in Natural Language Processing:
In NLP, the sparsity problem mainly occurs when doing language modeling. A language model is a model that learns the probability distribution of word sequences from text data. For example, it learns what word is likely to come after "I love to eat."
When training such a language model, word sequences (n-grams) that have **not appeared in the training data can occur**. For example, if the sentence "I enjoy eating broccoli" was not in the training data, the probability of the sequence "I enjoy eating broccoli" would be 0. However, "I enjoy eating broccoli" is actually a perfectly possible sentence.

The problem that arises **due to word sequences that have not appeared in the training data is called the sparsity problem.** No matter how large the training data is, it cannot contain all possible word sequences in the world, so the sparsity problem always occurs in language modeling.

Impact of the Sparsity Problem:
The sparsity problem is one of the main causes of performance degradation in language models.

- **Underfitting:** The model fails to predict properly for word sequences that are not in the training data.
- **Reduced Generalization Ability:** The model is overly fitted to the training data, and its prediction performance on new data decreases.

Solutions to the Sparsity Problem:
Various methods are used to solve the sparsity problem.

1. **Smoothing:** A method of assigning non-zero probabilities to word sequences that have not appeared in the training data. Examples include Laplace smoothing, Add-k smoothing, Good-Turing smoothing, and Kneser-Ney smoothing.
2. **Back-off Models:** A method of using shorter n-grams when longer n-grams cannot be used. For example, when a trigram (3-gram) cannot be used, a bigram (2-gram) or unigram (1-gram) is used.
3. **Interpolation:** A method of using a weighted average of different n-gram models.
4. **Word Embeddings:** A method of representing words in a low-dimensional vector space. Words with similar meanings are located close to each other in the vector space, so the model can generalize to some extent even for words that were not in the training data. Examples include Word2Vec, GloVe, and FastText.
5. **Neural Language Models:** Neural network models such as Recurrent Neural Networks (RNNs) and Transformers are more robust to the sparsity problem than traditional n-gram models.