Leek's Algorithm for Word Sense Disambiguation (WSD)

Leek's Algorithm is a supervised machine learning approach for Word Sense Disambiguation (WSD) that relies on the construction of a decision list. Below is a detailed explanation of its steps:

Steps in Leek's Algorithm

1. Data Preparation

- A labeled corpus is used, where each instance of the target word is annotated with its correct sense.
- Features such as contextual words, collocations, and parts of speech are extracted from the surrounding text.

2. Feature Extraction

The algorithm extracts features such as:

- Words in a fixed window around the target word (e.g., n words to the left and right).
- Collocations or fixed word pairs (e.g., "bank of" or "deposit in").
- Syntactic features such as part-of-speech tags or dependency relations.

3. Rule Generation

For each feature, the probability of a sense given the feature is computed as:

$$P(\text{sense} \mid \text{feature}) = \frac{\text{Count}(\text{sense}, \text{feature})}{\text{Count}(\text{feature})}$$

Where:

- $\text{Count}(\text{sense}, \text{feature})$ is the number of times the feature co-occurs with a particular sense.
- $\text{Count}(\text{feature})$ is the total number of times the feature occurs.

4. Rule Weighting

The weight of a rule is determined using the log-likelihood ratio:

$$\text{Weight}(\text{feature}) = \log \frac{P(\text{sense}_1 \mid \text{feature})}{P(\text{sense}_2 \mid \text{feature})}$$

Where:

- sense_1 and sense_2 are two possible senses of the target word.
- A higher weight indicates that the feature strongly predicts one sense over another.

5. Decision List Construction

- Features are ranked by their weights.
- A threshold is applied to retain only those features with weights above a specific value (e.g., Weight > 0.97).
- The resulting ranked list of features forms the decision list.

6. Application of the Decision List

- For a new instance of the target word, the features are extracted from the context.
- The decision list is applied to predict the most probable sense based on the highest-weighted matching rule.

Example

Word: *bank*

- Training Data:
  - Sentence: “He sat by the **bank** of the river.”
  - Features: “river”, “bank of”
  - Sense: *river bank*
- Decision List:

    - Rule: If the word "river" is in the context, predict *river bank*, Weight = 2.0.
    - Rule: If the word "money" is in the context, predict *financial bank*, Weight = 3.0.

- Testing:

    - Sentence: "He sat by the bank of the river."
    - Feature: "river"
    - Predicted Sense: *river bank*

Advantages

- Simple and interpretable.
- Works well when strong, clear features are available.

Disadvantages

- Requires labeled training data.
- May overfit to specific features.
- Struggles with highly ambiguous contexts.