WSD Methods Summary

Word Sense Disambiguation (WSD) – Project SummaryOverview:
Word Sense Disambiguation (WSD) refers to identifying the correct meaning (sense) of a word in a given context. For instance, the word "bank" can mean a financial institution or the side of a river. WSD is essential for natural language processing (NLP) applications like machine translation, question answering, and information retrieval.

1. Baseline: Word Embeddings (Word2Vec / GloVe)

- **Method:**
    - Use a Word2Vec model which will use windows over the sentence and will adjust its word embeddings accordingly
    - Measure cosine similarity between ambiguous word ("bank") and context words.
- **Limitation:**
    - Produces a single static vector per word.
    - Cannot distinguish between different senses (e.g., financial vs river bank).

2. BERT-Based Contextual Embeddings (I have used BERT-model directly)

- **Method:**
    - Use BERT to extract contextual vectors for the word in different sentences.
    - Measure cosine similarity between contexts.
- **Limitation:**
    - Similarity values remain positive even for unrelated senses, showing lack of complete disambiguation.

3. Mathematical Enhancement: Sense-Specific Projection Matrices

- **Method:**
    - Learn a unique projection matrix for each sense of a word.
    - Multiply the word vector by this matrix to get a sense-specific vector.
- **Analogy (For easy to understand consider a prism):**
    - Like a prism splitting white light into multiple colors, this method separates word meanings into subspaces.
- **Application:**

- Use dot products to distinguish mixed-context usage (e.g., both financial and river) based on projections.

4. WordNet-Based Disambiguation

- **Method:**
    - Leverage WordNet synsets (synsets are basically the different sense embedding of a word) to define distinct word senses.
    - Extract example sentences and compute sense-specific embeddings.
    - Measure similarity using cosine or Euclidean distance e.g L1 norm , L2norm etc.
- **Benefit:**
    - Human-readable definitions and clear contextual separation of senses.

---

5. Neurosymbolic Dartboard Embedding (Analogy can be a 2d dart used in games)

- **Method:**
    - Important note point is normalize each sense vector first for equal spreadness..
    - Model each sense as a spherical region in vector space.
    - A contextualized word vector is treated as a "dart" landing in the correct sphere.
- **Advantages:**
    - Outperforms traditional deep learning methods.
    - Breaks the 80% F1-score ceiling of conventional WSD models.
- **Limitation:**
    - Uses isotropic (equal spread) spheres, which is unrealistic for real language data.

---

6. Modified Dartboard with Ellipsoids

- **Method:**
    - Generalize spherical regions to ellipsoids with different spread and orientation.

- Use a covariance matrix to shape each ellipsoid.

- **Improvement:**
  - Provides a more accurate and probabilistic representation of overlapping senses.

---

7. Contextual Box Embedding (ProtoBox)

- **Method:**
  - Represent each sense as a high-dimensional box (hyperrectangle).
  - Use BERT to extract contextual vectors and generate "instance" boxes.
  - Compare with "prototype" sense boxes using Intersection-over-Union (IoU).
- **Interpretability:**
  - Supports hierarchy (containment), disjointness, and overlap.
  - Enhances transparency and semantic reasoning.

---

8. GlossBERT and Sentence-Level Embeddings

- **Method:**
  - Combine BERT's [CLS] token (sentence-level) with target word embedding.
  - Prepend WordNet gloss (definition) to the sentence.
  - Compare sentence with gloss vectors using cosine similarity.
- **Benefit:**
  - Helps disambiguate cases with ambiguous or minimal context.

---

9. Knowledge Graph Enhanced Neural WSD (EWISER : Enhanced WSD Integrated Synsets Embedding and regulations)

- **Method:**
  - Use WordNet or BabelNet (I have used WordNet) as a graph of senses (nodes) and relations (edges).
  - Apply Graph Neural Networks (GNNs) to propagate and refine sense embeddings.

    - Score each candidate sense using contextual vectors and refined graph embeddings.
- **Advantage:**
    - Combines symbolic knowledge with neural models.
    - Offers rich, explainable, context-conditioned predictions.

Mathematical Techniques Used:

- Cosine Similarity and Euclidean Distance
- Dot Product for Context Matching
- Intersection over Union (IoU)
- Cross-Entropy Loss for Training
- Graph Convolution Layers (GNN)
- Mean Pooling over BERT outputs

Experimental Highlights:

- ProtoBox excels in interpretability and performance for hierarchical reasoning.
- Modified Neurosymbolic Dartboard provides probabilistic, ellipsoidal sense modeling.
- EWISER integrates symbolic graph knowledge for superior sense prediction accuracy.