INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Introduction to NLP

Word Sense Disambiguation

Team : InsightMinds

1. K Sai Kiran (2023201067)
2. T V S Vishnu Vardhan (2023202021)
3. L Sai Chaitanya Reddy (2023202023)

Introduction

- WSD is a critical task in Natural Language Processing (NLP) aimed at determining the correct sense of a word in context.
- Essential for various NLP applications such as machine translation, information retrieval, and text summarization.
- **Challenges** arise due to polysemy, where words have multiple meanings depending on context.
- **Scope of the Project:** Explore and evaluate machine learning models for WSD, develop a model from scratch.
- **Dataset Selection:** Utilize SemCor and SemEval datasets known for contextual diversity.
- **Systematic Evaluation:** Assess model performance using standard metrics like accuracy for comparative analysis.

Dataset Overview

- Total Sentences - 37,767 sentences
- Semantically Annotated: Sentences are labeled with their intended meanings.
- Diverse Categories: Covers various genres like news, fiction, etc., ensuring broad applicability.
- Data Structure: Two files—semcor.data for sentence and sense ID information, semcor.gold.key for WordNet sense IDs.
- Foundation for WSD: Crucial resource for training and evaluating Word Sense Disambiguation (WSD) models.
- Enables Research: Facilitates exploration of contextual disambiguation methods and model development in NLP.

Knowledge Base - WordNet

- Overview: Lexical database of English, organizing words into synsets (sets of synonyms).
- Structure: Arranged as a graph, with synsets as nodes and lexical-semantic relations as edges.
- Synsets: Groups of contextual synonyms representing different senses of a word.
- Lexical-Semantic Relations: Include hypernymy (is-a) and meronymy (part of), forming a hierarchical meaning structure

Key Points:

- Sense Inventory: De facto resource for WSD in English.
- Hierarchical Structure: Facilitates understanding of word meanings and semantic relationships.
- Practical Application: Essential for both defining sense labels and exploring semantic connections in NLP projects.

Approaches - 1 Nearest Sense

Training Phase

- Each sentence in the SemCor labeled dataset is passed through a contextual embedding model (e.g., BERT).
- Pooling: Summing vector representations from the last four BERT layers.
- For each sense s of any word in the corpus, average the contextual representations vi of each token representing that sense to produce a contextual sense embedding V(s).

$$\mathbf{v}_s = \frac{1}{n}\sum_i \mathbf{v}_i \qquad \forall \mathbf{v}_i \in \text{tokens}(s)$$

Approaches

Testing Phase:

Given a token of a target word t in context:

- Compute its contextual embedding t.
- Choose its nearest neighbor sense from the training set based on cosine similarity, selecting the sense whose embedding has the highest cosine similarity with t.

$$\text{sense}(t) = \underset{s \in \text{senses}(t)}{\text{argmax}} \ \text{cosine}(\mathbf{t}, \mathbf{v}_s)$$

1 Nearest Sense - Results

Parameters Used
Embedding Dimension : 300
No. of Epochs : 5

Evaluation Datasets	Accuracies
SenseEval 3	52%
SenseEval 2	54%
SemEval 2007	45%
SemEval 2013	52%
Concatenation of all above	53%

Approaches - Context2Vec

Architecture Overview:

- Bidirectional LSTM (BiLSTM) networks process sentence words from left to right and right to left separately for each word in the sentence.
- LSTM outputs from both directions are concatenated to capture comprehensive sentential context.
- Multi-layer perceptron (MLP) captures complex dependencies between the two sides of the context.
- Joint sentential context around the target word and the target word itself are embedded into the same low-dimensional space.

Context2Vec - Training Objective

Bidirectional LSTM (BiLSTM) Context Representation:

- Utilizes lLS (left-to-right) and rLS (right-to-left) LSTM networks to capture sentence-level context. Shallow bidirectional LSTM context representation for target word wi then concatenates distinct left-to-right/right-to-left word embeddings of sentence words, excluding the target word itself.

$$\mathrm{biLS}(w_{1:n}, i) = \mathrm{lLS}(l_{1:i-1}) \oplus \mathrm{rLS}(r_{n:i+1})$$

Non-linear Transformation:

- Applies a non-linear function on the concatenation of left and right context representations, utilizes Multi Layer Perceptron (MLP) with Rectified Linear Unit (ReLU) activation function for transformation.

$$\mathrm{MLP}(x) = L_2(\mathrm{ReLU}(L_1(x)))$$

Context2Vec - Training Objective

Context2vec Representation

- Defines context2vec's representation of the sentential context c, Represents the entire joint sentential context around the target word.

$$\vec{c} = \text{MLP}(\text{biLS}(w_{1:n}, i)).$$

Learning Objective

- Learn target word and context repr using word2vec negative sampling objective function.
- Minimizes the difference between the dot product of target word and context representations and the sigmoid function value.
- Utilizes negative sampling for efficient training, with negative samples sampled from the training corpus.

$$S = \sum_{t,c} \left(\log \sigma(\vec{t} \cdot \vec{c}) + \sum_{i=1}^{k} \log \sigma(-\vec{t_i} \cdot \vec{c}) \right)$$

Sense-Specific Representation Generation

- Obtain word representations for each word in the training dataset using the trained model.
- Average the representations based on the word senses, creating sense-specific representations.
- Store these sense-specific representations for future use during testing.

$$\mathbf{v}_s = \frac{1}{n}\sum_i \mathbf{v}_i \qquad \forall \mathbf{v}_i \in \text{tokens}(s)$$

Context2Vec – Testing

- When testing, compute the contextual representation of the test word.
- Calculate the similarity scores between the test word's representation and the stored sense-specific representations.
- Predict the word sense by selecting the sense with the highest similarity score as the most likely interpretation for the test word.

$$\mathrm{sense}(t) = \operatorname*{argmax}_{s \in \mathrm{senses}(t)} \ \mathrm{cosine}(\mathbf{t}, \mathbf{v}_s)$$

Context2Vec - Results

Parameters Used

No. of negative Samples : 5

Embedding Dimension : 300

No. of Epochs : 5

Evaluation Datasets	Accuracies
SenseEval 3	58%
SenseEval 2	62%
SemEval 2007	52%
SemEval 2013	67%

GlossBERT

"

- **"GlossBERT** extends BERT by not only considering the contextual cues but also integrating the definitions (glosses) of words directly into the model, enhancing the model's ability to discern the correct sense based on both usage and definition.
- **Benefit:** "This approach allows GlossBERT to achieve a deeper understanding of context and semantic meanings, outperforming traditional models especially in nuanced contexts.

Negative Sampling Concept

"

- **Negative sampling:** Negative Sampling is a training strategy used to teach models to distinguish not just the correct answer but also to identify what are incorrect answers."
- **Application to WSD**: "In the context of GlossBERT, negative sampling involves presenting both correct sense and multiple incorrect senses during training, enhancing the model's ability to accurately perform sense disambiguation."
- **Input Preparation:** "For each target word in a context, GlossBERT takes the sentence as input along with multiple glosses of the word. The model is then trained to associate the sentence with the correct gloss."

GlossBert – Results

Evaluation Datasets	Accuracies
SenseEval 3	53%
SenseEval 2	51%
SemEval 2007	44%
SemEval 2013	73%

Thank You