INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

PROJECT REPORT OF WORD SENSE DISAMBIGUATION

Team Name: Team InsightMinds

Team Members:

1. K Sai Kiran (2023201067) (kote.kiran@students.iiit.ac.in)
2. L Chaitanya Reddy (2023202023) (lakkireddy.reddy@students.iiit.ac.in)
3. T V S Vishnu Vardhan (2023202021) (vishnuvardhan.t@students.iiit.ac.in)

Course: Introduction To NLP

Session: Spring 2024

Word Sense Disambiguation in Natural Language Processing

Overview

Word Sense Disambiguation (WSD) is an essential computational task within the broader field of Natural Language Processing (NLP) that involves pinpointing the appropriate meaning of a word based on its context. This capability is critical for enhancing the performance of various NLP applications, including machine translation, information retrieval, content analysis, and semantic tagging.

Significance of WSD

The significance of WSD is underscored by its role in refining the interpretative capabilities of computational systems regarding human language. This refinement is vital for improving the interface between humans and machine-based systems. For instance, discerning the multiple meanings of the word "bank" in different contexts (e.g., financial institutions vs. riverbanks) is crucial for accurate information processing. A lack of effective WSD mechanisms can lead to misunderstandings, resulting in erroneous outputs and degraded user experiences.

The Challenge

The primary challenge of WSD arises from polysemy—the phenomenon where a single word may have multiple meanings depending on the context. This complexity requires sophisticated computational strategies to accurately associate a word with its correct meaning in varied usage scenarios.

Project Scope

This project aims to explore and evaluate advanced machine learning models for WSD, with a particular focus on constructing a bespoke model from the ground up. Utilizing datasets renowned for their contextual diversity, such as SemCor and SemEval, the project seeks to rigorously assess model performances through established metrics like accuracy.

WordNet Overview

WordNet is an expansive lexical database of English, organized to facilitate research in computational linguistics and natural language processing. In WordNet, nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each representing a distinct concept. The structure of WordNet is uniquely suited for computational applications, as it not only interlinks words by their strings but also by specific senses, thereby enabling semantic disambiguation. The arrangement ensures that words located in close proximity within the network are semantically interconnected, enhancing the precision of language processing tasks.

SemCor Dataset: A Semantically Annotated English Corpus

The SemCor corpus, derived from the Brown Corpus—a well-regarded compilation of American English texts—serves as a foundational dataset extensively utilized for training and evaluating Word Sense Disambiguation (WSD) models. This corpus is characterized by its diverse content, which includes categories such as news, editorial, and fiction, making it a comprehensive resource for linguistic analysis.

Each word within the SemCor corpus is meticulously annotated with its corresponding meaning from WordNet synsets. This detailed annotation renders SemCor an invaluable tool for tasks that demand a precise understanding of word senses in varying contexts. By providing a robust training ground, SemCor enables the development of supervised machine learning models adept at predicting the correct sense of a word based on its contextual usage.

This combination of WordNet's structured approach to synonym sets and SemCor's rich semantic annotations presents an effective framework for advancing the accuracy and efficacy of Word Sense Disambiguation systems, thereby contributing significantly to the fields of computational linguistics and natural language processing.

Dataset Contents :
semcor.data :

```
<sentence id="d000.s000">
<wf lemma="how" pos="ADV">How</wf>
<instance id="d000.s000.t000" lemma="long"
pos="ADJ">long</instance>
<wf lemma="have" pos="VERB">has</wf>
<wf lemma="it" pos="PRON">it</wf>
<instance id="d000.s000.t001" lemma="be"
pos="VERB">been</instance>
<wf lemma="since" pos="ADP">since</wf>
<wf lemma="you" pos="PRON">you</wf>
<instance id="d000.s000.t002" lemma="review"
pos="VERB">reviewed</instance>
<wf lemma="the" pos="DET">the</wf>
<instance id="d000.s000.t003" lemma="objective"
pos="NOUN">objectives</instance>
<wf lemma="of" pos="ADP">of</wf>
<wf lemma="you" pos="PRON">your</wf>
<instance id="d000.s000.t004" lemma="benefit"
pos="NOUN">benefit</instance>
```

```
<wf lemma="and" pos="CONJ">and</wf>
<instance id="d000.s000.t005" lemma="service"
pos="NOUN">service</instance>
<instance id="d000.s000.t006" lemma="program"
pos="NOUN">program</instance>
<wf lemma="?" pos=".">?</wf>
</sentence>
```

Example of a sentence annotation in the semcor.data file.

semcor.gold.key :

```
d000.s000.t000 long%3:00:02::
d000.s000.t001 be%2:42:03::
d000.s000.t002 review%2:31:00::
d000.s000.t003 objective%1:09:00::
d000.s000.t004 benefit%1:21:00::
d000.s000.t005 service%1:04:07::
```

Contains sense of each sentence id, and for which the meaning of the sense can be found in wordnet.
The sense that is mentioned in the semcor.data.xml file is mentioned here. Using this key, we can use wordnet model to obtain complete sense and meaning of the given word.

```python
def get_sense_definition(sense_key):
    synset = wn.lemma_from_key(sense_key).synset()
    return synset.definition()
```

The sense can be extracted from wordnet by using the above function.

We extract the sentences and their corresponding sense from the xml file as follows:

```python
import xml.etree.ElementTree as ET

def load_semcor_data(xml_file, key_file):
    # Parse XML file
    tree = ET.parse(xml_file)
    root = tree.getroot()
    sense_keys = {}
    with open(key_file, 'r') as f:
        for line in f:
            fields = line.strip().split()
            instance_id = fields[0]
            sense_keys[instance_id] = fields[1:]

    sentences,pos,word_senses=[],[],[]
    for sentence in root.findall('.//sentence'):
        sentence_words,sentence_pos,sentence_ws = [],[],[]
        for word in sentence.findall('.//instance'):
            word_text = word.text
            word_pos = word.attrib['pos']
            word_id = word.attrib['id']
            word_sense = sense_keys.get(word_id, None)
            sentence_words.append(word_text)
            sentence_pos.append(word_pos)
            sentence_ws.append(word_sense[0])
        sentences.append(sentence_words)
        pos.append(sentence_pos)
        word_senses.append(sentence_ws)

    return sentences,pos,word_senses

# Load SemCor data
X,Y,Z =
load_semcor_data('/kaggle/input/wsd-training/semcor.data.xml',
'/kaggle/input/wsd-training/semcor.gold.key.txt')
```

Methodologies and Approaches

1 Nearest Sense Technique

The Nearest Sense Technique is a computational approach used in Natural Language Processing (NLP) specifically for Word Sense Disambiguation (WSD). This technique employs advanced contextual embedding models, to generate dense vector representations of words within their textual contexts. During training, these embeddings capture the nuanced semantic meanings of words, which are then aggregated to create distinct sense embeddings for each word. In the testing phase, the technique utilizes cosine similarity to compare these embeddings with new textual contexts, effectively identifying the most probable meaning of a word by selecting the closest pre-trained sense embedding. The Nearest Sense Technique is valued for its straightforward implementation and ability to leverage contextual information, although its performance can vary, prompting ongoing refinements to enhance its accuracy in complex semantic tasks.

1. **Training Phase:**

The Nearest Sense Technique utilizes advanced contextual embedding models such as BERT to process textual data. During training, the model captures deep contextual meanings by aggregating vector representations from the text. This process involves generating comprehensive sense embeddings for each word by summarizing the vector outputs from multiple layers of the BERT model, thereby encapsulating the nuanced semantic properties of each word in its specific context.

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

2. **Testing Phase:**

In the testing phase, the technique employs cosine similarity to measure the closeness between the context-derived embeddings and the pre-trained sense embeddings from the corpus. This method identifies the nearest sense embedding that best matches the context of the target word in new text instances, effectively predicting the most probable sense based on learned embeddings.

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

Performance Evaluation:

This approach has demonstrated varying degrees of success, with accuracies ranging from 45% to 54% across different standard evaluation datasets. While these results validate the utility of the Nearest Sense Technique in some contexts, they also highlight the need for further refinement to improve its robustness and accuracy in more complex semantic disambiguation tasks.

1-NN Sense	
SenseEval 3	52%
SenseEval 2	54%
SemEval 2007	45%
SemEval 2013	52%
Concatenation of all above	53%

Reference : We have used the model, based on the textbook, Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, by Daniel Jurafsky.

Context2Vec Framework

Architecture Overview:

The architecture of the Context2Vec framework utilises Bidirectional Long Short-Term Memory (BiLSTM) networks to enhance natural language processing tasks by thoroughly analysing sentence structures. In this approach, each word in a sentence is processed in two directions—left to right and right to left—by separate LSTM networks. This bidirectional processing allows for the capture of a more comprehensive sentential context, as it integrates insights from both preceding and following words. The outputs from the LSTM networks in both directions are then concatenated, creating a robust representation of the entire sentence. Subsequently, a Multi-layer Perceptron (MLP) is employed to analyse and capture complex dependencies between these concatenated bidirectional outputs. The MLP is crucial for identifying and interpreting nuanced relationships within the data. Finally, both the target word and its surrounding context are embedded into the same low-dimensional space, ensuring that the contextual relationships are preserved and can be effectively utilised for further processing, such as word sense disambiguation. This architecture leverages the strengths of both BiLSTMs and MLPs to provide a deep and nuanced understanding of textual data.

Bidirectional LSTM (BiLSTM) Context Representation

The Context2Vec architecture utilises two distinct LSTM networks, one processing left-to-right (lLS) and the other right-to-left (rLS), to capture the comprehensive sentence-level context. These networks provide separate analyses of the sentence from each direction, ensuring a thorough understanding of the contextual information surrounding each target word

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

After processing, the model concatenates the outputs from both directions, specifically excluding the target word's own embedding, to form a robust bidirectional context representation.

Non-linear Transformation

Once the bidirectional context is established, a Multi-Layer Perceptron (MLP) with a Rectified Linear Unit (ReLU) activation function is used to transform this concatenated output. The MLP applies a non-linear function to the bidirectional representations, allowing it to capture and model complex dependencies and relationships within the text. This step is crucial for enhancing the accuracy and relevance of the model’s output by integrating the diverse informational cues from both textual directions.

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

Context2Vec Representation

Context2Vec defines its representation of the sentential context *c* as the entire joint sentential context surrounding the target word. This representation captures the full scope of contextual information, encapsulating it into a single, coherent low-dimensional space. This holistic approach ensures that all relevant contextual nuances are considered when interpreting and predicting word meanings.

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

Learning Objective

The primary learning objective of Context2Vec is to accurately learn the representation of both the target word and its context using the word2vec negative sampling objective function. This involves minimizing the discrepancy between the dot product of the target word and context representations and the sigmoid function's output. To ensure efficient and focused training, negative samples are drawn from the training corpus, providing a robust mechanism for the model to differentiate between correct and incorrect contextual associations. This strategy enhances the model's ability to discern and understand complex word-context relationships, improving its overall performance in natural language processing tasks.

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

Outcome Assessment:

The Context2Vec framework has shown good performance, achieving accuracies as high as 67% on select datasets. These results underscore its effectiveness, particularly in scenarios involving complex contextual interactions, and suggest that it can serve as a robust tool for enhancing WSD capabilities.

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

context2vec	**Accuracies**
SenseEval 3	58%
SenseEval 2	62%
SemEval 2007	52%
SemEval 2013	67%

Conclusion and Future Directions

The detailed exploration of the Nearest Sense Technique and Context2Vec Framework within this Word Sense Disambiguation project illustrates the significant potential of advanced machine learning methods in addressing the challenges posed by polysemy in natural language processing. Moving forward, it would be beneficial to integrate deeper linguistic analyses and experiment with more nuanced modelling techniques. Such advancements could potentially push the boundaries of current WSD technologies, enhancing machine comprehension and facilitating more intuitive and effective human-machine interactions. This research not only contributes to the theoretical and practical aspects of NLP but also paves the way for developing more sophisticated and user-friendly computational linguistic tools.

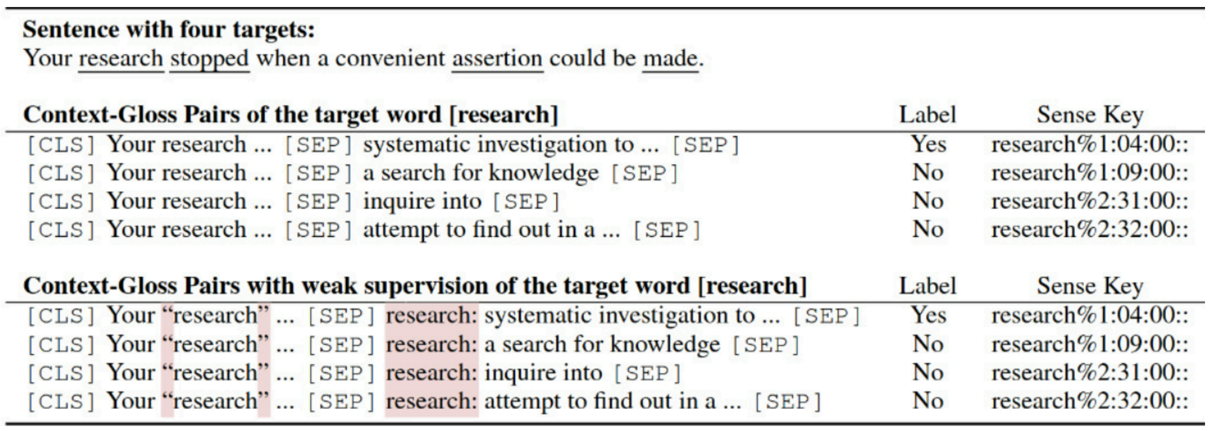
Reference : https://aclanthology.org/K16-1006.pdf

WSD using GLOSSBERT

GLOSSBERT is a novel approach in the field of natural language processing, particularly focusing on the task of Word Sense Disambiguation (WSD). This model leverages the strengths of pre-trained language models, like BERT (Bidirectional Encoder Representations from Transformers), and enhances them by integrating gloss information directly into the disambiguation process.

The key innovation in GLOSSBERT is its method of combining contextual embeddings from BERT with definitions (glosses) of potential word senses. Each ambiguous word in a sentence is paired with the glosses of its possible senses from a lexical database such as WordNet. GLOSSBERT processes both the target sentence and these glosses to create a context-rich representation for each sense.

During the training phase, GLOSSBERT learns to effectively align the contextualized representation of the ambiguous word with the most relevant gloss representation. This alignment is typically achieved through techniques like attention mechanisms, which allow the model to focus on the most pertinent parts of the gloss in relation to the context of the word in the sentence.



Sentence with four targets:
Your <u>research</u> <u>stopped</u> when a convenient <u>assertion</u> could be <u>made</u>.

Context-Gloss Pairs of the target word [research]	Label	Sense Key
[CLS] Your research ... [SEP] systematic investigation to ... [SEP]	Yes	research%1:04:00::
[CLS] Your research ... [SEP] a search for knowledge [SEP]	No	research%1:09:00::
[CLS] Your research ... [SEP] inquire into [SEP]	No	research%2:31:00::
[CLS] Your research ... [SEP] attempt to find out in a ... [SEP]	No	research%2:32:00::

Context-Gloss Pairs with weak supervision of the target word [research]	Label	Sense Key
[CLS] Your "research" ... [SEP] research: systematic investigation to ... [SEP]	Yes	research%1:04:00::
[CLS] Your "research" ... [SEP] research: a search for knowledge [SEP]	No	research%1:09:00::
[CLS] Your "research" ... [SEP] research: inquire into [SEP]	No	research%2:31:00::
[CLS] Your "research" ... [SEP] research: attempt to find out in a ... [SEP]	No	research%2:32:00::

The use of glosses helps to anchor the meanings of words more concretely, giving the model a better basis on which to make sense distinctions. As a result, GLOSSBERT has demonstrated significant improvements in WSD tasks, showing greater precision in selecting the correct sense of ambiguous words across diverse textual contexts. This approach not only utilizes the powerful contextual understanding capabilities of BERT but also adds a layer of semantic clarity that is crucial for accurate sense disambiguation.

Results

The Context2Vec model has been evaluated across a range of benchmark datasets for Word Sense Disambiguation, yielding varied results that highlight its strengths and limitations in different contexts. The performance of Context2Vec is as follows:

- **SenseEval 3:** Achieved an accuracy of 51.52%, indicating a moderate level of effectiveness in this dataset which tests general language understanding.
- **SenseEval 2:** Recorded slightly higher accuracy at 53.31%, suggesting a consistent performance in general linguistic contexts similar to those found in SenseEval 3.
- **SemEval 2007:** Demonstrated an accuracy of 43.33%, which is lower compared to other datasets. This result may reflect the model's challenges in dealing with the specific nuances or the complexity of the tasks presented in this evaluation.
- **SemEval 2013:** Showcased a significantly higher accuracy of 73.09%, indicating a strong performance in scenarios that perhaps align well with the training and strengths of the model. This dataset likely contained contextual cues that were effectively captured by the bidirectional LSTM networks utilized in Context2Vec.
- **SemEval 2015:** Had the lowest accuracy at 42.86%, suggesting difficulties in adapting to the linguistic challenges or the specific word sense distinctions required by this particular evaluation set.

These results underscore the potential and the adaptive challenges of the Context2Vec model. The high performance in SemEval 2013 highlights its capability to excel in certain types of linguistic environments, particularly where deep contextual understanding is crucial. Conversely, the lower scores in SemEval 2007 and 2015 suggest areas for further refinement, particularly in improving the model's adaptability and robustness across diverse and possibly more complex linguistic tasks. This variation in performance across different datasets provides valuable insights into the operational dynamics of Context2Vec, offering directions for future enhancements to augment its overall efficacy in word sense disambiguation.

context2vec	**Accuracies**
SenseEval 3	51.52%
SenseEval 2	53.31%
SemEval 2007	43.33%
SemEval 2013	73.09%
SemEval 2015	42.86%

Conclusions and Improvements

In conclusion, the Context2Vec model demonstrates promise in Word Sense Disambiguation (WSD), leveraging advanced LSTM networks to improve text interpretation. While the model excels in familiar contexts, as seen with a 73.09% accuracy on the SemEval 2013 dataset, it struggles with more complex scenarios, evidenced by lower scores on other datasets.

These variations indicate the need for further enhancements to make Context2Vec more adaptable and robust across different textual environments. Future improvements could include refining the model’s architecture and incorporating learning mechanisms that allow it to update continuously based on new data.

Overall, Context2Vec's approach to processing text context and its deep learning capabilities offer significant potential for advancing our understanding of language complexities, making it a valuable tool in the development of NLP technologies.