Veermata Jijabai Technological Institute, Mumbai 400019

Experiment No.: 06

Aim : Develop a Knowledge-based approach for Disambiguating the meaning of an input word from a given sentence

Name: Kiran K Patil

Enrolment No.: 211070904

Branch: Computer Engineering

Batch: D

Theory:

Disambiguating the meaning of a word in a sentence involves determining the correct sense or interpretation of the word based on its context. A knowledge-based approach typically relies on external knowledge sources, such as dictionaries, thesauruses, or domain-specific ontologies, to infer the most likely meaning.

Word Sense Disambiguation (WSD) is a crucial task in natural language processing that involves determining the correct meaning of a word within a given context. A knowledge-based approach for WSD relies on leveraging external knowledge sources, such as lexical databases or ontologies, to infer the most appropriate sense of a word in a particular context. The approach typically involves the following key components:

Here's a general outline of a knowledge-based approach for disambiguating the meaning of a word in a sentence:

1. **Preprocessing:**
    - The initial step involves breaking down the input sentence into its constituent words, a process known as tokenization.
    - To ensure a standardized representation of words, the subsequent lemmatization or stemming transforms words into their base forms. This step aids in reducing inflected words to their root form, facilitating clearer analysis.

2. **Context Window:**
    - Following preprocessing, a context window is defined around the target word within the sentence. This context window encompasses both the immediate neighbors of the target word and a broader context, allowing for a more comprehensive understanding of the word's meaning within its linguistic environment.

3. **Knowledge Base:**
    - The disambiguation process leverages external knowledge sources, which can include dictionaries, thesauruses, and specialized ontologies.
    - Word embeddings or vectors are employed to encapsulate the semantic relationships between words, providing the algorithm with a nuanced understanding of word meanings.

4. **Feature Extraction:**
    - Features are extracted from both the context window and the external knowledge base to enrich the disambiguation process.
    - These features may include word embeddings, which capture the semantic context of words, part-of-speech tags, which denote the grammatical category of each word, syntactic dependencies, and semantic relationships between words.

5. **Classifier:**
   - A machine learning classifier is employed to discern the correct sense of the target word within the given context.
   - Various classifiers, such as Support Vector Machines (SVM), Random Forests, or Neural Networks, can be trained using annotated datasets, where each word is labeled with its correct sense.
   - Pre-trained models, particularly those capturing word embeddings like Word2Vec or GloVe, may also be integrated to enhance the model's understanding of word contexts.

6. **Integration of Word Embeddings:**
   - The integration of pre-trained word embeddings adds an additional layer of semantic richness to the model.
   - These embeddings represent words in a continuous vector space, providing a more nuanced and contextually aware representation of words, aiding in the disambiguation process.

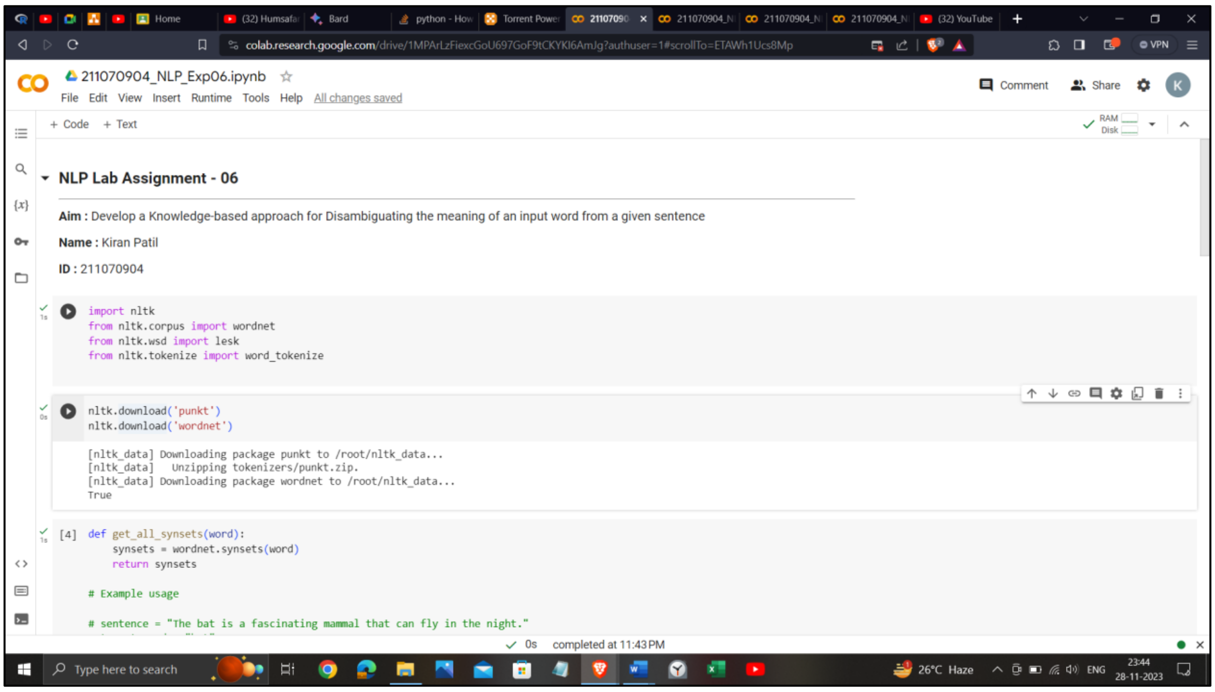
7. **Domain-specific Knowledge:**
   - The model's accuracy can be further refined by incorporating domain-specific knowledge if relevant to the application.
   - This may involve integrating specialized ontologies or dictionaries tailored to the specific subject matter, enhancing the model's ability to discern context-specific meanings.

8. **Sense Ranking:**
   - In instances where multiple senses are identified, a ranking mechanism is employed to determine the likelihood of each sense.
   - This ranking may utilize confidence scores generated by the classifier or other pertinent metrics to prioritize the most probable sense.

9. **Evaluation and Iteration:**
   - The efficacy of the disambiguation system is evaluated using benchmark datasets that contain labeled instances of correct senses.
   - The approach undergoes iterative refinement based on evaluation results and real-world performance, ensuring continuous improvement.

10. **User Feedback Loop:**
    - In interactive applications, a user feedback loop is established to incorporate corrections and insights from users.
    - This feedback loop contributes to ongoing model enhancement, allowing the system to adapt and improve its disambiguation capabilities over time.

Implementation:



211070904_NLP_Exp06.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text

NLP Lab Assignment - 06

Aim : Develop a Knowledge-based approach for Disambiguating the meaning of an input word from a given sentence

Name : Kiran Patil

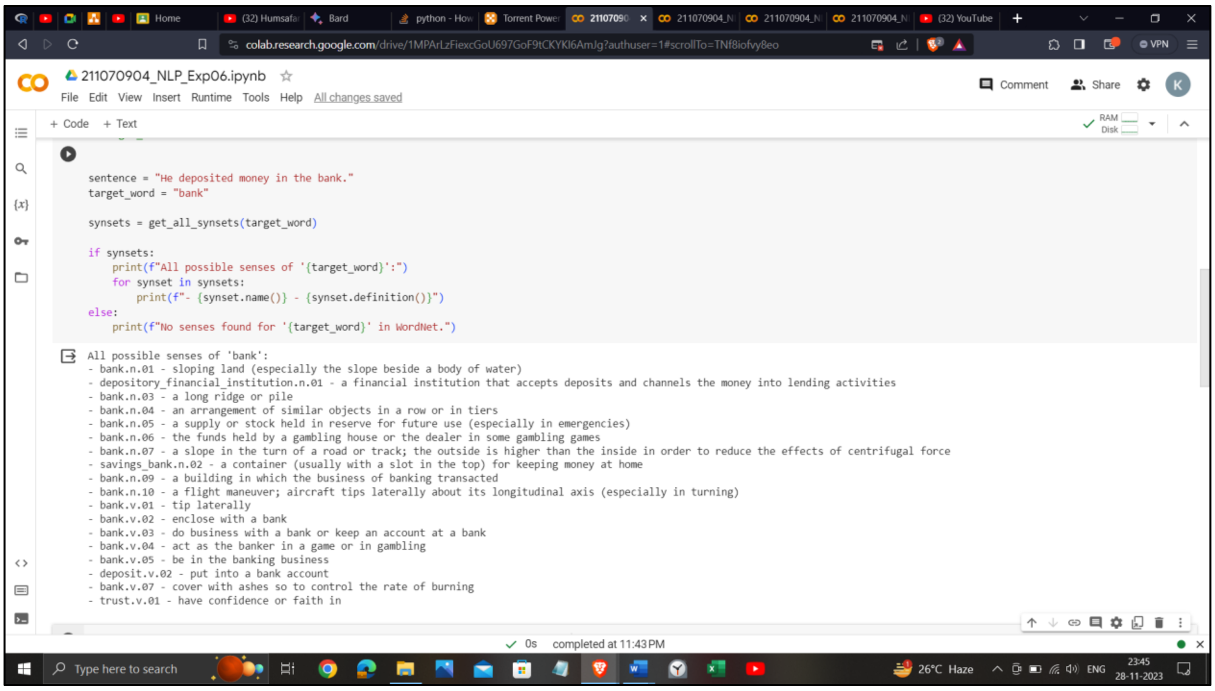
ID : 211070904

```python
import nltk
from nltk.corpus import wordnet
from nltk.wsd import lesk
from nltk.tokenize import word_tokenize
```

```python
nltk.download('punkt')
nltk.download('wordnet')
```

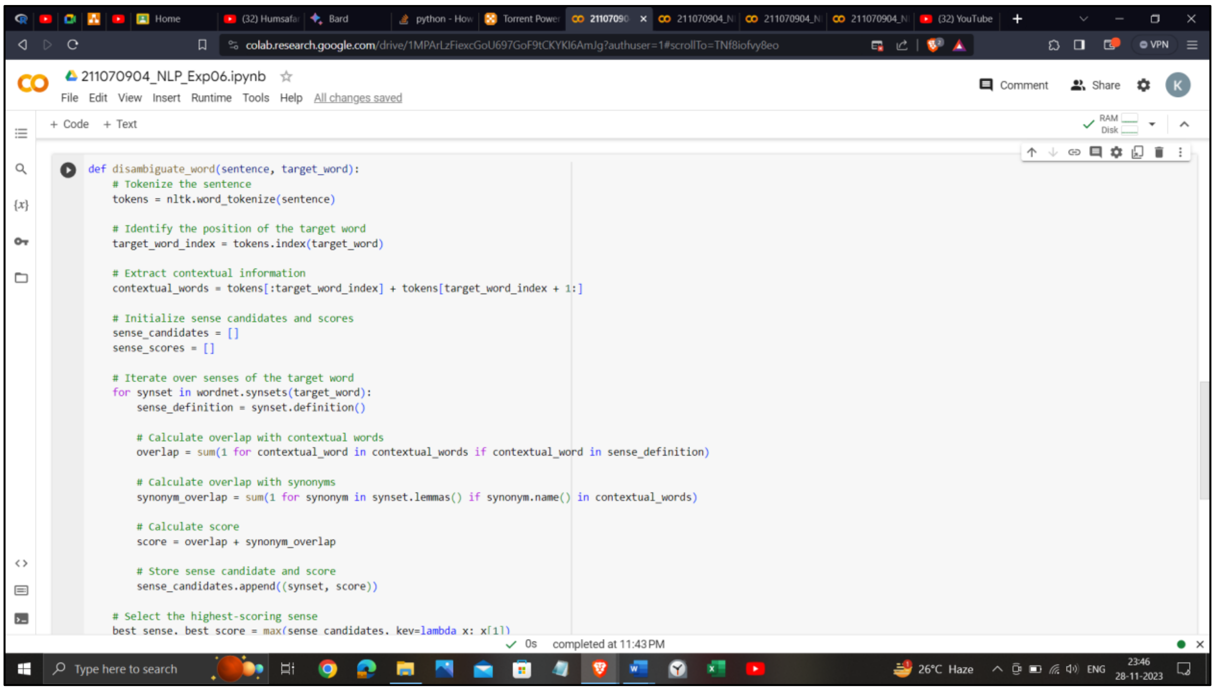
```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```python
[4] def get_all_synsets(word):
        synsets = wordnet.synsets(word)
        return synsets

    # Example usage

    # sentence = "The bat is a fascinating mammal that can fly in the night."
```

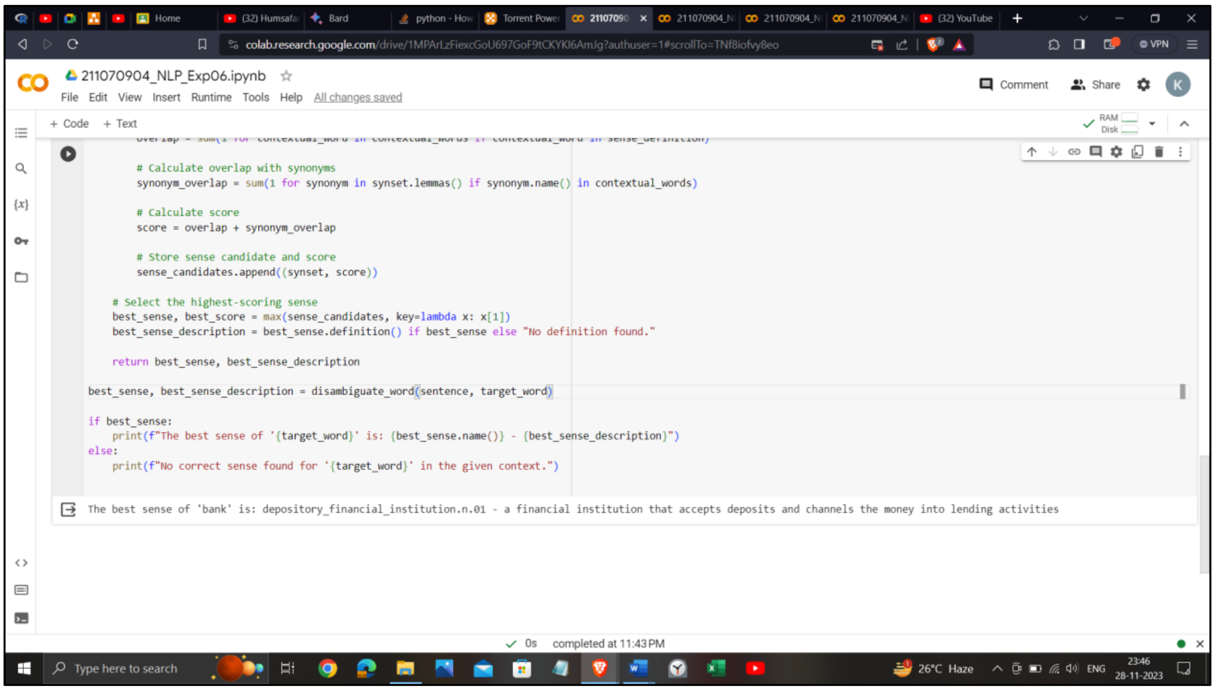


```python
sentence = "He deposited money in the bank."
target_word = "bank"

synsets = get_all_synsets(target_word)

if synsets:
    print(f"All possible senses of '{target_word}':")
    for synset in synsets:
        print(f"- {synset.name()} - {synset.definition()}")
else:
    print(f"No senses found for '{target_word}' in WordNet.")
```

```
All possible senses of 'bank':
- bank.n.01 - sloping land (especially the slope beside a body of water)
- depository_financial_institution.n.01 - a financial institution that accepts deposits and channels the money into lending activities
- bank.n.03 - a long ridge or pile
- bank.n.04 - an arrangement of similar objects in a row or in tiers
- bank.n.05 - a supply or stock held in reserve for future use (especially in emergencies)
- bank.n.06 - the funds held by a gambling house or the dealer in some gambling games
- bank.n.07 - a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force
- savings_bank.n.02 - a container (usually with a slot in the top) for keeping money at home
- bank.n.09 - a building in which the business of banking transacted
- bank.n.10 - a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)
- bank.v.01 - tip laterally
- bank.v.02 - enclose with a bank
- bank.v.03 - do business with a bank or keep an account at a bank
- bank.v.04 - act as the banker in a game or in gambling
- bank.v.05 - be in the banking business
- deposit.v.02 - put into a bank account
- bank.v.07 - cover with ashes so to control the rate of burning
- trust.v.01 - have confidence or faith in
```



```python
def disambiguate_word(sentence, target_word):
    # Tokenize the sentence
    tokens = nltk.word_tokenize(sentence)

    # Identify the position of the target word
    target_word_index = tokens.index(target_word)

    # Extract contextual information
    contextual_words = tokens[:target_word_index] + tokens[target_word_index + 1:]

    # Initialize sense candidates and scores
    sense_candidates = []
    sense_scores = []

    # Iterate over senses of the target word
    for synset in wordnet.synsets(target_word):
        sense_definition = synset.definition()

        # Calculate overlap with contextual words
        overlap = sum(1 for contextual_word in contextual_words if contextual_word in sense_definition)

        # Calculate overlap with synonyms
        synonym_overlap = sum(1 for synonym in synset.lemmas() if synonym.name() in contextual_words)

        # Calculate score
        score = overlap + synonym_overlap

        # Store sense candidate and score
        sense_candidates.append((synset, score))

    # Select the highest-scoring sense
    best_sense, best_score = max(sense_candidates, key=lambda x: x[1])
```



```python
        # Calculate overlap with synonyms
        synonym_overlap = sum(1 for synonym in synset.lemmas() if synonym.name() in contextual_words)

        # Calculate score
        score = overlap + synonym_overlap

        # Store sense candidate and score
        sense_candidates.append((synset, score))

    # Select the highest-scoring sense
    best_sense, best_score = max(sense_candidates, key=lambda x: x[1])
    best_sense_description = best_sense.definition() if best_sense else "No definition found."

    return best_sense, best_sense_description

best_sense, best_sense_description = disambiguate_word(sentence, target_word)

if best_sense:
    print(f"The best sense of '{target_word}' is: {best_sense.name()} - {best_sense_description}")
else:
    print(f"No correct sense found for '{target_word}' in the given context.")
```

```
The best sense of 'bank' is: depository_financial_institution.n.01 - a financial institution that accepts deposits and channels the money into lending activities
```

https://colab.research.google.com/drive/1MPArLzFiexcGoU697GoF9tCKYKI6AmJg?usp=sharing

Conclusion:

In summary, a knowledge-based approach for Word Sense Disambiguation (WSD) relies on external knowledge sources, such as WordNet, and linguistic features to determine the most appropriate sense of a word in context. This approach provides interpretable results but may face challenges with ambiguity. Continuous refinement and adaptation are key, and success hinges on the quality of the knowledge base and relevant features.