B.Sc. (Honours) Computer Science

(CBCS)

6th Semester

Module: Introduction to Computer Linguistics and NLTK.

Computer Linguistics

Paper Code: DSE 2

Arup Das

Assistant Professor,

Department of Computer Science,

Raja N. L. Khan Women's College (Autonomous),

Midnapore, West Bengal

Date :21/4/2020

Computer Linguistics

Table of Contents

1. Introduction

1.1 Computers in Linguistics and Natural Language Processing (NLP)
The field of computer linguistics involves using computational methods to process and analyze human language data. This includes both spoken and written forms of language. Natural Language Processing (NLP) is a subfield that focuses on enabling computers to understand, interpret, and generate human language. Key applications of NLP include machine translation, sentiment analysis, chatbots, and information retrieval.

1.2 The Nature and Use of Text Corpora
A corpus is a large and structured set of texts used for linguistic analysis. Text corpora are essential in NLP for training models, understanding language patterns, and statistical analysis. Corpora can be categorized into general corpora (covering a wide range of texts) and specialized corpora (focused on specific domains like medical, legal, or technical language).

---

2. Introduction to Python Programming and NLTK

2.1 Basics of Python Programming
Python is a popular programming language used in computational linguistics because of its simplicity and readability. It has a wide variety of libraries, including NLTK, which is specifically designed for NLP tasks. Basic Python concepts include variables, control structures (if-else, loops), functions, and data types (strings, lists, dictionaries).

2.2 Introduction to NLTK (Natural Language Toolkit)
NLTK is an open-source Python library for working with human language data. It provides easy access to text corpora, tools for text processing, and algorithms for various NLP tasks. With NLTK, users can tokenize text, tag parts of speech, parse sentence structure, and perform semantic analysis.

2.3 Regular Expressions
Regular expressions (regex) are sequences of characters that define a search pattern, primarily for string matching. In NLP, regex is used for pattern matching tasks like finding specific word forms or phrases in a text.

---

3. Pattern Matching

3.1 Corpus Search and Counting
Pattern matching allows for the identification of specific words, phrases, or patterns within a corpus. This can be used for tasks like word frequency counting, searching for particular linguistic structures, or extracting specific linguistic features.

3.2 Regular Languages
Regular languages are a class of languages that can be expressed using regular expressions. They are used to model simple linguistic patterns such as word formation rules or sentence structures.

3.3 Finite-State Automata
Finite-state automata (FSA) are mathematical models used to represent regular languages. FSAs can be used to describe and analyze linguistic phenomena such as word boundaries, morphological structures, and syntactic rules.

3.4 Operations and Closure Properties

Operations like union, intersection, and complementation can be performed on finite-state automata. These operations are important for understanding how languages can be combined and manipulated in NLP.

3.5 Pumping Lemma

The pumping lemma is a property of regular languages. It states that for any sufficiently long string in a regular language, it can be decomposed in a way that allows parts of the string to be repeated (pumped). This lemma helps in proving whether a language is regular.

---

4. Finite-State Linguistics

4.1 Transducers

A transducer is a type of finite-state machine that maps one sequence of symbols to another. In computational linguistics, transducers are used for tasks like morphological analysis and syntactic parsing.

4.2 Morphological Analysis

Morphological analysis involves breaking down words into their component morphemes, which are the smallest meaningful units in a language. This is important in NLP for tasks like word segmentation and stemming.

---

5. N-grams and Language Modeling

5.1 N-grams

An n-gram is a sequence of 'n' words occurring together in a text. N-grams are useful for modeling language patterns, such as predicting the likelihood of a given word sequence.

5.2 Language Modeling

Language models are probabilistic models used to predict the likelihood of word sequences. They are essential for tasks like speech recognition, machine translation, and text generation.

5.3 Smoothing

Smoothing techniques are used in language models to handle the problem of zero probability for unseen word sequences. Methods like add-one smoothing and back-off smoothing are commonly used.

5.4 Evaluation

Evaluation of language models involves testing their performance using metrics like perplexity and accuracy. Evaluation is essential to ensure the reliability and effectiveness of the model.

---

6. Part-of-Speech Tagging

6.1 Word Classes and Tagsets

Word classes (or parts of speech) include categories such as nouns, verbs, adjectives, and adverbs. Tagsets are sets of labels used to classify words into these categories. A comprehensive tagset is necessary for accurate POS tagging.

6.2 Rule-based and Stochastic POS Tagging

Rule-based POS tagging uses predefined linguistic rules to assign tags to words. Stochastic POS tagging, on the other hand, relies on statistical models trained on large corpora to make tagging decisions.

6.3 Hidden Markov Models (HMMs)

Hidden Markov Models are statistical models used for POS tagging. They assume that the current state (word tag) depends on the previous state (previous word tag) and a set of observed outputs (word forms).

6.4 Evaluation

The evaluation of POS tagging involves checking the accuracy of the tagged output against a gold standard corpus. Metrics such as precision, recall, and F1-score are used for evaluation.

---

7. Word Meaning

7.1 Semantic Ambiguity

Semantic ambiguity occurs when a word or phrase has multiple meanings. Resolving this ambiguity is an important task in NLP, particularly in tasks like word sense disambiguation (WSD).

7.2 Semantic Relations

Semantic relations describe how words are related in terms of their meanings. These include relations like synonyms (similar meanings), antonyms (opposite meanings), hyponyms (specific instances), and hypernyms (general categories).

7.3 Semantic Roles

Semantic roles refer to the roles played by entities in a sentence, such as agent, patient, or experiencer. Understanding these roles is crucial for interpreting sentence meaning.

---

8. Computational Lexical Semantics

Computational lexical semantics involves using computational methods to model and understand the meanings of words and their relationships. This includes the development of lexical databases like WordNet, which organizes words into sets of synonyms and defines semantic relationships among them.

This note provides a high-level overview of the field of computer linguistics, with a particular focus on key concepts in Natural Language Processing, lexical semantics, and associated computational techniques. It incorporates essential tools such as Python programming, NLTK, and finite-state automata for solving problems in linguistic analysis.

---

1. Introduction to Computer Linguistics

1.1 Computers in Linguistics and Natural Language Processing (NLP)

Computational linguistics is an interdisciplinary field at the intersection of linguistics and computer science. It involves the use of computational techniques to process, analyze, and model human language data. The goal is to create systems that can understand, interpret, and generate human language in a useful way.

Natural Language Processing (NLP) is a branch of artificial intelligence focused on enabling computers to understand, generate, and respond to human language in a natural and meaningful way. NLP involves tasks like machine translation, speech recognition, text summarization, and question answering.

Applications of NLP:

- **Machine Translation:** Translating text or speech from one language to another.
- **Information Retrieval:** Extracting useful information from large amounts of unstructured data.
- **Sentiment Analysis:** Determining the sentiment or emotion behind a piece of text.
- **Speech Recognition:** Converting spoken language into written text.

1.2 The Nature and Use of Text Corpora

A **corpus** (plural: corpora) is a large and structured collection of text data that is used for linguistic analysis. In NLP, corpora serve as the foundation for training language models, identifying linguistic patterns, and evaluating NLP systems.

Types of Corpora:

- **General Corpora:** Contain a wide variety of texts, such as literature, news articles, and web pages.
- **Specialized Corpora:** Focus on specific domains, such as legal, medical, or scientific texts.
- **Annotated Corpora:** Include additional linguistic information, such as part-of-speech tags, syntactic structures, or semantic annotations.

Corpus Analysis:

- **Frequency Analysis:** Counting how often words or phrases appear in a corpus.
- **Collocation Analysis:** Identifying words that frequently appear together, which may indicate semantic relationships.

---

2. Introduction to Python Programming and NLTK

2.1 Basics of Python Programming

Python is a widely used programming language in computational linguistics because of its simplicity, versatility, and extensive library support. Key Python concepts include:

- **Variables and Data Types:** Python supports data types such as strings, integers, floats, lists, dictionaries, and tuples.
- **Control Structures:** Python includes if-else conditions, for and while loops, and functions for controlling program flow.
- **Functions and Modules:** Functions allow you to encapsulate code and reuse it. Modules in Python help organize code into reusable components.

2.2 Introduction to NLTK (Natural Language Toolkit)

NLTK is a powerful Python library for working with human language data. It provides easy access to corpora, tools for text processing, and algorithms for tasks like tokenization, parsing, and semantic analysis.

Key NLTK Features:

- **Tokenization:** Splitting text into individual words or sentences.
- **Part-of-Speech Tagging:** Labeling words with their syntactic roles (noun, verb, etc.).
- **Named Entity Recognition (NER):** Identifying named entities such as person names, organizations, and locations.

2.3 Regular Expressions

Regular expressions (regex) are a powerful tool for matching patterns in strings. They are used in NLP for tasks like word searching, text pattern matching, and data extraction.

Basic Regex Concepts:

- **Metacharacters:** Special symbols like . (any character), * (zero or more), and [] (character classes).
- **Quantifiers:** Indicate how many times a pattern should occur, such as {m,n} for matching between m and n occurrences of a pattern.

Example:

```
import re

text = "The cat sat on the mat."

pattern = r"\b\w+\b"  # matches words

words = re.findall(pattern, text)

print(words)  # ['The', 'cat', 'sat', 'on', 'the', 'mat']
```

---

3. Pattern Matching and Regular Languages

3.1 Corpus Search and Counting

Pattern matching in NLP allows you to search through corpora and count the occurrences of words or phrases. This is often the first step in text analysis and feature extraction. Using regex or specialized libraries like nltk.FreqDist, you can identify word frequency, collocations, and other linguistic patterns.

3.2 Regular Languages

A **regular language** is a formal language that can be expressed using regular expressions or finite-state automata (FSA). Regular languages are characterized by simple patterns and are often used to model word-level features such as word boundaries, suffixes, and prefixes.

Examples of regular languages:

- A language of strings containing only lowercase vowels: a|e|i|o|u.
- A language of binary strings with an even number of 0s.

3.3 Finite-State Automata (FSA)

Finite-State Automata (FSA) are a class of mathematical models used to represent regular languages. FSAs are made up of states and transitions between those states based on input symbols. They are widely used in computational linguistics for tasks like tokenization and morphological analysis.

Types of FSAs:

- **Deterministic Finite Automata (DFA):** Every state has exactly one transition for each input symbol.
- **Non-deterministic Finite Automata (NFA):** A state may have multiple transitions for a given input symbol.

3.4 Operations and Closure Properties

Operations on FSAs include:

- **Union:** Combining two FSAs into a new FSA that accepts strings from both.
- **Intersection:** Creating an FSA that accepts only the strings accepted by both FSAs.
- **Complementation:** Creating an FSA that accepts strings not accepted by a given FSA.

Closure Properties of Regular Languages: Regular languages are closed under operations such as union, intersection, and complementation, meaning that applying these operations to regular languages will result in another regular language.

3.5 Pumping Lemma

The **Pumping Lemma** is a property of regular languages that states that for sufficiently long strings in a regular language, there exists a way to "pump" (repeat) a portion of the string without leaving the language. This lemma is used to prove that certain languages are not regular.

---

4. Finite-State Linguistics

4.1 Transducers

A **finite-state transducer** (FST) is an extension of a finite-state automaton, where the transition between states also produces an output. FSTs are used in computational linguistics for tasks such as morphological analysis (e.g., converting verb forms to their root forms) and syntactic analysis.

Example: A transducer could map the word "running" to its root form "run."

4.2 Morphological Analysis

Morphological analysis involves breaking words down into their smallest meaningful units, or **morphemes**. This process helps identify word forms and their base meanings. For example, "unhappiness" can be broken into the morphemes "un-", "happy", and "-ness".

---

5. N-grams and Language Modeling

5.1 N-grams

An **n-gram** is a contiguous sequence of n items from a given text or speech corpus. N-grams are used in probabilistic models to predict the likelihood of a word or phrase occurring based on the previous words.

Example: For the sentence "I love NLP", the 2-grams (bigrams) are:

- ("I", "love")
- ("love", "NLP")

5.2 Language Modeling

A **language model** is a statistical model that assigns probabilities to sequences of words. It helps predict the next word in a sentence based on the words that preceded it. The most common models are based on n-grams, hidden Markov models, or deep learning methods.

5.3 Smoothing

Smoothing techniques are used to handle situations where a word sequence has not been observed in the training data. Common smoothing techniques include:

- **Additive smoothing (Laplace smoothing):** Adds a small constant to the probability estimates to ensure no probability is zero.
- **Back-off smoothing:** Reduces the order of the n-gram model when the full n-gram is not observed.

5.4 Evaluation

The evaluation of language models can be done using metrics like:

- **Perplexity:** Measures how well a probability model predicts a sample. Lower perplexity indicates better performance.
- **Accuracy:** The percentage of correct predictions made by the model.

---

6. Part-of-Speech Tagging

6.1 Word Classes and Tagsets

Word classes (parts of speech) refer to categories like nouns, verbs, adjectives, adverbs, etc. Tagsets are sets of labels used to categorize words. A comprehensive POS tagset may include tags for different types of verbs, adjectives, adverbs, etc.

6.2 Rule-based and Stochastic POS Tagging

Rule-based tagging: Uses predefined linguistic rules to assign POS tags. These rules might depend on the word's context or its suffix.

Stochastic tagging: Uses statistical methods to predict POS tags based on probabilities derived from training data.

6.3 Hidden Markov Models (HMMs)

HMMs are a class of statistical models used in POS tagging. They assume that the current state (tag) depends on the previous state (previous tag), and a set of observed outputs (words) are generated by these states.

6.4 Evaluation

POS tagging is typically evaluated using accuracy, which measures the percentage of correctly tagged words in a test corpus.

---

7. Word Meaning

7.1 Semantic Ambiguity

Semantic ambiguity occurs when a word or phrase has multiple meanings. Resolving this ambiguity is a major challenge in NLP and requires context analysis.

Example: The word "bank" could refer to a financial institution or the side of a river.

7.2 Semantic Relations

Semantic relations describe how words are related in terms of meaning. Some common semantic relations include:

- **Synonymy:** Words with similar meanings (e.g., "happy" and "joyful").
- **Antonymy:** Words with opposite meanings (e.g., "hot" and "cold").
- **Hyponymy:** A word whose meaning is more specific than a general term (e.g., "cat" is a hyponym of "animal").

7.3 Semantic Roles

Semantic roles describe the roles played by entities in a sentence, such as agent (doer), patient (receiver of action), and experiencer (subject of perception).

---

8. Computational Lexical Semantics

Computational lexical semantics is the study of word meanings and their interrelationships using computational models. Lexical databases like **WordNet** are widely used for organizing word meanings into synonym sets (synsets) and defining relationships between words (e.g., hypernyms, hyponyms, antonyms).

References

1. **Bird, S., Klein, E., & Loper, E.** (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.
2. **Manning, C. D., & Schütze, H.** (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
3. **Goldberg, A.** (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers.
4. **Russell, S., & Norvig, P.** (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
5. **Miller, G. A.** (1995). *WordNet: A Lexical Database for English*. Communications of the ACM, 38(11), 39-41.
6. **Sproat, R.** (2000). *A Computational Theory of Writing Systems*. Cambridge University Press.
7. **Manning, C. D., & Hinrich Schütze, H.** (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
8. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., & Polosukhin, I.** (2017). *Attention is All You Need*. Advances in Neural Information Processing Systems (NeurIPS), 30.
9. **Harvard University NLP Group**. (n.d.). *Natural Language Processing with Python*. Retrieved from http://www.nltk.org/