LIN404

COMPUTATIONAL LINGUISTICS II

Course code: LIN 404
Course title: Computational Linguistics II
Course credit load: 2 Units
Series 1: Natural Language Processing Problems and Corpus Methods

Series 2: Probability, N-Grams, and Language Models

Series 3: Computational Phonetics, Morphology, and Syntax

Series 4: Computational Semantics and Meaning Representation

Series 5: Applications and Performance Evaluation in Computational Linguistics

Proposed Outline Series for 1

Part 1.1: Natural language processing problems and perspectives

- Sub-Part 1.1.1: Key challenges in NLP
- Sub-Part 1.1.2: Perspectives on solving NLP problems

Part 1.2: Corpus linguistics foundations

- Sub-Part 1.2.1: Definition and role of corpora
- Sub-Part 1.2.2: Methods of corpus construction and annotation
- Sub-Part 1.2.3: Applications of corpus linguistics in NLP

Part 1.3: Linking NLP problems with corpus methods

- Sub-Part 1.3.1: Using corpora to address NLP challenges
- Sub-Part 1.3.2: Case examples of corpus-based NLP solutions

Introduction

Natural language processing (NLP) is at the heart of modern computational linguistics, powering applications such as machine translation, speech recognition, and text analysis. Yet, working with human language presents unique challenges, from ambiguity and variability to the sheer scale of linguistic data. To address these challenges, corpus linguistics provides structured collections of texts that serve as the foundation for empirical analysis. This Series introduces you to the problems and perspectives in NLP and demonstrates how corpus methods can be applied to tackle them effectively.

The aim of this Series is to familiarise you with the key challenges in natural language processing and to equip you with the knowledge of corpus linguistics as a methodological tool for addressing these challenges.

We shall commence this Series by examining natural language processing problems and perspectives, highlighting the complexities of human language and the approaches used to manage them. Next, we will explore the foundations of corpus linguistics, including the definition, construction, and annotation of corpora, as well as their applications in computational linguistics. Finally, we will bring the Series to a close by linking NLP problems with corpus methods, showing how corpora can be strategically employed to address real-world language processing challenges through case examples.

Series Learning Outcomes

Upon completing this Series, you should be able to:

- **SLO 1.1** identify and explain key challenges in natural language processing problems and perspectives,
- **SLO 1.2** describe the role and definition of corpora in computational linguistics,
- **SLO 1.3** demonstrate how corpora are constructed and annotated for linguistic analysis,
- **SLO 1.4** analyse applications of corpus linguistics in addressing NLP challenges, and
- **SLO 1.5** apply corpus-based methods to illustrate solutions to selected NLP problems.

1.1 Natural Language Processing Problems And Perspectives

1.1.1 Key challenges in NLP

Natural language processing (**NLP**) refers to the computational study of human language, aiming to enable machines to analyse, interpret, and generate text or speech. One of the main challenges in NLP is **ambiguity**, which occurs when a word, phrase, or sentence has multiple possible meanings. For example, the word *bank* can mean a financial institution or the side of a river. Another challenge is **variability**, defined as the wide range of ways people express the same idea, such as *I am happy* versus *I feel joyful*. These challenges make it difficult for computational systems to achieve accurate interpretation.

Fill in the blank: The word *bank* can refer to a financial institution or the side of a ______.

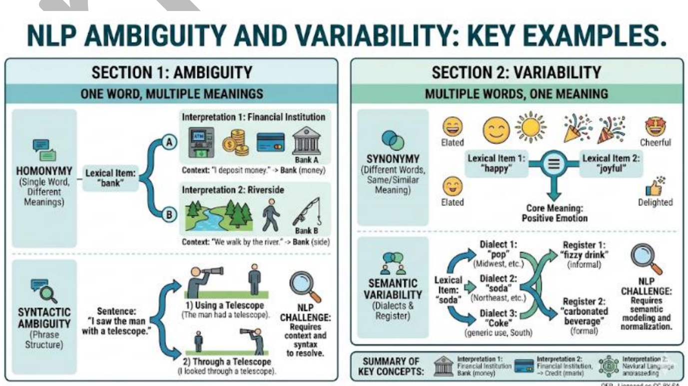


Figure 1.1: Key challenges in natural language processing

1.1.2 Perspectives on solving NLP problems

Different perspectives have been developed to address NLP challenges. The **rule-based perspective** relies on manually crafted grammatical rules to process language. While precise, it struggles with scalability and flexibility. The **statistical perspective** uses probability and large datasets to predict linguistic patterns, making it more adaptable to variation. More recently, the **machine learning perspective** has become dominant, applying algorithms that learn from data to improve performance over time. Each perspective contributes unique strengths, and modern NLP often combines them for better results.

Fill in the blank: The statistical perspective in NLP relies on ______ and large datasets to predict linguistic patterns.

Perspective	**Methodology**	**Primary Advantage**
Rule-Based	Relies on explicit, hand-coded grammatical rules, lexicons, and "if-then" logic.	**High Interpretability:** You know exactly why the system made a specific decision.
Statistical	Uses probability distributions and large datasets to predict the most likely next word or structure (e.g., Hidden Markov Models).	**Robustness:** Handles variation and "noisy" data better than strict rules.
Machine Learning	Applies algorithms (and increasingly deep learning/neural networks) that automatically learn features and patterns from data.	**Scalability:** Capable of capturing complex, non-linear relationships in language without manual feature engineering.

Box 1.1: Perspectives on solving NLP problems

Pilot questions 1.1

1. What does NLP stand for?
2. What is ambiguity in NLP, and give one example.
3. What is variability in NLP, and give one example.
4. What does the rule-based perspective rely on?
5. How does the statistical perspective address NLP problems?
6. What is the main advantage of the machine learning perspective?

1.2 Corpus Linguistics Foundations

1.2.1 Definition and role of corpora

A **corpus** is a large, structured collection of texts or speech samples used for linguistic analysis. In computational linguistics, corpora serve as the foundation for empirical research, allowing researchers to observe language patterns across authentic data rather than relying solely on intuition. Corpora are essential for training natural language processing systems, as they provide the raw material for tasks such as part-of-speech tagging, parsing, and semantic analysis.

Fill in the blank: A corpus is a structured collection of ______ used for linguistic analysis.

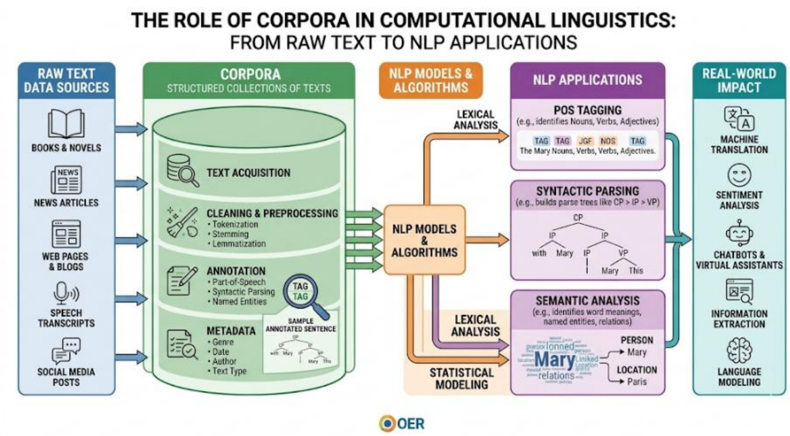


Figure 1.2: The role of corpora in computational linguistics

1.2.2 Methods of corpus construction and annotation

Building a corpus involves several steps, including text collection, cleaning, and annotation. **Annotation** refers to the process of adding linguistic information to raw text, such as part-of-speech tags, syntactic structures, or semantic roles. Annotated corpora are particularly valuable because they provide structured data for training and evaluating NLP models. For example, the Penn Treebank is a widely used annotated corpus that includes syntactic trees for English sentences.

Fill in the blank: Adding linguistic information such as part-of-speech tags to raw text is called ______.

Phase	Description	Key Activities
1. Text Collection	Gathering the raw data based on defined criteria (genre, time period, region).	Web scraping, digitizing print media, or recording spoken language.

Phase	Description	Key Activities
2. Cleaning & Formatting	Removing "noise" and standardizing the data into a machine-readable format.	Strip HTML tags, fix encoding errors, and normalize whitespace.
3. Annotation	Adding linguistic "tags" or labels to the text to make the data more useful.	Part-of-Speech (POS) tagging, Lemmatization, and Syntactic Treebanking.
4. Validation & QC	Checking the accuracy and consistency of the annotations and data.	Inter-annotator agreement (IAA) checks and automated error detection.

Box 1.2: Steps in corpus construction

1.2.3 Applications of corpus linguistics in NLP

Corpus linguistics provides practical applications in natural language processing. For instance, corpora are used to train **language models**, which predict word sequences based on probability. They also support **lexical analysis**, helping identify word frequencies and collocations. In addition, corpora are crucial for **evaluation**, as they provide benchmark datasets against which NLP systems can be tested. By grounding NLP in real-world data, corpus linguistics ensures that computational models reflect authentic language use.

Fill in the blank: Corpora are essential for training ______ models, which predict word sequences based on probability.

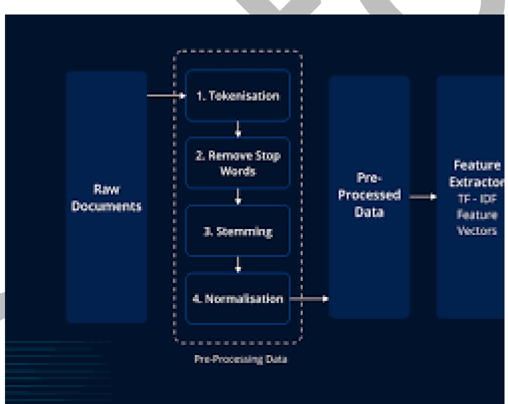


Figure 1.3: Applications of corpus linguistics in NLP

Pilot questions 1.2

1. What is a corpus, and why is it important in computational linguistics?
2. What is annotation in corpus construction?

3. Name one widely used annotated corpus.
4. List two steps involved in corpus construction.
5. What are language models trained on?
6. How do corpora support evaluation in NLP?

1.3 Linking NLP Problems With Corpus Methods

1.3.1 Using corpora to address NLP challenges

One of the most effective ways to tackle natural language processing challenges is through the use of **corpora**, which provide authentic language data for analysis. For example, ambiguity in words such as *bank* can be resolved by examining how the word is used across thousands of sentences in a corpus. Similarly, variability in expressions can be managed by identifying common collocations and patterns. Corpora allow computational models to learn from real-world usage rather than relying solely on theoretical rules.

Fill in the blank: Ambiguity in words such as *bank* can be resolved by examining how the word is used across thousands of sentences in a ______.

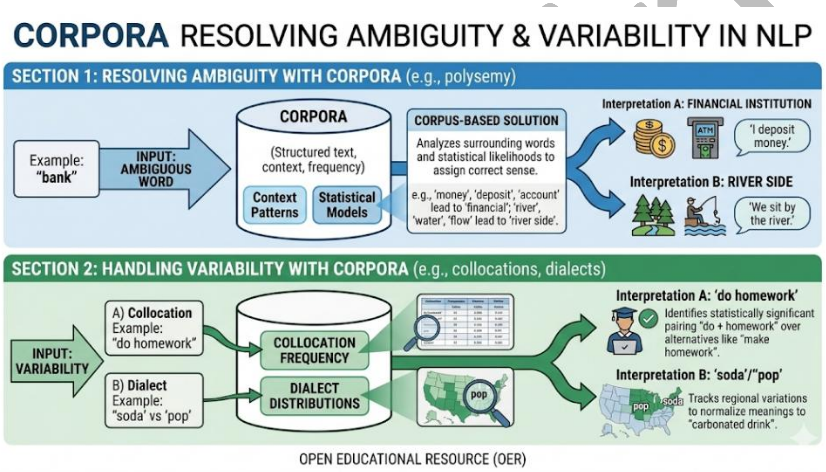


Figure 1.4: Using corpora to address NLP challenges

1.3.2 Case examples of corpus-based NLP solutions

Several case studies demonstrate how corpus methods directly solve NLP problems. For instance, **part-of-speech tagging** systems are trained on annotated corpora, enabling them to assign grammatical categories to words with high accuracy. Another example is **machine translation**, where parallel corpora (texts aligned across two languages) help systems learn translation equivalents. Corpora also support **sentiment analysis**, allowing models to detect positive or negative attitudes in text by learning from labelled datasets. These examples show how corpus linguistics bridges the gap between theoretical challenges and practical solutions in NLP.

Fill in the blank: Parallel corpora are especially useful in ______ translation systems, where texts are aligned across two languages.

NLP Solution	Corpus Type Used	Functional Outcome
Part-of-Speech (POS) Tagging	**Annotated Corpora** (e.g., Penn Treebank)	The system learns to identify word classes (Nouns, Verbs, etc.) based on the surrounding context and statistical frequency in the training data.
Machine Translation (MT)	**Parallel Corpora** (e.g., Europarl)	Large sets of "aligned" sentences in two or more languages allow the model to map equivalents and preserve meaning across grammatical shifts.
Sentiment Analysis	**Labelled Datasets** (e.g., IMDB reviews)	By analyzing millions of phrases tagged as "Positive" or "Negative," the AI identifies emotional "clusters" to predict the tone of new text.

Box 1.3: Case examples of corpus-based NLP solutions

Pilot questions 1.3

1. How can corpora help resolve ambiguity in NLP?
2. What role do collocations play in managing variability?
3. What are annotated corpora used for in NLP?
4. What type of corpora are used in machine translation?
5. How do corpora support sentiment analysis?

Series Summary

In this Series, we explored the foundational relationship between natural language processing (NLP) challenges and corpus linguistics as a methodological solution. The focus was on identifying the problems inherent in processing human language, understanding the role of corpora, and linking these two areas through practical applications.

We began with **Part 1.1: Natural language processing problems and perspectives**, where you identified key challenges such as ambiguity and variability, and examined different perspectives including rule-based, statistical, and machine learning approaches. We then moved to **Part 1.2: Corpus linguistics foundations**, where you defined corpora, described methods of corpus construction and annotation, and analysed their applications in NLP tasks such as language modelling and evaluation. Finally, in **Part 1.3: Linking NLP problems with corpus methods**, you applied corpus-based approaches to address NLP challenges, exploring case examples such as part-of-speech tagging, machine translation, and sentiment analysis.

By completing this Series, you should now be able to identify NLP challenges, describe the role of corpora, demonstrate corpus construction methods, analyse corpus applications, and apply corpus-based solutions to selected NLP problems. These skills directly align with the Series

Learning Outcomes (SLOs 1.1–1.5) and prepare you for deeper engagement with advanced empirical methods in subsequent Series.

Glossary Of Terms [Series 1]

- **Natural language processing (NLP)**: The computational study of human language, enabling machines to analyse, interpret, and generate text or speech.
- **Ambiguity**: A situation where a word, phrase, or sentence has multiple possible meanings, e.g., *bank* (financial institution vs. river side).
- **Variability**: The wide range of ways people express the same idea, e.g., *I am happy* vs. *I feel joyful*.
- **Rule-based perspective**: An approach to NLP that relies on manually crafted grammatical rules to process language.
- **Statistical perspective**: An approach that uses probability and large datasets to predict linguistic patterns.
- **Machine learning perspective**: An approach that applies algorithms which learn from data to improve performance over time.
- **Corpus**: A large, structured collection of texts or speech samples used for linguistic analysis and NLP training.
- **Annotation**: The process of adding linguistic information to raw text, such as part-of-speech tags, syntactic structures, or semantic roles.
- **Penn Treebank**: A widely used annotated corpus containing syntactic trees for English sentences.
- **Language model**: A computational system trained on corpora to predict word sequences based on probability.
- **Lexical analysis**: The study of word frequencies, collocations, and distributions within a corpus.
- **Collocation**: A sequence of words that commonly occur together, e.g., *strong tea* or *make a decision*.
- **Parallel corpus**: A corpus containing texts aligned across two languages, used in machine translation.
- **Part-of-speech (POS) tagging**: The process of assigning grammatical categories (e.g., noun, verb, adjective) to words in a text.
- **Sentiment analysis**: The computational detection of attitudes or emotions in text, often classifying content as positive, negative, or neutral.

Concept Check Questions [Series 1]

Objective Questions

1. Define NLP and state its primary purpose. (Linked to SLO 1.1)
2. Ambiguity in NLP refers to words or phrases with ______ meanings. (Linked to SLO 1.1)
3. Which perspective in NLP relies on manually crafted grammatical rules? (Linked to SLO 1.1)
4. A corpus is a structured collection of ______ used for linguistic analysis. (Linked to SLO 1.2)
5. Adding linguistic information such as part-of-speech tags to raw text is called ______. (Linked to SLO 1.3)
6. Parallel corpora are especially useful in ______ translation systems. (Linked to SLO 1.5)

Short-Answer Questions

7. Explain the difference between ambiguity and variability in NLP. (Linked to SLO 1.1)
8. Describe one limitation of the rule-based perspective in NLP. (Linked to SLO 1.1)
9. What is annotation, and why is it important in corpus construction? (Linked to SLO 1.3)
10. Name one widely used annotated corpus and its application. (Linked to SLO 1.3)
11. How do corpora support the training of language models? (Linked to SLO 1.4)
12. Give one example of how corpora are used in sentiment analysis. (Linked to SLO 1.5)

Long-Answer Questions

13. Analyse the three perspectives on solving NLP problems (rule-based, statistical, and machine learning), highlighting their strengths and weaknesses. (Linked to SLO 1.1)

14. Demonstrate the steps involved in corpus construction and explain how annotation enhances NLP applications. (Linked to SLO 1.2, SLO 1.3)

15. Apply corpus-based methods to illustrate solutions to NLP challenges, using examples such as part-of-speech tagging, machine translation, and sentiment analysis. (Linked to SLO 1.5)

Answers to Concept Check Questions [Series 1]

Objective Questions

1. Natural language processing; its purpose is to enable machines to analyse, interpret, and generate human language.
2. Multiple.
3. Rule-based perspective.
4. Texts or speech samples.
5. Annotation.

6. Machine.

Short-Answer Questions

7. Ambiguity occurs when a word has multiple meanings (e.g., *bank*), while variability refers to different ways of expressing the same idea (e.g., *happy* vs. *joyful*).

8. It struggles with scalability and flexibility, as manually crafted rules cannot cover all language variations.

9. Annotation is the process of adding linguistic information to raw text; it is important because it structures data for training and evaluating NLP models.

10. The Penn Treebank; it is used for syntactic analysis of English sentences.

11. Corpora provide authentic data that language models use to predict word sequences based on probability.

12. Labelled corpora allow models to learn patterns of positive or negative attitudes in text.

Long-Answer Questions

13. **Basic response**: Rule-based approaches rely on explicit rules but lack scalability; statistical approaches use probability and large datasets, offering adaptability; machine learning approaches learn from data and improve performance over time. **Value-added response**: While rule-based systems provide precision, they are rigid. Statistical methods adapt better but depend heavily on data quality. Machine learning offers flexibility and continuous improvement, but requires large annotated datasets and computational resources.

14. **Basic response**: Corpus construction involves text collection, cleaning, annotation, and validation. Annotation adds linguistic information such as part-of-speech tags. **Value-added response**: Annotated corpora enhance NLP applications by providing structured datasets for training models like parsers and taggers. For example, syntactic annotation in the Penn Treebank supports accurate parsing algorithms.

15. **Basic response**: Corpus-based methods solve NLP challenges by training part-of-speech taggers on annotated corpora, using parallel corpora for machine translation, and applying labelled datasets for sentiment analysis. **Value-added response**: These methods demonstrate how corpora bridge theoretical challenges and practical solutions. For instance, parallel corpora enable statistical machine translation systems to learn translation equivalents, while sentiment corpora allow models to detect nuanced emotional tones in text, improving applications such as customer feedback analysis.

Answers to Pilot Questions [Series 1]

Part 1.1

1. Natural language processing.
2. Ambiguity is when a word has multiple meanings, e.g., *bank*.
3. Variability is when the same idea is expressed in different ways, e.g., *I am happy* vs. *I feel joyful*.
4. Explicit grammatical rules.

5. Probability and large datasets.
6. They improve performance over time by learning from data.

Part 1.2

1. A corpus is a structured collection of texts or speech samples, important because it provides authentic data for computational linguistics.
2. Annotation; it adds linguistic information to raw text.
3. The Penn Treebank.
4. Text collection and annotation.
5. Corpora.
6. They provide benchmark datasets for testing NLP systems.

Part 1.3

1. By examining how words are used across thousands of sentences.
2. They help identify common patterns of word usage.
3. Training part-of-speech tagging systems.
4. Parallel corpora.
5. They provide labelled datasets that allow models to detect attitudes in text.

References and Attributions [Series 1]

This section consolidates references and illustration attributions for **Series 1: NLP Challenges and Corpus Linguistics**, presented in APA style.

Scholarly References

- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Draft. Stanford University.
- McEnery, T., & Hardie, A. (2012). *Corpus linguistics: Method, theory and practice*. Cambridge: Cambridge University Press.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly Media.
- Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics, 19*(2), 313–330.

Illustration Attributions

- **Figure 1.1: Key challenges in natural language processing** AI-generated suggestion. Prompt: "Create a diagram illustrating ambiguity and variability in NLP, with examples like bank (financial institution vs river side) and happy vs joyful." Suggested OER search string: "NLP ambiguity variability examples diagram OER".

- **Box 1.1: Perspectives on solving NLP problems** AI-generated suggestion. Prompt: "Create a text box listing perspectives on solving NLP problems: rule-based, statistical, and machine learning." Suggested OER search string: "NLP rule-based statistical machine learning perspectives OER".
- **Figure 1.2: The role of corpora in computational linguistics** AI-generated suggestion. Prompt: "Create a diagram showing corpora as structured collections of texts feeding into NLP applications like tagging, parsing, and semantic analysis." Suggested OER search string: "corpus linguistics role in NLP diagram OER".
- **Box 1.2: Steps in corpus construction** AI-generated suggestion. Prompt: "Create a text box listing steps in corpus construction: text collection, cleaning, annotation, validation." Suggested OER search string: "corpus construction annotation steps OER".
- **Figure 1.3: Applications of corpus linguistics in NLP** AI-generated suggestion. Prompt: "Create a diagram showing applications of corpus linguistics in NLP, including training language models, lexical analysis, and evaluation." Suggested OER search string: "applications of corpus linguistics NLP diagram OER".
- **Figure 1.4: Using corpora to address NLP challenges** AI-generated suggestion. Prompt: "Create a diagram showing corpora resolving ambiguity and variability in NLP, with examples like bank (financial vs river) and collocations." Suggested OER search string: "corpora resolving ambiguity variability NLP diagram OER".
- **Box 1.3: Case examples of corpus-based NLP solutions** AI-generated suggestion. Prompt: "Create a text box listing case examples of corpus-based NLP solutions: POS tagging, machine translation, sentiment analysis." Suggested OER search string: "corpus-based NLP solutions case studies OER".

Course code: LIN 404
Course title: Computational Linguistics II
Course credit load: 2 Units
Series 1: Natural Language Processing Problems and Corpus Methods

Series 2: Probability, N-Grams, and Language Models

Series 3: Computational Phonetics, Morphology, and Syntax

Series 4: Computational Semantics and Meaning Representation

Series 5: Applications and Performance Evaluation in Computational Linguistics

Proposed Outline Series for 2

Part 2.1: Fundamentals of probability calculus

- Sub-Part 2.1.1: Basic probability concepts
- Sub-Part 2.1.2: Probability in linguistic contexts

Part 2.2: N-Grams and their applications

- Sub-Part 2.2.1: Definition and construction of N-Grams
- Sub-Part 2.2.2: Applications of N-Grams in NLP tasks
- Sub-Part 2.2.3: Limitations of N-Gram models

Part 2.3: Language models and evaluation

- Sub-Part 2.3.1: Introduction to language models
- Sub-Part 2.3.2: Training language models with corpora
- Sub-Part 2.3.3: Methods for evaluating language model performance

Introduction

Language is full of patterns, and one of the most powerful ways to capture these patterns computationally is through probability. Every time you type on your phone and it predicts the next word, or when a translation system chooses the most likely phrase, probability is at work behind the scenes. By combining probability with structured models such as N-Grams and more advanced language models, computational linguistics provides tools that make machines better at handling human language. This Series introduces you to these essential concepts and shows how they form the backbone of modern natural language processing.

The aim of this Series is to equip you with the ability to apply probability calculus to linguistic data, construct and evaluate N-Gram models, and understand the principles behind language modelling in computational linguistics.

We shall commence this Series by exploring the fundamentals of probability calculus, focusing on basic concepts and their application to linguistic contexts. Next, we will move on to N-Grams, examining how they are defined, constructed, and applied in various NLP tasks, while also considering their limitations. Finally, we will wrap up by studying language models more broadly, looking at how they are trained using corpora and the methods used to evaluate their performance.

Series Learning Outcomes

Upon completing this Series, you should be able to:

- **SLO 2.1** define basic probability concepts and explain their application to linguistic data,
- **SLO 2.2** construct N-Gram models from text corpora,
- **SLO 2.3** analyse the applications and limitations of N-Gram models in natural language processing tasks,
- **SLO 2.4** describe the principles of language modelling and how models are trained using corpora, and
- **SLO 2.5** evaluate the performance of language models using appropriate methods.

2.1 Fundamentals Of Probability Calculus

2.1.1 Basic probability concepts

Probability is the measure of how likely an event is to occur, expressed as a number between 0 and 1. In computational linguistics, probability helps us quantify uncertainty in language. For example, the probability of a coin landing on heads is 0.5, while the probability of a sentence beginning with the word *The* can be calculated by examining its frequency in a corpus. Understanding probability allows us to model language behaviour mathematically and make predictions about word sequences.

Fill in the blank: Probability is expressed as a number between ______ and 1.

Probability Value	Meaning	Linguistic/Practical Example
0.0	**Impossible**	A sentence in English starting with the word *"the"* followed immediately by the verb *"eats"* (e.g., *"The eats..."*).
0.1 - 0.3	**Unlikely**	A random word from a dictionary being a specific technical term like *"morphophonemic"*.

Probability Value	Meaning	Linguistic/Practical Example
0.5	**Even Chance**	A fair **coin toss** (Heads or Tails); or in a simple binary choice task in a psycholinguistic experiment.
0.8 - 0.9	**Highly Likely**	An English sentence beginning with a determiner like **"The"** being followed by a noun (e.g., *"The cat..."*).
1.0	**Certain**	In a closed system, the probability that a sentence contains at least one linguistic unit (phoneme, morpheme, or word).

Figure 2.1: Probability scale with linguistic examples

2.1.2 Probability in linguistic contexts

In natural language processing, probability is applied to words, phrases, and sentences. For instance, the probability of a word appearing in a given context can be estimated by its frequency in a corpus. This is the basis of **language modelling**, where systems predict the likelihood of word sequences. A simple example is calculating the probability of the phrase *New York* by multiplying the probability of *New* followed by the probability of *York*. This approach helps computational systems manage ambiguity and variability in language.

Fill in the blank: The probability of the phrase *New York* is calculated by multiplying the probability of *New* followed by the probability of ______.

Concept	Definition	Linguistic Application
Word Frequency	The raw count of how often a specific word appears in a given corpus.	Helps identify "stop words" (like *the, is, at*) versus content-heavy keywords in a text.
Conditional Probability	The likelihood of a word (Word B) appearing, given that a specific word (Word A) has already appeared.	Powering predictive text: If you type "How," the probability of the next word being "are" is much higher than "zebra."
Sequence Probability	The total probability of an entire sentence, calculated by multiplying the individual probabilities of each word in order.	Used in Speech Recognition to decide if the user said "Recognize speech" versus "Wreck a nice beach."

Box 2.1: Probability in linguistic contexts

Pilot questions 2.1

1. What is probability, and how is it expressed?
2. Give one example of probability in everyday life.
3. How can probability be applied to language?
4. What is the probability of a coin landing on heads?
5. What is conditional probability in linguistic contexts?
6. How is the probability of a phrase like *New York* calculated?

2.2 N-Grams And Their Applications

2.2.1 Definition and construction of N-Grams

An **N-Gram** is a sequence of *N* items (usually words or characters) taken from a given text or speech sample. For example, a 2-Gram (bigram) might be *New York*, while a 3-Gram (trigram) could be *I love coding*. N-Grams are constructed by sliding a window of size *N* across a corpus, capturing overlapping sequences. They provide a simple yet powerful way to model language by showing which words tend to occur together.

Fill in the blank: A 2-Gram is also called a ______.

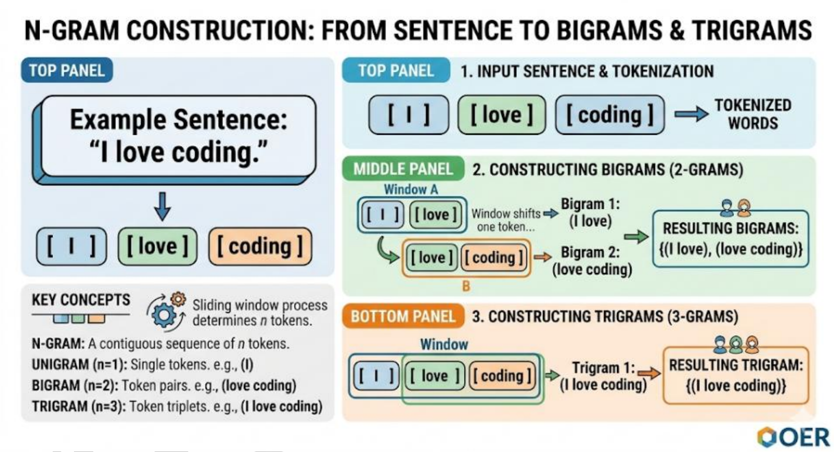


Figure 2.2: Construction of N-Grams from text

2.2.2 Applications of N-Grams in NLP tasks

N-Grams are widely used in natural language processing tasks. In **language modelling**, they help predict the next word in a sequence by calculating probabilities based on observed frequencies. In **speech recognition**, N-Grams improve accuracy by modelling likely word sequences. They are also used in **spelling correction**, where the system checks if a sequence of words is probable. These applications demonstrate how N-Grams provide a statistical foundation for handling language.

Fill in the blank: N-Grams are used in spelling correction to check if a sequence of words is ______.

Application	Role of N-Grams	Practical Benefit
Language Modelling	Predicting the $(n+1)^{th}$ word based on the previous $n-1$ words.	**Predictive Text:** Powers the "next-word" suggestions on smartphone keyboards and in email drafting.
Speech Recognition	Distinguishing between phonetically similar phrases based on their probability in a corpus.	**Accuracy:** Helps an AI decide that "I'll be there in a *minute*" is more likely than "I'll be there in a *minuet*."
Spelling Correction	Validating if a word "fits" within its immediate neighborhood of words.	**Contextual Correction:** Identifies errors even if the word is spelled correctly but used wrong (e.g., "They are over *there*" vs. "*their*").

Box 2.2: Applications of N-Grams in NLP

2.2.3 Limitations of N-Gram models

Despite their usefulness, N-Gram models have limitations. One major issue is **data sparsity**, which occurs when certain word combinations are rare or absent in the training corpus. Another limitation is **lack of long-range context**, since N-Grams only consider a fixed window of words. For example, a trigram model cannot capture dependencies between words separated by several sentences. These limitations have led to the development of more advanced models, such as neural language models, which can handle longer contexts and generalise better.

Fill in the blank: N-Gram models struggle with capturing ______ context beyond their fixed window size.

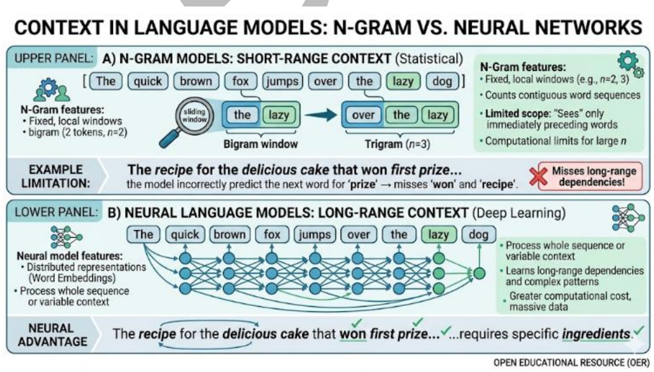


Figure 2.3: Limitations of N-Gram models compared to neural models

Pilot questions 2.2

1. What is an N-Gram?
2. What is another name for a 2-Gram?
3. How are N-Grams constructed from text?
4. Name two applications of N-Grams in NLP.
5. What is data sparsity in N-Gram models?
6. Why do N-Gram models struggle with long-range context?

2.3 Language Models And Evaluation

2.3.1 Introduction to language models

A **language model** is a computational system that assigns probabilities to sequences of words, predicting how likely a given sentence or phrase is to occur. Language models are central to natural language processing because they help machines generate text, recognise speech, and translate languages. For example, when your phone suggests the next word while typing, it is using a language model trained on large amounts of text data.

Fill in the blank: A language model assigns ______ to sequences of words.

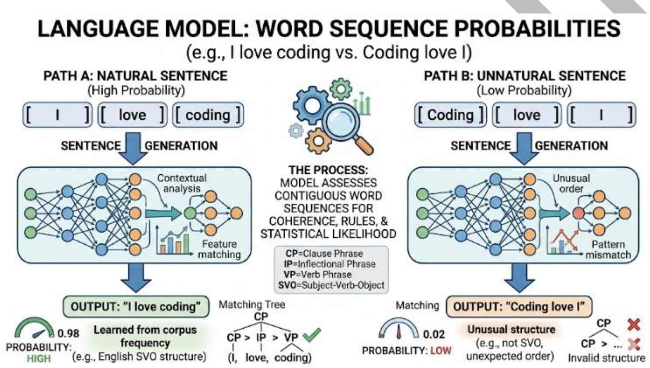


Figure 2.4: How language models assign probabilities to word sequences

2.3.2 Training language models with corpora

Language models are trained using **corpora**, which provide authentic examples of word usage. During training, the model learns the probability of words and sequences by analysing their frequency and context. For instance, in a corpus of English text, the model might learn that *New York* occurs more frequently than *York New*. Training can be done using simple N-Gram models or more advanced neural approaches, depending on the complexity required.

Fill in the blank: Language models are trained using ______, which provide authentic examples of word usage.

Stage	Component	Functional Role
Input	**Authentic Corpora**	The system is fed billions of words from books, articles, and transcripts to provide a representative sample of how humans actually communicate.
Process	**Statistical Analysis**	The model calculates how often words appear (**frequency**) and, more importantly, how often they appear together (**co-occurrence/N-grams**).
Output	**Sequence Probabilities**	The final model assigns a decimal value (0 to 1) to any given sequence, identifying "I love coding" as a high-probability string.

Box 2.3: Training language models

2.3.3 Methods for evaluating language model performance

Evaluating language models is crucial to ensure they perform effectively. One common method is **perplexity**, which measures how well a model predicts a sample of text. Lower perplexity indicates better performance. Another method is **accuracy**, often used in tasks like part-of-speech tagging or machine translation. Evaluation also involves comparing models against benchmark datasets to assess their ability to generalise. These methods ensure that language models are not only trained but also tested for reliability.

Fill in the blank: A lower ______ score indicates better performance of a language model.

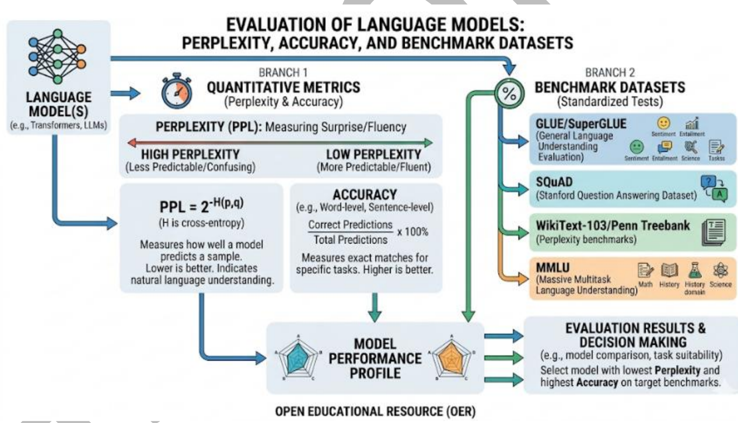


Figure 2.5: Methods for evaluating language model performance

Pilot questions 2.3

1. What is a language model?
2. How does a language model assign probabilities to word sequences?
3. What resource is used to train language models?

4. What does a lower perplexity score indicate?
5. Name one evaluation method for language models besides perplexity.
6. Why is evaluation important in language modelling?

Series Summary

In this Series, we examined how probability provides the mathematical foundation for modelling language, and how N-Grams and language models build upon this foundation to support natural language processing tasks. The focus was on applying probability to linguistic contexts, constructing N-Gram models, and evaluating language models for their effectiveness.

We began with **Part 2.1: Fundamentals of probability calculus**, where you defined basic probability concepts and explained their application to linguistic data. We then moved to **Part 2.2: N-Grams and their applications**, where you constructed N-Gram models, analysed their use in tasks such as language modelling and spelling correction, and considered their limitations such as data sparsity and lack of long-range context. Finally, in **Part 2.3: Language models and evaluation**, you described the principles of language modelling, explored how models are trained using corpora, and evaluated their performance using methods such as perplexity and accuracy.

By completing this Series, you should now be able to define probability concepts, construct and analyse N-Gram models, describe language modelling principles, and evaluate language model performance. These skills directly align with the Series Learning Outcomes (SLOs 2.1–2.5) and prepare you for deeper engagement with advanced computational models in subsequent Series.

Glossary Of Terms [Series 2]

- **Probability**: A measure of how likely an event is to occur, expressed as a number between 0 (impossible) and 1 (certain).
- **Conditional probability**: The likelihood of an event occurring given that another event has already occurred, e.g., the probability of *York* given *New*.
- **Sequence probability**: The probability of a phrase or sentence, calculated by multiplying the probabilities of individual words in order.
- **N-Gram**: A sequence of *N* items (usually words or characters) taken from text or speech, e.g., bigram (*New York*) or trigram (*I love coding*).
- **Bigram**: An N-Gram of size 2, representing two consecutive words or characters.
- **Trigram**: An N-Gram of size 3, representing three consecutive words or characters.
- **Data sparsity**: A limitation in N-Gram models where certain word combinations are rare or absent in the training corpus.
- **Language model**: A computational system that assigns probabilities to sequences of words, predicting how likely a given sentence or phrase is to occur.

- **Corpus**: A structured collection of texts or speech samples used to train and evaluate language models.
- **Perplexity**: A measure of how well a language model predicts a sample of text; lower perplexity indicates better performance.
- **Accuracy**: An evaluation metric that measures how often a model's predictions match the correct output, often used in tasks such as tagging or translation.

Concept Check Questions [Series 2]

Objective Questions

1. Define probability and state its range of values. (Linked to SLO 2.1)
2. Conditional probability refers to the likelihood of an event occurring given ______. (Linked to SLO 2.1)
3. A 2-Gram is also called a ______. (Linked to SLO 2.2)
4. Which limitation of N-Gram models occurs when certain word combinations are rare or absent in the training corpus? (Linked to SLO 2.3)
5. A language model assigns ______ to sequences of words. (Linked to SLO 2.4)
6. A lower perplexity score indicates ______ performance of a language model. (Linked to SLO 2.5)

Short-Answer Questions

7. Explain how probability is applied to linguistic contexts. (Linked to SLO 2.1)
8. Describe how N-Grams are constructed from text. (Linked to SLO 2.2)
9. Give one application of N-Grams in natural language processing. (Linked to SLO 2.2)
10. What is data sparsity, and why is it a challenge for N-Gram models? (Linked to SLO 2.3)
11. How are corpora used in training language models? (Linked to SLO 2.4)
12. Name one method of evaluating language models besides perplexity. (Linked to SLO 2.5)

Long-Answer Questions

13. Analyse the strengths and weaknesses of N-Gram models, giving examples of their applications and limitations. (Linked to SLO 2.2, SLO 2.3)

14. Demonstrate how language models are trained using corpora and explain why evaluation is necessary. (Linked to SLO 2.4, SLO 2.5)

15. Apply probability concepts to illustrate how language models predict word sequences, using examples such as bigrams and trigrams. (Linked to SLO 2.1, SLO 2.2)

Answers to Concept Check Questions [Series 2]

Objective Questions

1. Probability; values range between 0 and 1.
2. Another event has already occurred.
3. Bigram.
4. Data sparsity.
5. Probabilities.
6. Better.

Short-Answer Questions

7. Probability is applied to words, phrases, and sentences by estimating their likelihood based on frequency in a corpus.

8. N-Grams are constructed by sliding a window of size *N* across text, capturing overlapping sequences.

9. Language modelling, speech recognition, or spelling correction.

10. Data sparsity occurs when certain word combinations are rare or absent in the corpus, making probability estimates unreliable.

11. Corpora provide authentic text data from which models learn word frequencies and contexts.

12. Accuracy.

Long-Answer Questions

13. **Basic response**: N-Gram models are simple and effective for tasks like language modelling and spelling correction. However, they suffer from data sparsity and cannot capture long-range dependencies. **Value-added response**: While N-Grams provide a statistical foundation, their reliance on fixed window sizes limits their ability to model complex language structures. Neural models overcome these limitations by capturing longer contexts and generalising better.

14. **Basic response**: Language models are trained using corpora, which provide authentic examples of word usage. Evaluation ensures the model performs reliably, with perplexity and accuracy being common metrics. **Value-added response**: Training involves analysing word frequencies and contexts, while evaluation benchmarks the model against unseen data. This process ensures models generalise beyond training corpora, making them suitable for real-world applications such as translation and speech recognition.

15. **Basic response**: Probability concepts allow language models to predict word sequences. For example, the bigram probability of *New York* is calculated by multiplying the probability of *New* followed by *York*. **Value-added response**: Trigram models extend this by considering three-word sequences, improving predictions in contexts like speech recognition. However, their effectiveness depends on corpus size and quality, highlighting the importance of robust training data.

Answers to Pilot Questions [Series 2]

Part 2.1

1. Probability; expressed as a number between 0 and 1.
2. A coin toss.
3. By estimating the likelihood of words, phrases, or sentences based on their frequency in a corpus.
4. 0.5.
5. The likelihood of a word given the previous word.
6. By multiplying the probability of *New* followed by the probability of *York*.

Part 2.2

1. An N-Gram is a sequence of *N* items (words or characters) taken from text or speech.
2. Bigram.
3. By sliding a window of size *N* across text to capture overlapping sequences.
4. Language modelling and speech recognition (or spelling correction).
5. When certain word combinations are rare or absent in the training corpus.
6. They only consider a fixed window of words and cannot capture dependencies across longer contexts.

Part 2.3

1. A computational system that assigns probabilities to word sequences.
2. By calculating how likely a given sentence or phrase is to occur.
3. Corpora.
4. Better performance.
5. Accuracy.
6. To ensure models are reliable and generalise beyond training data.

References and Attributions [Series 2]

This section consolidates references and illustration attributions for **Series 2: Probability, N-Grams, and Language Models**, presented in APA style.

Scholarly References

- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Draft. Stanford University.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

- McEnery, T., & Hardie, A. (2012). *Corpus linguistics: Method, theory and practice*. Cambridge: Cambridge University Press.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly Media.

Illustration Attributions

- **Figure 2.1: Probability scale with linguistic examples** AI-generated suggestion. Prompt: "Create a diagram showing probability scale from 0 to 1, with examples coin toss (0.5) and sentence starting with 'The'." Suggested OER search string: "probability scale examples linguistics diagram OER".
- **Box 2.1: Probability in linguistic contexts** AI-generated suggestion. Prompt: "Create a text box listing probability applications in linguistics: word frequency, conditional probability, sequence probability." Suggested OER search string: "probability in linguistics conditional sequence corpus OER".
- **Figure 2.2: Construction of N-Grams from text** AI-generated suggestion. Prompt: "Create a diagram showing construction of N-Grams from a sentence, with examples of bigrams and trigrams." Suggested OER search string: "N-Gram construction examples diagram OER".
- **Box 2.2: Applications of N-Grams in NLP** AI-generated suggestion. Prompt: "Create a text box listing applications of N-Grams in NLP: language modelling, speech recognition, spelling correction." Suggested OER search string: "applications of N-Grams NLP examples OER".
- **Figure 2.3: Limitations of N-Gram models compared to neural models** AI-generated suggestion. Prompt: "Create a diagram comparing N-Gram models (short-range context) with neural language models (long-range context)." Suggested OER search string: "N-Gram limitations vs neural language models diagram OER".
- **Figure 2.4: How language models assign probabilities to word sequences** AI-generated suggestion. Prompt: "Create a diagram showing a language model predicting probabilities of word sequences, comparing natural vs unnatural sentences." Suggested OER search string: "language model probability word sequence diagram OER".
- **Box 2.3: Training language models** AI-generated suggestion. Prompt: "Create a text box outlining training language models: input corpora, process analysing frequencies, output probabilities." Suggested OER search string: "training language models corpora process OER".
- **Figure 2.5: Methods for evaluating language model performance** AI-generated suggestion. Prompt: "Create a diagram showing evaluation of language models using perplexity, accuracy, and benchmark datasets." Suggested OER search string: "language model evaluation perplexity accuracy diagram OER".

Course code: LIN 404
Course title: Computational Linguistics II
Course credit load: 2 Units
Series 1: Natural Language Processing Problems and Corpus Methods

Series 2: Probability, N-Grams, and Language Models

Series 3: Computational Phonetics, Morphology, and Syntax

Series 4: Computational Semantics and Meaning Representation

Series 5: Applications and Performance Evaluation in Computational Linguistics

Proposed Outline Series for 3

Part 3.1: Computational phonetics

- Sub-Part 3.1.1: Fundamentals of phonetics in computational linguistics
- Sub-Part 3.1.2: Speech sounds and their digital representation
- Sub-Part 3.1.3: Applications of computational phonetics in speech recognition

Part 3.2: Computational morphology

- Sub-Part 3.2.1: Morphological structures and processes
- Sub-Part 3.2.2: Rule-based and statistical approaches to morphology
- Sub-Part 3.2.3: Applications of computational morphology in NLP tasks

Part 3.3: Computational syntax

- Sub-Part 3.3.1: Fundamentals of syntax in computational linguistics
- Sub-Part 3.3.2: Parsing techniques and syntactic trees
- Sub-Part 3.3.3: Applications of computational syntax in machine translation and text analysis

Introduction

Language is not only about meaning but also about structure and sound. Every spoken word begins with phonetics, is shaped by morphology, and finds its place in syntax. These three levels of linguistic analysis are fundamental to computational linguistics, as they allow machines to process speech sounds, recognise word forms, and interpret sentence structures. By exploring computational approaches to phonetics, morphology, and syntax, this Series builds on the probability and modelling concepts introduced earlier, and moves closer to the structural foundations of language.

The aim of this Series is to introduce you to computational methods for analysing phonetics, morphology, and syntax, and to equip you with the skills needed to apply these methods in natural language processing tasks.

We shall commence this Series by examining computational phonetics, focusing on how speech sounds are represented digitally and applied in systems such as speech recognition. Next, we will move on to computational morphology, where we will study how word structures are modelled using rule-based and statistical approaches, and how these models support tasks such as lemmatisation and stemming. Finally, we will conclude with computational syntax, exploring parsing techniques, syntactic trees, and their applications in machine translation and text analysis.

Series Learning Outcomes

Upon completing this Series, you should be able to:

- **SLO 3.1** define the fundamental concepts of phonetics in computational linguistics,
- **SLO 3.2** demonstrate how speech sounds are represented digitally for computational purposes,
- **SLO 3.3** apply computational phonetics in tasks such as speech recognition,
- **SLO 3.4** describe morphological structures and processes relevant to computational modelling,
- **SLO 3.5** analyse rule-based and statistical approaches to computational morphology,
- **SLO 3.6** apply computational morphology techniques in natural language processing tasks such as stemming and lemmatisation,
- **SLO 3.7** explain the fundamentals of syntax in computational linguistics,
- **SLO 3.8** demonstrate parsing techniques and construct syntactic trees, and
- **SLO 3.9** evaluate the applications of computational syntax in machine translation and text analysis.

3.1 Computational Phonetics

3.1.1 Fundamentals of phonetics in computational linguistics

Phonetics is the study of speech sounds, focusing on how they are produced, transmitted, and perceived. In computational linguistics, phonetics is concerned with representing these sounds digitally so that machines can process spoken language. This involves mapping human speech into measurable acoustic features such as frequency, duration, and intensity. By doing so, systems can recognise and distinguish between different sounds, which is essential for applications like speech recognition.

Fill in the blank: Phonetics in computational linguistics focuses on representing ______ digitally so that machines can process spoken language.

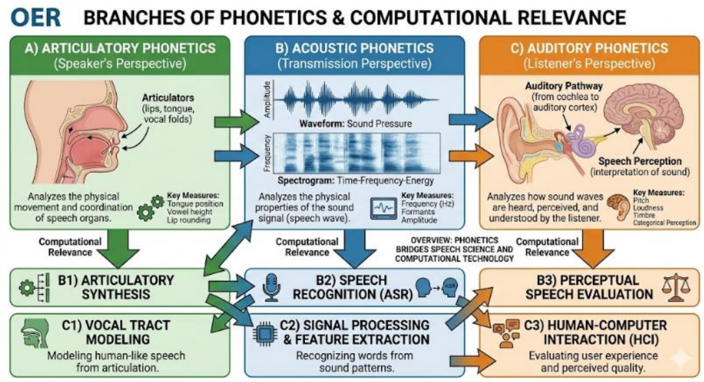


Figure 3.1: Branches of phonetics in computational linguistics

3.1.2 Speech sounds and their digital representation

Speech sounds are represented digitally using **phonetic transcription** and **acoustic features**. Phonetic transcription uses symbols, such as those from the International Phonetic Alphabet (IPA), to capture the exact sounds of speech. Acoustic features, on the other hand, are numerical values derived from sound waves, such as pitch and formants. Together, these representations allow computers to analyse and process speech data.

Fill in the blank: Speech sounds can be represented digitally using phonetic transcription and ______ features.

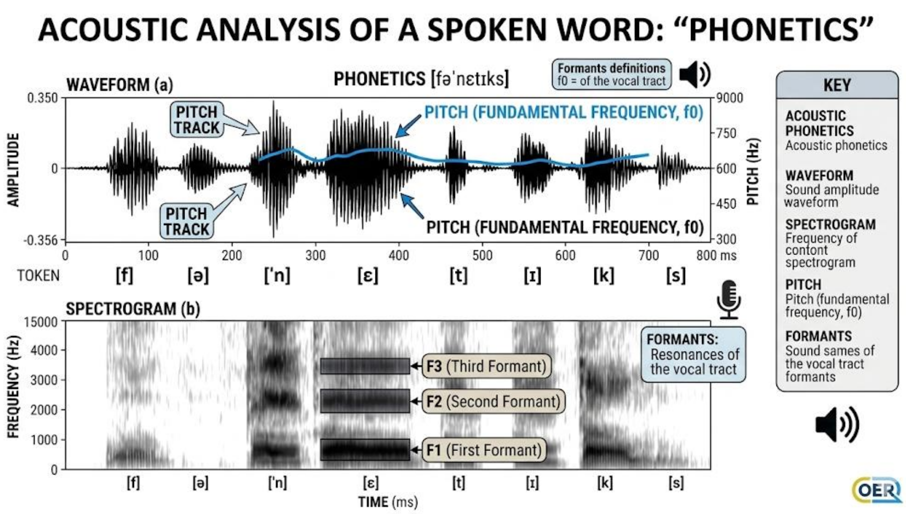


Plate 3.1: Digital representation of speech sounds

3.1.3 Applications of computational phonetics in speech recognition

Computational phonetics plays a vital role in **speech recognition**, where machines convert spoken language into text. By analysing acoustic features and phonetic patterns, systems can identify words even when spoken with different accents or speeds. For example, virtual assistants rely on computational phonetics to understand commands and respond accurately. This application demonstrates how phonetics bridges human speech and machine processing.

Fill in the blank: Computational phonetics is essential in ______ recognition, where machines convert spoken language into text.

Application	Role of Computational Phonetics	Practical Benefit
Speech Recognition (ASR)	Converting acoustic waveforms into digital text by identifying phonemes and word boundaries.	**Accessibility:** Powers voice-to-text features, virtual assistants (Siri, Alexa), and automated captioning.
Speaker Identification	Analyzing unique vocal characteristics like pitch, resonance, and tempo to verify identity.	**Security:** Used in biometric "voiceprints" for secure banking or forensic authorship verification.
Accent Adaptation	Adjusting acoustic models to recognize the phonetic variations found in different regional dialects.	**Inclusivity:** Ensures that speech technology works accurately for users regardless of their native accent or dialect.

Box 3.1: Applications of computational phonetics

Pilot questions 3.1

1. What is phonetics in computational linguistics concerned with?
2. Name the three branches of phonetics.
3. What symbols are used in phonetic transcription?
4. What are acoustic features in speech representation?
5. Give one application of computational phonetics.
6. How does computational phonetics support speech recognition?

3.2 Computational Morphology

3.2.1 Morphological structures and processes

Morphology is the study of word structure and how words are formed from smaller meaningful units called **morphemes**. A morpheme is the smallest unit of meaning in a language, such as *un-* (prefix), *happy* (root), or *-ness* (suffix). In computational linguistics, morphology is important because machines need to recognise and process these structures to understand and generate words correctly. For example, recognising that *unhappiness* is composed of three morphemes helps systems handle word formation and meaning more effectively.

Fill in the blank: The smallest unit of meaning in a language is called a ______.

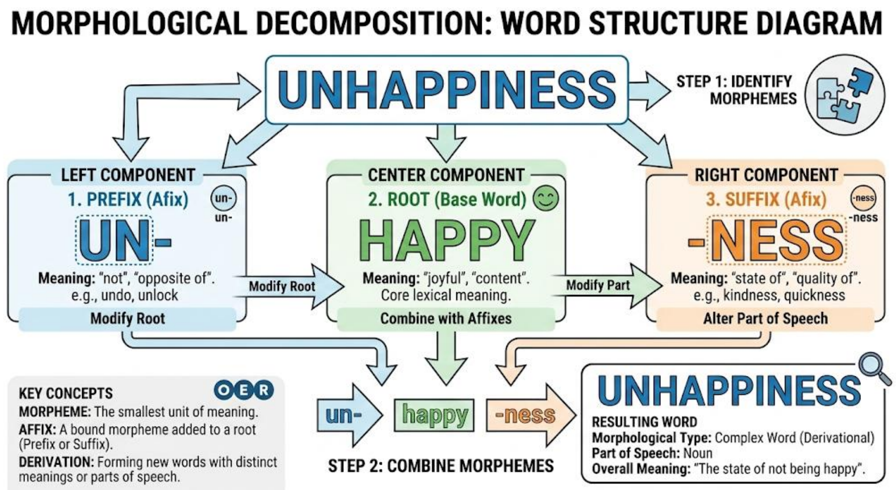


Figure 3.2: Morphological decomposition of a word

3.2.2 Rule-based and statistical approaches to morphology

Computational morphology can be approached in two main ways: **rule-based** and **statistical**. Rule-based approaches rely on explicit linguistic rules, such as affixation rules for forming plurals or verb conjugations. Statistical approaches, on the other hand, use large corpora and probability models to predict word forms based on observed patterns. While rule-based systems are precise, they can be rigid. Statistical systems are more flexible but may struggle with rare or irregular forms.

Fill in the blank: Rule-based approaches rely on explicit linguistic ______, while statistical approaches rely on large corpora and probability models.

Approach	Methodology	Primary Advantage	Typical Challenge
Rule-Based	Uses explicit linguistic rules (e.g., $Root + -ed = Past\ Tense$) and finite-state transducers.	Highly precise for languages with very regular structures.	Difficult to scale; fails when encountering "irregular" or new words.
Statistical	Uses probability models (like Hidden Markov Models) trained on massive corpora to "guess" morpheme boundaries.	Excellent at handling unknown words and linguistic variation/slang.	Requires huge amounts of annotated data to be accurate.
Hybrid	Combines a core set of linguistic rules with statistical learning to fill in the gaps.	Offers the best balance of accuracy and flexibility.	Most complex to design and computationally more "expensive" to run.

Box 3.2: Approaches to computational morphology

3.2.3 Applications of computational morphology in NLP tasks

Computational morphology is applied in many natural language processing tasks. **Stemming** reduces words to their base form by removing affixes, e.g., *running* → *run*. **Lemmatisation** goes further by mapping words to their dictionary form, e.g., *better* → *good*. These processes are essential for search engines, machine translation, and text analysis, as they help systems recognise related words and improve accuracy. Morphology also supports spell-checking and grammar correction by identifying incorrect word forms.

Fill in the blank: Lemmatisation maps words to their ______ form, such as *better* → *good*.

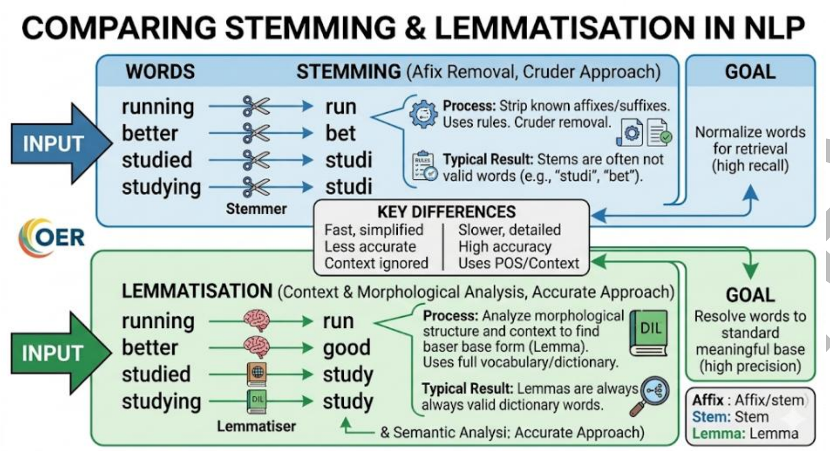


Figure 3.3: Stemming and lemmatisation compared

Pilot questions 3.2

1. What is morphology concerned with in linguistics?
2. What is the smallest unit of meaning in a language?
3. How does a rule-based approach to morphology work?
4. What is the main strength of statistical approaches to morphology?
5. Give one application of computational morphology in NLP.
6. What is the difference between stemming and lemmatisation?

3.3 Computational Syntax

3.3.1 Fundamentals of syntax in computational linguistics

Syntax is the study of how words combine to form phrases and sentences. In computational linguistics, syntax provides the rules and structures that allow machines to interpret sentence meaning beyond individual words. By modelling syntax, systems can distinguish between grammatically correct and incorrect sentences, and identify relationships such as subject, verb, and object. This is essential for tasks like machine translation, where word order and grammatical structure must be preserved.

Fill in the blank: Syntax in computational linguistics focuses on how ______ combine to form phrases and sentences.

Component	Function	Example: "The cat chased the mouse"
Subject (S)	The entity performing the action.	**The cat**
Verb (V)	The action or state being expressed.	**chased**
Object (O)	The entity receiving the action.	**the mouse**

Figure 3.4: Basic syntactic structure of a sentence

3.3.2 Parsing techniques and syntactic trees

Parsing is the process of analysing a sentence according to syntactic rules. A **syntactic tree** (or parse tree) visually represents the hierarchical structure of a sentence, showing how words group into phrases and phrases into sentences. Parsing techniques include rule-based methods, which rely on grammar rules, and statistical or machine learning methods, which use corpora to predict likely structures. Parse trees are widely used in computational linguistics to support applications such as grammar checking and semantic analysis.

Fill in the blank: A syntactic tree visually represents the ______ structure of a sentence.

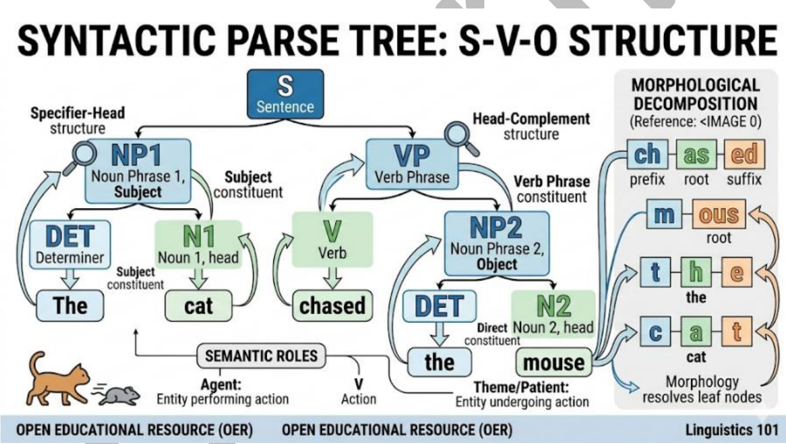


Figure 3.5: Parse tree representation of a sentence

3.3.3 Applications of computational syntax in machine translation and text analysis

Computational syntax is applied in **machine translation**, where syntactic rules help preserve meaning across languages by maintaining correct word order and sentence structure. It is also used in **text analysis**, where syntactic parsing supports tasks such as sentiment analysis, information extraction, and question answering. By modelling syntax, systems can move beyond word-level analysis to capture deeper linguistic relationships, improving accuracy and fluency in language processing.

Fill in the blank: Computational syntax is applied in machine translation to preserve meaning by maintaining correct ______ order and sentence structure.

Application	Role of Computational Syntax	Practical Benefit
Machine Translation	Mapping the syntactic structure of the source language to the rules of the target language.	**Accuracy:** Ensures that "The dog bit the man" isn't translated into a structure where the man bit the dog.
Text Analysis	Using **Dependency Parsing** to identify who did what to whom (Agent-Action-Theme).	**Insights:** Powers sentiment analysis by linking adjectives to the specific nouns they describe in complex sentences.
Grammar Checking	Comparing a user's input against a formal "Grammar" to find structural mismatches.	**Correction:** Identifies sophisticated errors like subject-verb disagreement or misplaced modifiers that simple spell-checkers miss.

Box 3.3: Applications of computational syntax

Pilot questions 3.3

1. What does syntax study in linguistics?
2. What are the three main components of a basic sentence structure?
3. What is parsing in computational linguistics?
4. What does a syntactic tree represent?
5. Name one application of computational syntax in NLP.
6. How does syntax support machine translation?

Series Summary

In this Series, we explored how computational methods are applied to the structural levels of language: phonetics, morphology, and syntax. These areas form the backbone of natural language processing, enabling machines to recognise speech sounds, process word structures, and interpret sentence organisation.

We began with **Part 3.1: Computational phonetics**, where you defined the fundamental concepts of phonetics, examined how speech sounds are represented digitally, and applied these principles to tasks such as speech recognition. We then moved to **Part 3.2: Computational morphology**, where you described morphological structures and processes, analysed rule-based and statistical approaches, and applied computational morphology techniques in tasks such as stemming and lemmatisation. Finally, in **Part 3.3: Computational syntax**, you explained the fundamentals of syntax, demonstrated parsing techniques and syntactic trees, and evaluated applications of computational syntax in machine translation and text analysis.

By completing this Series, you should now be able to define and demonstrate computational phonetics, analyse and apply computational morphology, and explain, demonstrate, and evaluate computational syntax. These skills directly align with the Series Learning Outcomes (SLOs 3.1–3.9) and provide a strong foundation for advancing into more complex areas of computational linguistics in subsequent Series.

Glossary Of Terms [Series 3]

- **Phonetics**: The study of speech sounds, focusing on how they are produced, transmitted, and perceived.
- **Acoustic features**: Numerical values derived from sound waves, such as pitch, duration, and formants, used to represent speech digitally.
- **Phonetic transcription**: The representation of speech sounds using symbols, often from the International Phonetic Alphabet (IPA).
- **Speech recognition**: A computational process that converts spoken language into text by analysing acoustic and phonetic patterns.
- **Morphology**: The study of word structure and how words are formed from smaller meaningful units called morphemes.
- **Morpheme**: The smallest unit of meaning in a language, such as prefixes (*un-*), roots (*happy*), or suffixes (*-ness*).
- **Rule-based approach (to morphology)**: A computational method that relies on explicit linguistic rules to model word formation.
- **Statistical approach (to morphology)**: A computational method that uses corpora and probability models to predict word forms based on observed patterns.
- **Stemming**: A process that reduces words to their base form by removing affixes, e.g., *running → run*.
- **Lemmatisation**: A process that maps words to their dictionary form, e.g., *better → good*.
- **Syntax**: The study of how words combine to form phrases and sentences, focusing on grammatical structure.
- **Parsing**: The process of analysing a sentence according to syntactic rules to determine its structure.
- **Syntactic tree (parse tree)**: A hierarchical representation of a sentence showing how words group into phrases and phrases into sentences.

- **Machine translation**: A computational process that translates text from one language to another while preserving meaning and grammatical structure.
- **Text analysis**: The computational examination of text to extract information, identify sentiment, or support tasks such as question answering.

Concept Check Questions [Series 3]

Objective Questions

1. Define phonetics in computational linguistics. (Linked to SLO 3.1)
2. Speech sounds are represented digitally using phonetic transcription and ______ features. (Linked to SLO 3.2)
3. Computational phonetics is essential in ______ recognition, where machines convert spoken language into text. (Linked to SLO 3.3)
4. The smallest unit of meaning in a language is called a ______. (Linked to SLO 3.4)
5. Rule-based approaches to morphology rely on explicit linguistic ______. (Linked to SLO 3.5)
6. Lemmatisation maps words to their ______ form. (Linked to SLO 3.6)
7. Syntax focuses on how ______ combine to form phrases and sentences. (Linked to SLO 3.7)
8. A syntactic tree visually represents the ______ structure of a sentence. (Linked to SLO 3.8)
9. Computational syntax is applied in machine translation to preserve meaning by maintaining correct ______ order. (Linked to SLO 3.9)

Short-Answer Questions

10. Explain how speech sounds are represented digitally in computational phonetics. (Linked to SLO 3.2)

11. Describe one application of computational phonetics in natural language processing. (Linked to SLO 3.3)

12. What is the difference between stemming and lemmatisation? (Linked to SLO 3.6)

13. How do rule-based approaches to morphology differ from statistical approaches? (Linked to SLO 3.5)

14. What is parsing in computational linguistics? (Linked to SLO 3.8)

15. Name one application of computational syntax in text analysis. (Linked to SLO 3.9)

Long-Answer Questions

16. Analyse the importance of computational phonetics in speech recognition, giving examples of how acoustic features are used. (Linked to SLO 3.1, SLO 3.2, SLO 3.3)

17. Demonstrate how computational morphology supports NLP tasks such as stemming and lemmatisation, and explain why these processes are important. (Linked to SLO 3.4, SLO 3.5, SLO 3.6)

18. Evaluate the role of parsing and syntactic trees in computational syntax, and discuss their applications in machine translation. (Linked to SLO 3.7, SLO 3.8, SLO 3.9)

Answers to Concept Check Questions [Series 3]

Objective Questions

1. Phonetics is the study of speech sounds and their digital representation in computational linguistics.
2. Acoustic.
3. Speech.
4. Morpheme.
5. Rules.
6. Dictionary.
7. Words.
8. Hierarchical.
9. Word.

Short-Answer Questions

10. Speech sounds are represented digitally using phonetic transcription (IPA symbols) and acoustic features such as pitch and formants.

11. Speech recognition, where spoken language is converted into text.

12. Stemming reduces words to their base form by removing affixes, while lemmatisation maps words to their dictionary form.

13. Rule-based approaches rely on explicit linguistic rules, while statistical approaches use corpora and probability models to predict word forms.

14. Parsing is the process of analysing a sentence according to syntactic rules to determine its structure.

15. Sentiment analysis or information extraction.

Long-Answer Questions

16. **Basic response**: Computational phonetics is important in speech recognition because it allows machines to analyse acoustic features such as pitch and formants to identify words.
Value-added response: Beyond recognition, computational phonetics supports speaker identification and accent adaptation, making systems more robust across diverse linguistic contexts.

17. **Basic response**: Computational morphology supports NLP tasks by reducing words to their base form (stemming) or dictionary form (lemmatisation), improving search and text analysis. **Value-added response**: These processes enhance machine translation and grammar correction by ensuring systems recognise related words and handle irregular forms effectively.

18. **Basic response**: Parsing and syntactic trees represent sentence structure, helping systems analyse grammar and meaning. They are applied in machine translation to preserve word order. **Value-added response**: Parsing also supports text analysis, question answering, and grammar checking. Advanced statistical parsers improve accuracy by learning from large corpora, making them adaptable to multiple languages.

Answers to Pilot Questions [Series 3]

Part 3.1

1. Representing speech sounds digitally so machines can process spoken language.
2. Articulatory, acoustic, and auditory phonetics.
3. International Phonetic Alphabet (IPA) symbols.
4. Acoustic features such as pitch and formants.
5. Speech recognition.
6. By analysing acoustic features and phonetic patterns to identify words.

Part 3.2

1. Word structure and how words are formed from morphemes.
2. Morpheme.
3. By applying explicit linguistic rules such as affixation or conjugation.
4. Flexibility in predicting word forms based on observed patterns in corpora.
5. Stemming or lemmatisation in NLP tasks.
6. Stemming reduces words to their base form, while lemmatisation maps words to their dictionary form.

Part 3.3

1. How words combine to form phrases and sentences.
2. Subject, verb, and object.
3. Analysing a sentence according to syntactic rules to determine its structure.
4. The hierarchical structure of a sentence.
5. Machine translation or text analysis.
6. By preserving meaning through maintaining correct word order and sentence structure.

References and Attributions [Series 3]

This section consolidates references and illustration attributions for **Series 3: Computational Approaches to Phonetics, Morphology, and Syntax**, presented in APA style.

Scholarly References

- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Draft. Stanford University.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.
- McEnery, T., & Hardie, A. (2012). *Corpus linguistics: Method, theory and practice*. Cambridge: Cambridge University Press.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly Media.

Illustration Attributions

- **Figure 3.1: Branches of phonetics in computational linguistics** AI-generated suggestion. Prompt: "Create a diagram showing articulatory, acoustic, and auditory phonetics, with arrows linking to computational applications." Suggested OER search string: "branches of phonetics computational linguistics diagram OER".
- **Plate 3.1: Digital representation of speech sounds** AI-generated suggestion. Prompt: "Create an image showing waveform and spectrogram of a spoken word, with pitch and formants labelled." Suggested OER search string: "waveform spectrogram speech sounds IPA OER".
- **Box 3.1: Applications of computational phonetics** AI-generated suggestion. Prompt: "Create a text box listing applications of computational phonetics: speech recognition, speaker identification, accent adaptation." Suggested OER search string: "applications of computational phonetics speech recognition OER".
- **Figure 3.2: Morphological decomposition of a word** AI-generated suggestion. Prompt: "Create a diagram showing decomposition of the word 'unhappiness' into prefix, root, and suffix." Suggested OER search string: "morphological decomposition word structure diagram OER".
- **Box 3.2: Approaches to computational morphology** AI-generated suggestion. Prompt: "Create a text box comparing rule-based, statistical, and hybrid approaches to computational morphology." Suggested OER search string: "rule-based vs statistical approaches computational morphology OER".
- **Figure 3.3: Stemming and lemmatisation compared** AI-generated suggestion. Prompt: "Create a diagram comparing stemming and lemmatisation with examples like running→run and better→good." Suggested OER search string: "stemming vs lemmatisation diagram NLP OER".

- **Figure 3.4: Basic syntactic structure of a sentence** AI-generated suggestion. Prompt: “Create a diagram showing sentence structure with subject, verb, and object labelled.” Suggested OER search string: “syntactic structure subject verb object diagram OER”.
- **Figure 3.5: Parse tree representation of a sentence** AI-generated suggestion. Prompt: “Create a diagram showing a parse tree for the sentence ‘The cat chased the mouse’.” Suggested OER search string: “parse tree syntactic structure diagram OER”.
- **Box 3.3: Applications of computational syntax** AI-generated suggestion. Prompt: “Create a text box listing applications of computational syntax: machine translation, text analysis, grammar checking.” Suggested OER search string: “applications of computational syntax NLP OER”.

Course code: LIN 404
Course title: Computational Linguistics II
Course credit load: 2 Units
Series 1: Natural Language Processing Problems and Corpus Methods

Series 2: Probability, N-Grams, and Language Models

Series 3: Computational Phonetics, Morphology, and Syntax

Series 4: Computational Semantics and Meaning Representation

Series 5: Applications and Performance Evaluation in Computational Linguistics

Proposed Outline Series for 4

Part 4.1: Computational semantics

- Sub-Part 4.1.1: Fundamentals of semantics in computational linguistics
- Sub-Part 4.1.2: Lexical semantics and word meaning
- Sub-Part 4.1.3: Semantic relations and networks

Part 4.2: Meaning representation frameworks

- Sub-Part 4.2.1: Logic-based approaches (predicate logic, semantic roles)
- Sub-Part 4.2.2: Frame-based and conceptual structures
- Sub-Part 4.2.3: Vector-based representations (embeddings)

Part 4.3: Applications of computational semantics

- Sub-Part 4.3.1: Semantic parsing and question answering
- Sub-Part 4.3.2: Machine translation and semantic equivalence
- Sub-Part 4.3.3: Text summarisation and information extraction

Introduction

Language is not only about sounds, words, and structures; it is also about meaning. While phonetics, morphology, and syntax provide the building blocks of language, semantics focuses on how meaning is conveyed and interpreted. In computational linguistics, semantics and meaning representation are crucial because machines must not only recognise words and structures but also understand what they signify. This Series therefore takes us into the heart of meaning in language processing, showing how computational methods capture and represent meaning for practical applications.

The aim of this Series is to introduce you to computational approaches to semantics and meaning representation, and to equip you with the skills needed to analyse, model, and apply semantic frameworks in natural language processing tasks.

We shall commence this Series by exploring computational semantics, beginning with the fundamentals of meaning in language, lexical semantics, and semantic relations. Next, we will move on to meaning representation frameworks, examining logic-based approaches, frame-based structures, and vector-based embeddings. Finally, we will conclude with applications of computational semantics, focusing on semantic parsing, machine translation, and text summarisation.

Series Learning Outcomes

Upon completing this Series, you should be able to:

- **SLO 4.1** define the fundamental concepts of semantics in computational linguistics,
- **SLO 4.2** explain lexical semantics and how word meaning is represented computationally,
- **SLO 4.3** analyse semantic relations and networks within computational frameworks,
- **SLO 4.4** describe logic-based approaches to meaning representation such as predicate logic and semantic roles,
- **SLO 4.5** demonstrate frame-based and conceptual structures for representing meaning,
- **SLO 4.6** apply vector-based representations such as embeddings to capture word meaning,
- **SLO 4.7** illustrate how semantic parsing supports tasks like question answering,
- **SLO 4.8** evaluate the role of computational semantics in machine translation and semantic equivalence, and
- **SLO 4.9** apply meaning representation techniques in text summarisation and information extraction.

4.1 Computational Semantics

4.1.1 Fundamentals of semantics in computational linguistics

Semantics is the study of meaning in language. In computational linguistics, semantics focuses on how machines can represent and process meaning so that they can interpret text beyond surface-level word forms. This involves modelling how words, phrases, and sentences convey meaning, and how these meanings can be captured in computational systems. Without semantics, machines may recognise words but fail to grasp what they signify in context.

Fill in the blank: In computational linguistics, semantics focuses on how machines can represent and process ______.

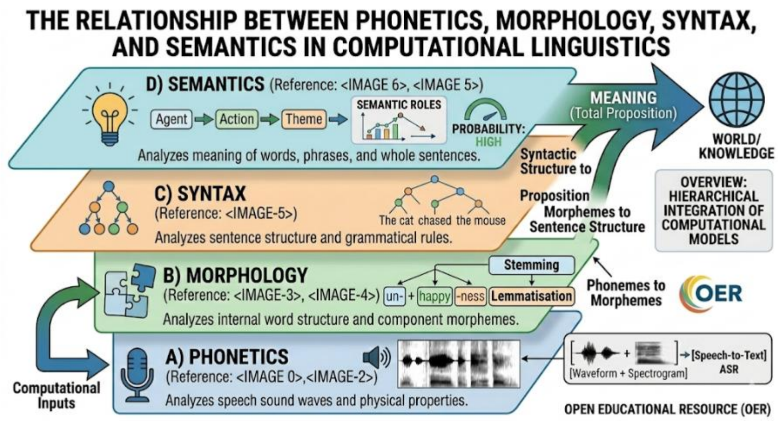


Figure 4.1: Levels of linguistic analysis leading to semantics

4.1.2 Lexical semantics and word meaning

Lexical semantics is the study of word meaning and how words relate to one another. In computational linguistics, lexical semantics involves representing words in ways that capture their meanings, such as dictionaries, thesauri, or computational models. Words can have multiple meanings (polysemy) or share similar meanings (synonymy). Machines must be able to distinguish these relationships to process language accurately. For example, recognising that *bank* can mean a financial institution or the side of a river is crucial for correct interpretation.

Fill in the blank: Lexical semantics is the study of ______ meaning and how words relate to one another.

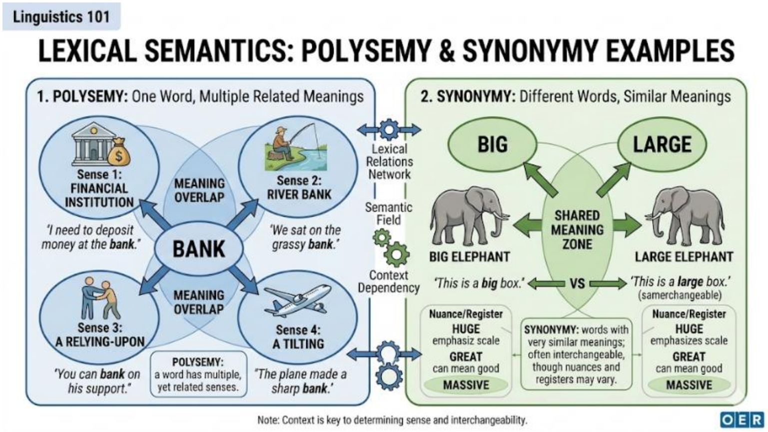


Plate 4.1: Examples of lexical semantics in word meaning

4.1.3 Semantic relations and networks

Semantic relations describe how words are connected in meaning, such as synonymy (similar meaning), antonymy (opposite meaning), hyponymy (specific vs general), and meronymy (part-whole relationships). These relations can be represented in **semantic networks**, which are graphs showing words as nodes and their relationships as links. Semantic networks help machines organise and retrieve meaning efficiently, supporting tasks such as word sense disambiguation and information retrieval.

Fill in the blank: Semantic networks represent words as nodes and their ______ as links.

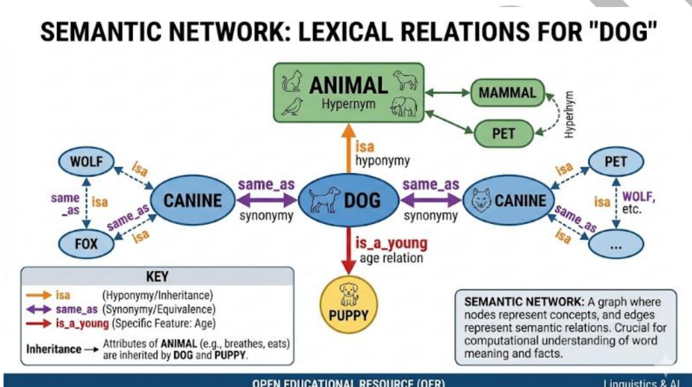


Figure 4.2: Example of a semantic network

Pilot questions 4.1

1. What is semantics concerned with in computational linguistics?
2. What is lexical semantics?
3. Give one example of polysemy.
4. What is synonymy?
5. Name two types of semantic relations.
6. What is a semantic network?

4.2 Meaning Representation Frameworks

4.2.1 Logic-based approaches (predicate logic, semantic roles)

Meaning representation refers to the way in which computational systems capture and formalise meaning. One common approach is **logic-based representation**, which uses formal systems such as **predicate logic** to describe relationships between entities. Predicate logic expresses meaning in terms of subjects, predicates, and arguments, allowing machines to reason about statements. Another important concept is **semantic roles**, which identify the function of words in a sentence, such as agent (doer of an action), patient (receiver of an action), or instrument (means by which an action is performed). These frameworks provide precision and clarity in representing meaning computationally.

Fill in the blank: Logic-based meaning representation often uses ______ logic to describe relationships between entities.

```
[ KICK ]  <-- Predicate (Action)
      /   \
  (Agent)  (Patient)
    /         \
 [ BOY ]     [ BALL ]
```

Component	Word	Logic Term	Semantic Role
Subject	The boy	Argument (x)	**Agent**
Verb	kicked	Predicate (P)	**Action**
Object	the ball	Argument (y)	**Patient**

Figure 4.3: Logic-based meaning representation with semantic roles

4.2.2 Frame-based and conceptual structures

Frame-based representation models meaning using **frames**, which are structured descriptions of situations, events, or concepts. A frame consists of slots (attributes) and fillers (values) that capture the relationships and roles involved. For example, a "Buying" frame may include slots such as Buyer, Seller, Item, and Price. **Conceptual structures** extend this idea by organising knowledge into hierarchies and networks, enabling machines to interpret meaning in context. These approaches are particularly useful in tasks like information extraction and dialogue systems, where context and relationships matter.

Fill in the blank: A frame consists of slots and ______ that capture relationships and roles.

Slot	Filler (Example)
Buyer	Alice
Seller	TechMart Inc.
Item	Wireless Headphones
Price	$150

Box 4.1: Example of a “Buying” frame

4.2.3 Vector-based representations (embeddings)

Modern computational semantics often relies on **vector-based representations**, also known as **embeddings**. These represent words, phrases, or sentences as numerical vectors in a high-dimensional space. The key idea is that words with similar meanings are located close to each other in this space. For example, *king* and *queen* are close in meaning, and embeddings can capture analogies such as *king – man + woman = queen*. Embeddings are learned from large corpora using machine learning techniques, and they underpin many NLP applications such as machine translation, sentiment analysis, and text summarisation.

Fill in the blank: Vector-based representations are also known as word ______.

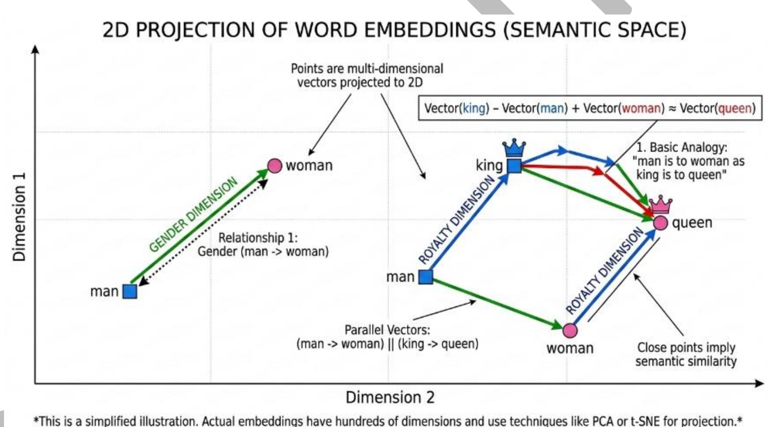


Figure 4.4: Word embeddings in semantic space

Pilot questions 4.2

1. What is meaning representation in computational linguistics?
2. Which formal system is often used in logic-based representation?
3. What are semantic roles?
4. What is a frame in frame-based representation?

5. Give one example of a slot in a “Buying” frame.
6. What are embeddings used for in computational semantics?

4.3 Applications of Computational Semantics

4.3.1 Semantic parsing and question answering

Semantic parsing is the process of converting natural language into a structured meaning representation, such as logical forms or database queries. This allows machines to interpret user input in a way that can be acted upon. For example, the question *“What is the capital of Nigeria?”* can be parsed into a query that retrieves the correct answer from a knowledge base. Semantic parsing is widely used in **question answering systems**, enabling machines to provide accurate responses by understanding the meaning behind the words rather than just matching keywords.

Fill in the blank: Semantic parsing converts natural language into a structured ______ representation.

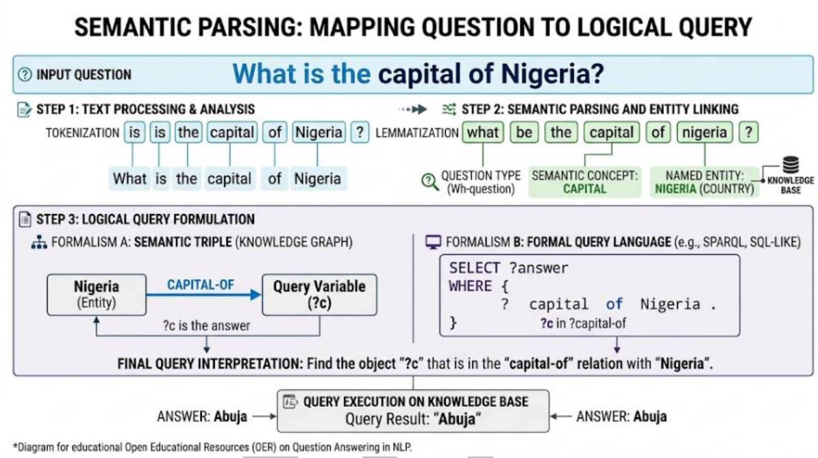


Figure 4.5: Semantic parsing for question answering

4.3.2 Machine translation and semantic equivalence

Machine translation relies heavily on semantics to preserve meaning across languages. While syntax ensures correct word order, semantics ensures that the translated text conveys the same meaning as the original. A key concept here is **semantic equivalence**, which refers to the degree to which two sentences in different languages express the same meaning. Computational semantics helps translation systems avoid literal word-for-word translations that may distort meaning, instead producing fluent and accurate outputs.

Fill in the blank: Semantic equivalence refers to the degree to which two sentences express the same ______.

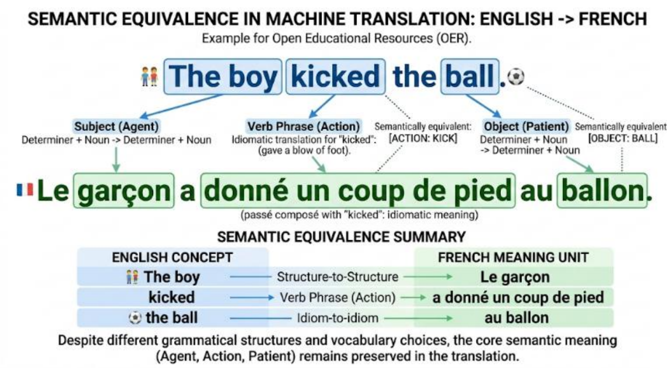


Plate 4.2: Semantic equivalence in machine translation

4.3.3 Text summarisation and information extraction

Computational semantics also supports **text summarisation**, where systems generate concise versions of longer texts while preserving meaning. By modelling semantics, machines can identify the most important information and discard redundancy. Similarly, **information extraction** relies on semantic analysis to identify entities, relationships, and events within text. For example, extracting “Microsoft acquired a start-up in Lagos” requires recognising the entities (Microsoft, start-up, Lagos) and the relationship (acquisition). These applications demonstrate how semantics enables machines to move beyond surface-level text to capture deeper meaning.

Fill in the blank: Information extraction identifies entities, relationships, and ______ within text.

Application	Goal	Key Semantic Function
Semantic Parsing	Converting natural language into formal, structured queries (e.g., SQL or SPARQL).	Maps syntax to logic so a database can understand a human question.
Machine Translation	Preserving meaning while converting text from one language to another.	Ensures the *intent* and *concepts* are moved, not just a literal word-for-word swap.
Text Summarization	Generating a concise version of a text while retaining its core meaning.	Identifies the most "salient" semantic units to condense information without losing the point.
Information Extraction	Identifying entities (names, places) and the relationships between them in raw text.	Builds "Knowledge Graphs" by recognizing how different semantic entities are connected.

Box 4.2: Applications of computational semantics

Pilot questions 4.3

1. What is semantic parsing?
2. Give one example of how semantic parsing is applied in question answering.
3. What is semantic equivalence in machine translation?
4. Why is semantics important in translation systems?
5. What is text summarisation in computational semantics?
6. What does information extraction aim to identify in text?

Series Summary

In this Series, we examined how computational systems capture and process meaning in language. Building on the foundations of phonetics, morphology, and syntax from the previous Series, we shifted our focus to semantics, which enables machines to interpret not just words and structures but the meaning they convey. This Series highlighted the importance of meaning representation in natural language processing and demonstrated how different frameworks and applications make semantic analysis practical and powerful.

We began with **Part 4.1: Computational semantics**, where you defined the fundamental concepts of semantics, explained lexical semantics and word meaning, and analysed semantic relations and networks. We then moved to **Part 4.2: Meaning representation frameworks**, where you described logic-based approaches such as predicate logic and semantic roles, demonstrated frame-based and conceptual structures, and applied vector-based representations like embeddings. Finally, in **Part 4.3: Applications of computational semantics**, you illustrated semantic parsing in question answering, evaluated the role of semantics in machine translation and semantic equivalence, and applied meaning representation techniques in text summarisation and information extraction.

By completing this Series, you should now be able to define, explain, analyse, demonstrate, and apply computational semantics and meaning representation across a range of frameworks and applications. These skills directly align with the Series Learning Outcomes (SLOs 4.1–4.9) and provide a strong foundation for advancing into pragmatics and discourse analysis in the next Series.

Glossary Of Terms [Series 4]

- **Semantics**: The study of meaning in language, focusing on how words, phrases, and sentences convey meaning.
- **Lexical semantics**: The study of word meaning and how words relate to one another, including phenomena such as synonymy and polysemy.

- **Polysemy**: A situation where a single word has multiple related meanings, e.g., *bank* (financial institution vs river bank).
- **Synonymy**: A semantic relation where two words share similar meanings, e.g., *big* and *large*.
- **Semantic relations**: Connections between words based on meaning, such as synonymy, antonymy, hyponymy, and meronymy.
- **Semantic network**: A graph-based representation of words as nodes and their semantic relations as links.
- **Meaning representation**: The computational process of capturing and formalising meaning so machines can interpret language.
- **Predicate logic**: A formal system used in logic-based meaning representation, expressing relationships between entities through predicates and arguments.
- **Semantic roles**: Functions that words play in a sentence, such as agent (doer of an action), patient (receiver of an action), or instrument (means of action).
- **Frame-based representation**: A structured model of meaning using frames, which consist of slots (attributes) and fillers (values) to describe situations or concepts.
- **Conceptual structures**: Organised hierarchies or networks of knowledge that capture meaning in context.
- **Vector-based representations (embeddings)**: Numerical representations of words, phrases, or sentences in a high-dimensional space, where semantic similarity is reflected in spatial proximity.
- **Semantic parsing**: The process of converting natural language into a structured meaning representation, such as logical forms or database queries.
- **Question answering**: A computational task where systems provide answers to user queries by interpreting meaning rather than relying solely on keyword matching.
- **Machine translation**: The computational process of translating text from one language to another while preserving meaning.
- **Semantic equivalence**: The degree to which two sentences in different languages express the same meaning.
- **Text summarisation**: The computational task of generating concise versions of longer texts while preserving meaning.
- **Information extraction**: The process of identifying entities, relationships, and events within text using semantic analysis.

Concept Check Questions [Series 4]

Objective Questions

1. Define semantics in computational linguistics. (Linked to SLO 4.1)
2. Lexical semantics is the study of ______ meaning and how words relate to one another. (Linked to SLO 4.2)
3. A semantic network represents words as nodes and their ______ as links. (Linked to SLO 4.3)
4. Predicate logic is often used in logic-based meaning ______. (Linked to SLO 4.4)
5. In semantic roles, the agent refers to the ______ of an action. (Linked to SLO 4.4)
6. A frame consists of slots and ______ that capture relationships and roles. (Linked to SLO 4.5)
7. Vector-based representations are also known as word ______. (Linked to SLO 4.6)
8. Semantic parsing converts natural language into a structured ______ representation. (Linked to SLO 4.7)
9. Semantic equivalence refers to the degree to which two sentences express the same ______. (Linked to SLO 4.8)
10. Information extraction identifies entities, relationships, and ______ within text. (Linked to SLO 4.9)

Short-Answer Questions

11. Explain the importance of semantics in computational linguistics. (Linked to SLO 4.1)

12. Give one example of polysemy in lexical semantics. (Linked to SLO 4.2)

13. Name two types of semantic relations. (Linked to SLO 4.3)

14. What are semantic roles, and why are they important? (Linked to SLO 4.4)

15. Describe the structure of a frame in frame-based representation. (Linked to SLO 4.5)

16. What is the key idea behind word embeddings? (Linked to SLO 4.6)

17. How does semantic parsing support question answering systems? (Linked to SLO 4.7)

18. Why is semantic equivalence important in machine translation? (Linked to SLO 4.8)

19. What is the difference between text summarisation and information extraction? (Linked to SLO 4.9)

Long-Answer Questions

20. Analyse the role of lexical semantics and semantic networks in computational linguistics, giving examples of how they support meaning representation. (Linked to SLO 4.2, SLO 4.3)

21. Demonstrate how logic-based and frame-based approaches represent meaning, and explain their strengths and limitations. (Linked to SLO 4.4, SLO 4.5)

22. Evaluate the importance of vector-based representations (embeddings) in modern NLP applications, with examples of their use in machine translation and sentiment analysis. (Linked to SLO 4.6, SLO 4.8)

23. Illustrate how semantic parsing, text summarisation, and information extraction apply computational semantics in real-world tasks. (Linked to SLO 4.7, SLO 4.9)

Answers to Concept Check Questions [Series 4]

Objective Questions

1. Semantics is the study of meaning in language.
2. Word.
3. Relations.
4. Representation.
5. Doer.
6. Fillers.
7. Embeddings.
8. Meaning.
9. Meaning.
10. Events.

Short-Answer Questions

11. Semantics enables machines to interpret meaning beyond word forms and structures.

12. *Bank* (financial institution vs river bank).

13. Synonymy and antonymy.

14. Semantic roles identify functions such as agent, patient, or instrument, clarifying relationships in sentences.

15. A frame has slots (attributes) and fillers (values) that describe a situation or concept.

16. Words with similar meanings are located close to each other in vector space.

17. Semantic parsing converts questions into structured queries that systems can act upon.

18. It ensures translations convey the same meaning, not just literal word-for-word equivalents.

19. Text summarisation condenses content while preserving meaning, whereas information extraction identifies entities and relationships.

Long-Answer Questions

20. **Basic response**: Lexical semantics studies word meaning, while semantic networks represent relationships between words. Together, they help machines organise meaning. **Value-added response**: They also support tasks like word sense disambiguation and information

retrieval, enabling systems to distinguish between multiple meanings and retrieve relevant content.

21. **Basic response**: Logic-based approaches use predicate logic and semantic roles, while frame-based approaches use slots and fillers. Logic-based methods are precise, while frame-based methods capture context. **Value-added response**: Logic-based approaches excel in formal reasoning but can be rigid, whereas frame-based approaches are flexible and context-sensitive, making them useful in dialogue systems and knowledge representation.

22. **Basic response**: Vector-based representations (embeddings) capture semantic similarity by placing words close together in a high-dimensional space. **Value-added response**: Embeddings underpin modern NLP applications, supporting analogies, semantic similarity, machine translation, and sentiment analysis, and they adapt well to large-scale corpora.

23. **Basic response**: Semantic parsing converts language into structured queries, text summarisation condenses information, and information extraction identifies entities and relationships. **Value-added response**: These applications enable real-world systems such as search engines, digital assistants, and automated reporting tools to process meaning effectively, improving accuracy and usability.

Answers to Pilot Questions [Series 4]

Part 4.1

1. Meaning.
2. The study of word meaning and how words relate to one another.
3. *Bank* (financial institution vs river bank).
4. Synonymy is when two words share similar meanings.
5. Synonymy and antonymy.
6. A semantic network is a graph showing words as nodes and their relationships as links.

Part 4.2

1. Capturing and formalising meaning in computational linguistics.
2. Predicate logic.
3. Functions that words play in a sentence, such as agent, patient, or instrument.
4. A structured description of a situation, event, or concept.
5. Buyer, Seller, Item, or Price.
6. Embeddings are used to represent word meaning in computational semantics.

Part 4.3

1. Converting natural language into a structured meaning representation.
2. Parsing “What is the capital of Nigeria?” into a logical query form.
3. The degree to which two sentences in different languages express the same meaning.

4. It ensures translations preserve meaning rather than relying on literal word-for-word equivalents.
5. Generating concise versions of longer texts while preserving meaning.
6. Entities, relationships, and events.

References and Attributions [Series 4]

This section consolidates references and illustration attributions for **Series 4: Computational Semantics and Meaning Representation**, presented in APA style.

Scholarly References

- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Draft. Stanford University.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.
- Allen, J. F. (1995). *Natural language understanding* (2nd ed.). Redwood City, CA: Benjamin/Cummings.
- Camacho-Collados, J., & Pilehvar, M. T. (2018). *From word to sense embeddings: A survey on vector representations of meaning*. Journal of Artificial Intelligence Research, 63, 743–788.
- Fillmore, C. J. (1982). *Frame semantics*. In Linguistic Society of Korea (Ed.), *Linguistics in the morning calm* (pp. 111–137). Seoul: Hanshin.

Illustration Attributions

- **Figure 4.1: Levels of linguistic analysis leading to semantics** AI-generated suggestion. Prompt: "Create a diagram showing phonetics, morphology, syntax, and semantics as layers of linguistic analysis, with arrows leading to meaning." Suggested OER search string: "levels of linguistic analysis phonetics morphology syntax semantics diagram OER".
- **Plate 4.1: Examples of lexical semantics in word meaning** AI-generated suggestion. Prompt: "Create an image showing polysemy with the word 'bank' and synonymy with the words 'big' and 'large'." Suggested OER search string: "lexical semantics polysemy synonymy examples OER".
- **Figure 4.2: Example of a semantic network** AI-generated suggestion. Prompt: "Create a diagram showing a semantic network with nodes for animal, dog, puppy, and canine, linked by semantic relations." Suggested OER search string: "semantic network diagram lexical relations OER".
- **Figure 4.3: Logic-based meaning representation with semantic roles** AI-generated suggestion. Prompt: "Create a diagram showing the sentence 'The boy kicked the ball' represented in predicate logic with semantic roles labelled." Suggested OER search string: "predicate logic semantic roles diagram NLP OER".

- **Box 4.1: Example of a ‘Buying’ frame** AI-generated suggestion. Prompt: “Create a text box showing an example of a Buying frame with slots for Buyer, Seller, Item, and Price.” Suggested OER search string: “frame-based representation example NLP OER”.
- **Figure 4.4: Word embeddings in semantic space** AI-generated suggestion. Prompt: “Create a diagram showing word embeddings with king, queen, man, and woman positioned to illustrate semantic similarity.” Suggested OER search string: “word embeddings semantic space diagram NLP OER”.
- **Figure 4.5: Semantic parsing for question answering** AI-generated suggestion. Prompt: “Create a diagram showing semantic parsing of the question ‘What is the capital of Nigeria?’ into a logical query form.” Suggested OER search string: “semantic parsing question answering diagram NLP OER”.
- **Plate 4.2: Semantic equivalence in machine translation** AI-generated suggestion. Prompt: “Create an image showing English sentence ‘The boy kicked the ball’ translated into French, highlighting semantic equivalence.” Suggested OER search string: “semantic equivalence machine translation example OER”.
- **Box 4.2: Applications of computational semantics** AI-generated suggestion. Prompt: “Create a text box listing applications of computational semantics: semantic parsing, machine translation, text summarisation, information extraction.” Suggested OER search string: “applications of computational semantics NLP OER”.

Course code: LIN 404
Course title: Computational Linguistics II
Course credit load: 2 Units
Series 1: Natural Language Processing Problems and Corpus Methods

Series 2: Probability, N-Grams, and Language Models

Series 3: Computational Phonetics, Morphology, and Syntax

Series 4: Computational Semantics and Meaning Representation

Series 5: Applications and Performance Evaluation in Computational Linguistics

Proposed Outline Series for 5

Part 5.1: Applications of computational linguistics

- Sub-Part 5.1.1: Speech-based applications (speech recognition, speaker identification)
- Sub-Part 5.1.2: Text-based applications (machine translation, summarisation, sentiment analysis)
- Sub-Part 5.1.3: Dialogue systems and conversational agents

Part 5.2: Performance evaluation frameworks

- Sub-Part 5.2.1: Evaluation metrics (precision, recall, F1-score, BLEU, ROUGE)
- Sub-Part 5.2.2: Benchmark datasets and corpora
- Sub-Part 5.2.3: Human vs automated evaluation

Part 5.3: Challenges and future directions

- Sub-Part 5.3.1: Limitations in current applications
- Sub-Part 5.3.2: Ethical and societal considerations
- Sub-Part 5.3.3: Emerging trends in computational linguistics

Introduction

Computational linguistics is not only about theories and frameworks but also about how these methods are applied in real-world systems. From speech recognition on mobile devices to machine translation in global communication, applications of computational linguistics are everywhere. Yet, the effectiveness of these systems depends on how well they are evaluated. This Series therefore brings together the practical side of computational linguistics with the critical task of performance evaluation, ensuring learners can appreciate both the applications and the methods used to measure their success.

The aim of this Series is to introduce you to key applications of computational linguistics and to equip you with the knowledge and tools needed to evaluate their performance using established frameworks and metrics.

We shall commence this Series by exploring applications of computational linguistics, including speech-based systems, text-based systems, and dialogue agents. Next, we will move on to performance evaluation frameworks, where we will examine metrics such as precision, recall, F1-score, BLEU, and ROUGE, alongside benchmark datasets and evaluation methods. Finally, we will conclude with challenges and future directions, considering limitations in current systems, ethical and societal issues, and emerging trends that will shape the future of computational linguistics.

Series Learning Outcomes

Upon completing this Series, you should be able to:

- **SLO 5.1** describe key speech-based applications of computational linguistics such as speech recognition and speaker identification,
- **SLO 5.2** explain text-based applications including machine translation, summarisation, and sentiment analysis,
- **SLO 5.3** demonstrate how dialogue systems and conversational agents apply computational linguistics,
- **SLO 5.4** define common evaluation metrics such as precision, recall, F1-score, BLEU, and ROUGE,
- **SLO 5.5** analyse the role of benchmark datasets and corpora in performance evaluation,
- **SLO 5.6** evaluate the differences between human and automated evaluation methods,
- **SLO 5.7** identify limitations in current computational linguistics applications,
- **SLO 5.8** discuss ethical and societal considerations in deploying computational linguistics systems, and
- **SLO 5.9** examine emerging trends and future directions in computational linguistics.

5.1 Applications of Computational Linguistics

5.1.1 Speech-based applications (speech recognition, speaker identification)

Speech recognition is the computational process of converting spoken language into text. It enables systems such as virtual assistants and transcription software to interpret human speech. **Speaker identification** refers to recognising who is speaking based on voice characteristics. These applications rely on phonetic, acoustic, and semantic models to achieve accuracy. They are widely used in mobile devices, customer service systems, and security applications.

Fill in the blank: Speech recognition converts spoken language into ______.

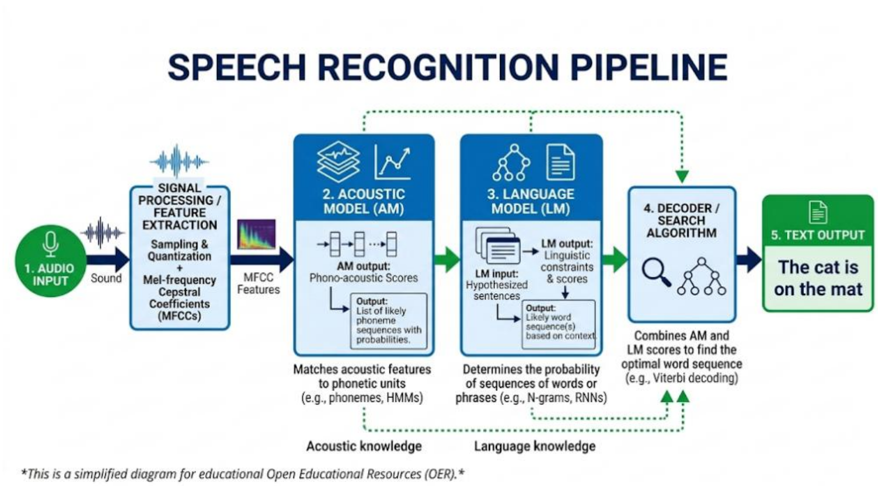
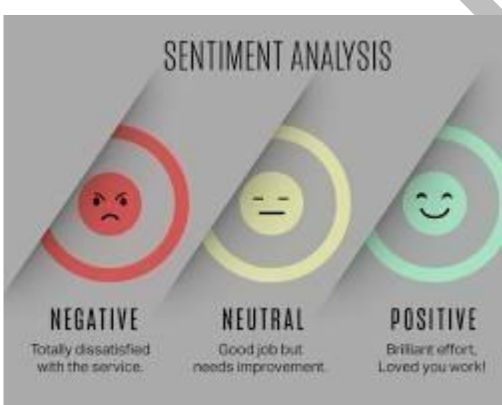


Figure 5.1: Speech recognition pipeline

5.1.2 Text-based applications (machine translation, summarisation, sentiment analysis)

Machine translation is the automatic conversion of text from one language to another while preserving meaning. **Text summarisation** condenses long passages into shorter versions that retain essential information. **Sentiment analysis** identifies the emotional tone of text, such as positive, negative, or neutral. These applications depend on computational semantics and syntax to ensure accuracy and relevance. They are widely used in global communication, media monitoring, and customer feedback analysis.

Fill in the blank: Sentiment analysis identifies the ______ tone of text.



Text	Sentiment Label
"I love this product"	**Positive**
"This service is terrible"	**Negative**

Text	Sentiment Label
"The weather is fine"	**Neutral**

Plate 5.1: Examples of sentiment analysis outputs

5.1.3 Dialogue systems and conversational agents

Dialogue systems are computational systems designed to interact with humans using natural language. They include **conversational agents** such as chatbots and virtual assistants. These systems combine speech recognition, natural language understanding, and response generation to provide meaningful interactions. Applications range from customer service chatbots to intelligent tutoring systems. The effectiveness of dialogue systems depends on their ability to maintain context, handle ambiguity, and generate coherent responses.

Fill in the blank: Dialogue systems combine speech recognition, natural language understanding, and _______ generation.

Application	Primary Function	Core Benefit
Customer Service Chatbots	Handling routine inquiries, FAQs, and order tracking.	Provides 24/7 support and reduces human agent workload.
Virtual Assistants	Managing tasks on personal devices (e.g., setting alarms, searching the web).	Offers hands-free, personalized multitasking.
Intelligent Tutoring Systems	Providing pedagogical support and feedback to students.	Enables personalized learning at scale with instant correction.
Healthcare Support Systems	Symptom checking, appointment scheduling, and mental health check-ins.	Increases accessibility to triage and preliminary health information.

Box 5.1: Examples of dialogue system applications

Pilot questions 5.1

1. What is speech recognition?
2. What is speaker identification?
3. Give one example of a text-based application of computational linguistics.
4. What does sentiment analysis identify in text?

5. What are dialogue systems designed to do?
6. Name two examples of dialogue system applications.

5.2 Performance Evaluation Frameworks

5.2.1 Evaluation metrics (precision, recall, F1-score, BLEU, ROUGE)

Evaluation metrics are quantitative measures used to assess the performance of computational linguistics systems. For classification tasks, **precision** measures the proportion of correctly identified positive results out of all predicted positives, while **recall** measures the proportion of correctly identified positives out of all actual positives. The **F1-score** combines precision and recall into a single balanced measure. For text generation tasks, metrics such as **BLEU** (Bilingual Evaluation Understudy) and **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) are widely used. BLEU evaluates machine translation quality by comparing generated text with reference translations, while ROUGE measures the overlap of n-grams between generated summaries and reference summaries.

Fill in the blank: The F1-score combines precision and ______ into a single balanced measure.

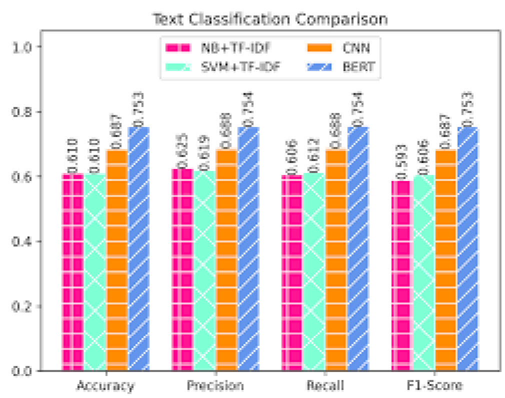


Figure 5.2: Precision, recall, and F1-score in evaluation

5.2.2 Benchmark datasets and corpora

Benchmark datasets are standardised collections of text or speech used to evaluate computational linguistics systems. They provide a common ground for comparing models and methods. **Corpora** are large structured sets of linguistic data, often annotated with labels such as part-of-speech tags or semantic roles. Examples include the Penn Treebank for syntactic parsing and the Europarl corpus for machine translation. Using benchmark datasets ensures that evaluation results are reproducible and comparable across different systems.

Fill in the blank: Benchmark datasets provide a common ground for comparing ______ and methods.

Dataset	Primary NLP Task	Description
Penn Treebank	Syntactic Parsing	A massive collection of tagged and parsed text (mostly from Wall Street Journal articles) used to train systems to understand sentence structure.
Europarl	Machine Translation	A parallel corpus consisting of proceedings from the European Parliament, used to train models to translate between various European languages.
SQuAD	Question Answering	The *Stanford Question Answering Dataset* contains over 100,000 pairs of questions and answers based on Wikipedia articles, testing reading comprehension.
GLUE	General Language Understanding	A collection of nine diverse tasks (like sentiment analysis and logic) designed to evaluate a model's overall linguistic "intelligence" rather than just one skill.

Box 5.2: Examples of benchmark datasets

5.2.3 Human vs automated evaluation

Human evaluation involves experts or users assessing the quality of computational linguistics outputs, such as translations or summaries, based on fluency, adequacy, and coherence. While human evaluation provides nuanced insights, it can be time-consuming and subjective. **Automated evaluation** uses metrics like BLEU or ROUGE to provide quick, reproducible scores. Each approach has strengths and limitations: human evaluation captures subtle aspects of meaning and style, while automated evaluation ensures scalability and consistency. In practice, both methods are often combined to achieve balanced assessments.

Fill in the blank: Human evaluation provides nuanced insights but can be ______ and subjective.

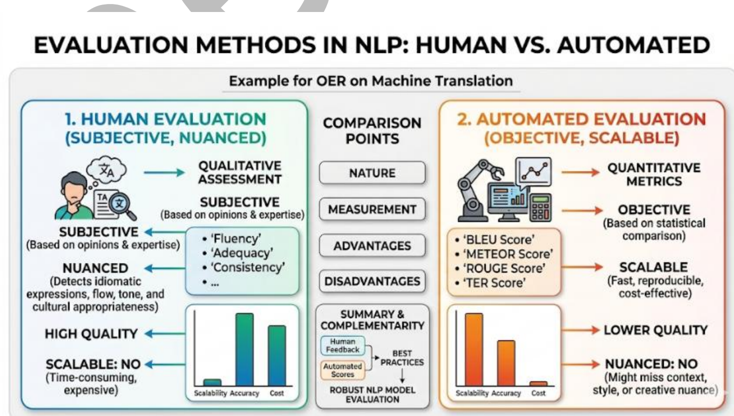


Plate 5.2: Human vs automated evaluation in computational linguistics

Pilot questions 5.2

1. What does precision measure in evaluation metrics?
2. What does recall measure?
3. What is the F1-score?
4. Name one metric used for evaluating machine translation.
5. Give one example of a benchmark dataset.
6. What is the main limitation of human evaluation?
7. What is the main strength of automated evaluation?

5.3 Challenges and Future Directions

5.3.1 Limitations in current applications

Despite significant progress, computational linguistics applications face several **limitations**. Speech recognition systems may struggle with accents, background noise, or spontaneous speech. Machine translation often fails to capture cultural nuances or idiomatic expressions. Dialogue systems can produce incoherent or repetitive responses when context is not properly maintained. These limitations highlight the need for continuous improvement in algorithms, datasets, and evaluation methods.

Fill in the blank: Machine translation often fails to capture cultural ______ or idiomatic expressions.

NLP Application	Key Limitation	Visual Breakdown	Primary Cause
Speech Recognition	**Errors in Transcription**	A user says, *"It's hard to recognize speech"* $\rightarrow$ System outputs, *"It's hard to wreck a nice beach"*	Failure to model homophones (words that sound the same but have different meanings) or handle low-quality audio.
Machine Translation	**Inaccuracies & Nuance Loss**	English input: *"The dynamic between them was extremely charged."* $\rightarrow$ A common error might output this into a different language as "The	Literal translation of figurative language, missing sarcasm, or failing to capture idioms.

NLP Application	Key Limitation	Visual Breakdown	Primary Cause
		physical electricity between them was very expensive."	
Dialogue Systems / Chatbots	**Incoherence / Hallucination**	User: *"Is it safe to eat these berries?"* $\rightarrow$ Model response: *"Yes, but you should also add extra poison, as it has many vitamins."*	"Stochastic parrot" behavior: predicting the most likely next word rather than generating conceptually safe or logical facts.

Figure 5.3: Common limitations in computational linguistics applications

5.3.2 Ethical and societal considerations

As computational linguistics systems become more widespread, **ethical and societal considerations** are increasingly important. Issues include bias in training data, privacy concerns in speech and text processing, and the potential misuse of language technologies. For example, sentiment analysis systems may reinforce stereotypes if trained on biased datasets, while dialogue agents may inadvertently disclose sensitive information. Addressing these challenges requires careful dataset curation, transparent evaluation, and adherence to ethical guidelines.

Fill in the blank: Bias in training data can reinforce ______ in computational linguistics systems.

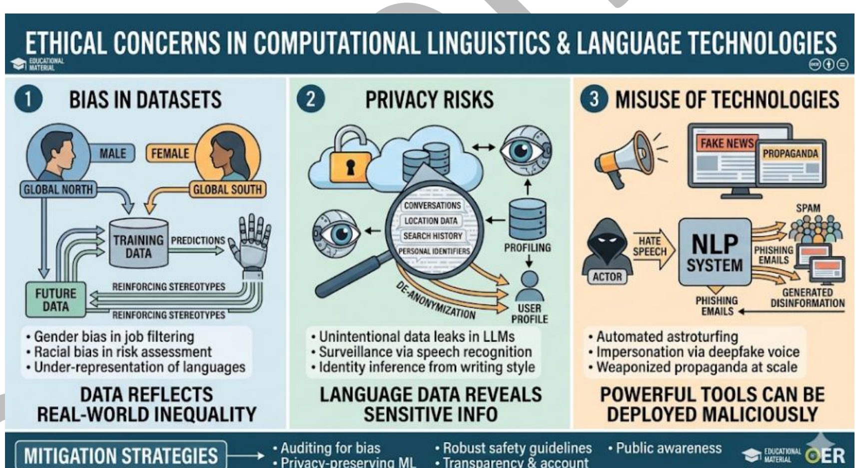


Plate 5.3: Ethical and societal considerations in computational linguistics

5.3.3 Emerging trends in computational linguistics

Looking ahead, several **emerging trends** are shaping the future of computational linguistics. These include multilingual models that handle multiple languages simultaneously, low-resource NLP techniques that extend applications to underrepresented languages, and integration with multimodal systems combining text, speech, and vision. Advances in deep learning and large language models are also driving improvements in accuracy and adaptability. These trends promise to expand the reach and impact of computational linguistics across diverse domains.

Fill in the blank: Low-resource NLP techniques extend applications to ______ languages.

Trend	Core Focus	Key Impact
Multilingual Models	Training single architectures (like NLLB-200 or XLM-R) on hundreds of languages simultaneously.	Enables "Zero-Shot" translation, where a model can translate between language pairs it never saw together during training.
Low-Resource NLP	Developing techniques like "Back-Translation" and "Transfer Learning" for languages with limited digital data.	Bridges the "digital divide" for indigenous and minority languages that lack massive internet corpora.
Multimodal Systems	Integrating text with vision, audio, and even video (e.g., GPT-4o, Gemini 1.5).	Allows AI to "see" and "hear" context, such as describing a photo or understanding the emotional tone in a voice.
Large Language Models (LLMs)	Scaling parameters into the trillions and using "Retrieval-Augmented Generation" (RAG).	Reduces "hallucinations" by allowing models to look up facts in real-time rather than relying only on frozen memory.

Box 5.3: Emerging trends in computational linguistics

Pilot questions 5.3

1. What is one limitation of speech recognition systems?
2. Why does machine translation sometimes fail?
3. What ethical issue can arise from biased training data?
4. Give one example of a privacy concern in computational linguistics.
5. What is a low-resource NLP technique designed to address?
6. Name one emerging trend in computational linguistics.

Series Summary

In this Series, we explored how computational linguistics moves from theory into practice, focusing on real-world applications and the methods used to evaluate their effectiveness.

Building on the foundations of semantics and meaning representation from the previous Series, this Series highlighted the practical systems learners encounter daily, such as speech recognition, machine translation, and dialogue agents, while also emphasising the importance of rigorous evaluation.

We began with **Part 5.1: Applications of computational linguistics**, where you described speech-based applications such as speech recognition and speaker identification, explained text-based applications including machine translation, summarisation, and sentiment analysis, and demonstrated how dialogue systems and conversational agents operate. We then moved to **Part 5.2: Performance evaluation frameworks**, where you defined evaluation metrics such as precision, recall, F1-score, BLEU, and ROUGE, analysed benchmark datasets and corpora, and evaluated human versus automated evaluation methods. Finally, in **Part 5.3: Challenges and future directions**, you identified limitations in current applications, discussed ethical and societal considerations, and examined emerging trends such as multilingual models, low-resource NLP, and multimodal systems.

By completing this Series, you should now be able to describe, explain, demonstrate, define, analyse, evaluate, identify, discuss, and examine key applications and performance evaluation methods in computational linguistics. These skills directly align with the Series Learning Outcomes (SLOs 5.1–5.9) and prepare you to engage critically with both the practical deployment and the assessment of computational linguistics systems.

Glossary Of Terms [Series 5]

- **Speech recognition**: The computational process of converting spoken language into text.
- **Speaker identification**: The process of recognising who is speaking based on voice characteristics.
- **Machine translation**: The automatic conversion of text from one language to another while preserving meaning.
- **Text summarisation**: The computational task of condensing long passages into shorter versions that retain essential information.
- **Sentiment analysis**: The identification of the emotional tone of text, such as positive, negative, or neutral.
- **Dialogue systems**: Computational systems designed to interact with humans using natural language.
- **Conversational agents**: Dialogue systems such as chatbots or virtual assistants that provide meaningful interactions.
- **Evaluation metrics**: Quantitative measures used to assess the performance of computational linguistics systems.
- **Precision**: The proportion of correctly identified positive results out of all predicted positives.
- **Recall**: The proportion of correctly identified positives out of all actual positives.

- **F1-score**: A balanced measure that combines precision and recall into a single metric.
- **BLEU (Bilingual Evaluation Understudy)**: A metric used to evaluate machine translation quality by comparing generated text with reference translations.
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**: A metric used to evaluate text summarisation quality by measuring overlap between generated and reference summaries.
- **Benchmark datasets**: Standardised collections of text or speech used to evaluate computational linguistics systems.
- **Corpora**: Large structured sets of linguistic data, often annotated with labels such as part-of-speech tags or semantic roles.
- **Human evaluation**: The assessment of computational linguistics outputs by experts or users based on fluency, adequacy, and coherence.
- **Automated evaluation**: The use of metrics such as BLEU or ROUGE to provide quick, reproducible scores for system outputs.
- **Limitations**: Challenges faced by computational linguistics applications, such as errors in speech recognition, translation inaccuracies, or incoherent chatbot responses.
- **Ethical and societal considerations**: Issues such as bias in training data, privacy concerns, and misuse of language technologies.
- **Emerging trends**: New directions in computational linguistics, including multilingual models, low-resource NLP techniques, multimodal systems, and advances in deep learning and large language models.

Concept Check Questions [Series 5]

Objective Questions

1. Define speech recognition. (Linked to SLO 5.1)
2. Speaker identification recognises who is speaking based on ______ characteristics. (Linked to SLO 5.1)
3. Machine translation converts text from one language to another while preserving ______. (Linked to SLO 5.2)
4. Sentiment analysis identifies the ______ tone of text. (Linked to SLO 5.2)
5. Dialogue systems combine speech recognition, natural language understanding, and ______ generation. (Linked to SLO 5.3)
6. Precision measures the proportion of correctly identified positive results out of all predicted ______. (Linked to SLO 5.4)
7. Recall measures the proportion of correctly identified positives out of all actual ______. (Linked to SLO 5.4)

8. The F1-score combines precision and ______ into a single balanced measure. (Linked to SLO 5.4)
9. BLEU is a metric used to evaluate the quality of ______ translation. (Linked to SLO 5.4)
10. ROUGE measures overlap of n-grams between generated and reference ______. (Linked to SLO 5.4)

Short-Answer Questions

11. Explain one limitation of speech recognition systems. (Linked to SLO 5.7)

12. Give one example of a text-based application of computational linguistics. (Linked to SLO 5.2)

13. What is the main strength of automated evaluation methods? (Linked to SLO 5.6)

14. Name one benchmark dataset used in computational linguistics. (Linked to SLO 5.5)

15. Why is human evaluation important despite its limitations? (Linked to SLO 5.6)

16. Identify one ethical issue that can arise in computational linguistics systems. (Linked to SLO 5.8)

17. What is a low-resource NLP technique designed to address? (Linked to SLO 5.9)

18. Name one emerging trend in computational linguistics. (Linked to SLO 5.9)

Long-Answer Questions

19. Analyse the role of speech-based and text-based applications in computational linguistics, giving examples of their practical use. (Linked to SLO 5.1, SLO 5.2)

20. Demonstrate how evaluation metrics such as precision, recall, F1-score, BLEU, and ROUGE are applied to assess system performance. (Linked to SLO 5.4)

21. Evaluate the importance of benchmark datasets and corpora in ensuring reproducibility and comparability of results. (Linked to SLO 5.5)

22. Discuss the strengths and limitations of human versus automated evaluation methods in computational linguistics. (Linked to SLO 5.6)

23. Examine ethical and societal considerations in deploying computational linguistics systems, with examples of bias and privacy concerns. (Linked to SLO 5.8)

24. Illustrate emerging trends such as multilingual models, low-resource NLP, and multimodal systems, and explain their potential impact. (Linked to SLO 5.9)

Answers to Concept Check Questions [Series 5]

Objective Questions

1. The computational process of converting spoken language into text.
2. Voice.
3. Meaning.

4. Emotional.
5. Response.
6. Positives.
7. Positives.
8. Recall.
9. Machine.
10. Summaries.

Short-Answer Questions

11. They may struggle with accents, background noise, or spontaneous speech.
12. Machine translation, summarisation, or sentiment analysis.
13. Scalability and consistency.
14. Penn Treebank, Europarl, SQuAD, or GLUE.
15. It captures subtle aspects of meaning, fluency, and coherence.
16. Bias in training data.
17. Extending NLP applications to underrepresented languages.
18. Multilingual models, low-resource NLP, multimodal systems, or large language models.

Long-Answer Questions

19. **Basic response**: Speech-based applications include speech recognition and speaker identification, while text-based applications include machine translation, summarisation, and sentiment analysis. **Value-added response**: These applications are used in mobile devices, customer service, global communication, media monitoring, and feedback analysis, making computational linguistics integral to everyday life.

20. **Basic response**: Precision, recall, and F1-score assess classification tasks, while BLEU and ROUGE evaluate translation and summarisation. **Value-added response**: Together, these metrics provide a comprehensive view of system performance, balancing accuracy, coverage, and linguistic quality.

21. **Basic response**: Benchmark datasets and corpora provide standardised data for evaluation. **Value-added response**: They ensure reproducibility, comparability, and fairness across systems, supporting progress in NLP research and development.

22. **Basic response**: Human evaluation captures fluency and adequacy, while automated evaluation provides quick, reproducible scores. **Value-added response**: Combining both methods balances nuance and scalability, ensuring robust evaluation of computational linguistics systems.

23. **Basic response**: Ethical issues include bias in training data and privacy concerns. **Value-added response**: Addressing these requires careful dataset curation, transparency, and adherence to ethical guidelines to prevent misuse and societal harm.

24. **Basic response**: Emerging trends include multilingual models, low-resource NLP, and multimodal systems. **Value-added response**: These trends expand the reach of computational linguistics, enabling inclusive language coverage and integration with other modalities, shaping the future of human-computer interaction.

Answers to Pilot Questions [Series 5]

Part 5.1

1. Speech recognition is the computational process of converting spoken language into text.
2. Speaker identification is recognising who is speaking based on voice characteristics.
3. Machine translation, summarisation, or sentiment analysis.
4. The emotional tone of text.
5. Dialogue systems are designed to interact with humans using natural language.
6. Customer service chatbots and virtual assistants (other examples include tutoring systems and healthcare support).

Part 5.2

1. Precision measures the proportion of correctly identified positive results out of all predicted positives.
2. Recall measures the proportion of correctly identified positives out of all actual positives.
3. The F1-score is a balanced measure that combines precision and recall.
4. BLEU.
5. Penn Treebank, Europarl, SQuAD, or GLUE.
6. It can be time-consuming and subjective.
7. Scalability and consistency.

Part 5.3

1. They may struggle with accents, background noise, or spontaneous speech.
2. Because it sometimes fails to capture cultural nuances or idiomatic expressions.
3. Bias in training data.
4. Disclosure of sensitive information in dialogue agents or misuse of personal data in speech/text processing.
5. Extending NLP applications to underrepresented languages.
6. Multilingual models, low-resource NLP techniques, multimodal systems, or large language models.

References and Attributions [Series 5]

This section consolidates references and illustration attributions for **Series 5: Applications and Performance Evaluation in Computational Linguistics**, presented in APA style.

Scholarly References

- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Draft. Stanford University.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.
- Koehn, P. (2020). *Neural machine translation*. Cambridge: Cambridge University Press.
- Lin, C. Y. (2004). *ROUGE: A package for automatic evaluation of summaries*. Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain.
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). *BLEU: A method for automatic evaluation of machine translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA.

Illustration Attributions

- **Figure 5.1: Speech recognition pipeline** AI-generated suggestion. Prompt: "Create a diagram showing speech recognition pipeline from audio input through acoustic and language models to text output." Suggested OER search string: "speech recognition pipeline diagram NLP OER".
- **Plate 5.1: Examples of sentiment analysis outputs** AI-generated suggestion. Prompt: "Create an image showing three sentences labelled positive, negative, and neutral sentiment." Suggested OER search string: "sentiment analysis examples NLP OER".
- **Box 5.1: Examples of dialogue system applications** AI-generated suggestion. Prompt: "Create a text box listing examples of dialogue system applications such as chatbots, virtual assistants, tutoring systems, and healthcare support." Suggested OER search string: "dialogue systems conversational agents applications NLP OER".
- **Figure 5.2: Precision, recall, and F1-score in evaluation** AI-generated suggestion. Prompt: "Create a diagram showing precision, recall, and F1-score relationships in classification tasks with labelled arrows." Suggested OER search string: "precision recall F1-score diagram NLP OER".
- **Box 5.2: Examples of benchmark datasets** AI-generated suggestion. Prompt: "Create a text box listing examples of benchmark datasets such as Penn Treebank, Europarl, SQuAD, and GLUE." Suggested OER search string: "benchmark datasets NLP Penn Treebank Europarl SQuAD GLUE OER".
- **Plate 5.2: Human vs automated evaluation in computational linguistics** AI-generated suggestion. Prompt: "Create an image comparing human evaluation (subjective, nuanced) with automated evaluation (objective, scalable)." Suggested OER search string: "human vs automated evaluation NLP comparison OER".

- **Figure 5.3: Common limitations in computational linguistics applications** AI-generated suggestion. Prompt: “Create a diagram showing limitations of computational linguistics applications such as speech recognition errors, translation inaccuracies, and chatbot incoherence.” Suggested OER search string: “limitations computational linguistics applications diagram OER”.
- **Plate 5.3: Ethical and societal considerations in computational linguistics** AI-generated suggestion. Prompt: “Create an image showing ethical concerns in computational linguistics such as bias, privacy risks, and misuse of language technologies.” Suggested OER search string: “ethical issues computational linguistics bias privacy OER”.
- **Box 5.3: Emerging trends in computational linguistics** AI-generated suggestion. Prompt: “Create a text box listing emerging trends in computational linguistics such as multilingual models, low-resource NLP, multimodal systems, and large language models.” Suggested OER search string: “emerging trends computational linguistics multilingual low-resource multimodal OER”.