NATURAL LANGUAGE PROCESSING

Unit II — Text Representation & Feature Engineering

Topics Covered in this Unit

Corpus Linguistics: Types of Corpora, Annotation Schemes (POS, IOB/BIO NER) & Annotation Tools
Text Preprocessing Pipeline: Tokenization, Segmentation, Normalization, True-Casing & Stopwords
Stemming vs. Lemmatization: The Porter Stemmer Algorithm Rules vs. Morphological Dictionary Lookup
Handling Out-of-Vocabulary (OOV) Words: Subword Tokenization (BPE, WordPiece, SentencePiece)
Word Representation Paradigms: One-Hot Vectors, Distributional Semantics & Co-occurrence Matrices
Sparse vs. Dense Representations: High Dimensionality/Orthogonality vs. Distributed Embeddings
Lexical Semantic Resources: WordNet Structure, Synsets, Hypernymy/Hyponymy & Semantic Similarity
Feature Engineering & K-Grams: Character-level vs. Word-level K-grams in Spelling, IR & Text Models
Feature Extraction & Selection: Bag of Words (BoW), DTM, TF-IDF Math, Chi-Square (X2) & Mutual Info
Word Embeddings: Word2Vec (CBOW & Skip-Gram SGNS Math), GloVe Log-Bilinear Matrix Factorization
High-Yield University Exam Question Bank with Detailed Model Calculations & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & Data Science Students

COURSE SYLLABUS & MAPPING

[UNIT II SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit II. Verify with your university curriculum mapping.

Unit II Syllabus Breakdown:

Unit II - Text Representation & Feature Engineering: Corpus: Types of corpora, Annotation and formats, POS and NER annotation schemes, Annotation tools. Text Preprocessing: Tokenization, Sentence segmentation, Normalization, true casing, Stemming and lemmatization, Text Cleaning, stemming (Porter Stemmer algorithm), Stopword Removal, Handling OOV (Out-of-vocabulary) Words: Unknown tokens, subword techniques, and challenges with NLP systems. Word Representation: One-hot, distributional semantics, co-occurrence matrices, and information extraction. Sparse vs Dense Representations: Limitations of sparse vectors (high dimensionality, sparsity) and advantages of dense embeddings, Lexical Resources: WordNet, synsets, semantic relations. Feature Engineering: Concept of feature engineering, types of features, and K-gram models: Character-level and word-level k-grams and their applications in text representation, spelling correction, and information retrieval. Feature Extraction: Bag of Words, Document-Term Matrix, TF-IDF; Feature selection: Chi-square, Mutual Information. Word embeddings: Word2Vec (CBOW, Skip-gram), GloVe, Embedding intuition.

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

2.1 Corpus Linguistics & Annotation Schemes

Definition: Text Corpus (plural: Corpora)

A Corpus is a large, structured, electronically stored collection of machine-readable natural language texts sampled to be representative of a language or specific domain. Used for statistical linguistic analysis, training machine learning models, and testing NLP hypotheses.

Types of Text Corpora

- **1. Monolingual Corpora:** Texts written in a single language (e.g., British National Corpus, Brown Corpus).
- **2. Multilingual & Parallel Corpora:** Parallel texts containing exact sentence-by-sentence translations across multiple languages (e.g., Europarl parallel corpus; essential for Machine Translation).
- **3. Annotated Corpora:** Texts enriched with explicit linguistic metadata tags (e.g., Penn Treebank with POS and syntax trees, OntoNotes with NER and coreference).

POS & NER Annotation Schemes

- **POS Tagging Schemes:** Penn Treebank 36-tag set (e.g., NN = Noun singular, NNS = Noun plural, VB = Verb base form, VBD = Verb past tense, JJ = Adjective, DT = Determiner).
- **Named Entity Recognition (NER) Schemes (IOB / BIO / BILOU):**
  - IOB / BIO Format: B-Tag (Beginning of entity), I-Tag (Inside multi-word entity), O-Tag (Outside any entity). Example: 'Sundar' [B-PER], 'Pichai' [I-PER], 'leads' [O], 'Google' [B-ORG].
  - BILOU Format: B (Beginning), I (Inside), L (Last token of entity), O (Outside), U (Unit/Single-token entity).
- **Annotation Tools:** Brat (web-based visual annotator), Doccano (open-source text annotation tool), Prodigy (active-learning annotation tool).

2.2 Text Preprocessing Pipeline & OOV Handling

1. The Core Preprocessing Pipeline

- **1. Tokenization:** Segmenting continuous character streams into discrete linguistic tokens (words, punctuation). Handled via whitespace split, regex rules, or subword algorithms.

- **2. Sentence Segmentation:** Splitting raw text into individual sentences based on punctuation heuristics and sentence boundary classifiers (e.g., distinguishing period in 'Dr. Smith' from sentence-ending period).
- **3. Text Normalization & Cleaning:** Lowercasing, stripping HTML/XML tags, removing non-ASCII symbols, expanding contractions (e.g., 'don't' -> 'do not').
- **4. True-Casing:** Restoring the proper linguistic capitalization of words regardless of position in sentence (e.g., differentiating initial capitalized 'Apple' company vs common fruit 'apple').
- **5. Stopword Removal:** Filtering out high-frequency grammatical words with minimal semantic information (e.g., 'the', 'is', 'in', 'and', 'at') to reduce feature space dimensionality.

2. Stemming vs. Lemmatization

Comparison Dimension	Stemming (e.g., Porter Stemmer)	Lemmatization (WordNet Lemmatizer)
Core Mechanism	Heuristic, rule-based chopping of common inflectional prefixes and suffixes without vocabulary lookup.	Morphological linguistic analysis using a dictionary and Part-of-Speech context.
Output Validity	Output stem may NOT be a valid dictionary word (e.g., 'studies' -> 'studi', 'universal' -> 'univers').	Guarantees a valid base canonical dictionary form (Lemma) (e.g., 'studies' -> 'study', 'better' [Adj] -> 'good').
Computational Speed	Extremely fast; simple string pattern replacement rules.	Slower; requires morphological lookup and POS tag resolution.
The Porter Stemmer Algorithm	5-phase sequential cascade of condition-action suffix substitution rules (e.g., 'SSES' -> 'SS', 'EED' -> 'EE', 'ATIONAL' -> 'ATE').	Uses WordNet synset morpho-syntactic base tables.

3. Handling Out-of-Vocabulary (OOV) Words

When an NLP model encounters a word during inference that was absent from its training vocabulary V, it fails. Solutions include:

- **1. The `<UNK>` Special Token:** Replace all low-frequency training words with a special Unknown token `<UNK>`. During testing, any unseen word maps to `<UNK>`. Limitation: All unknown words share an identical generic representation.
- **2. Subword Tokenization Techniques:**
  - Byte-Pair Encoding (BPE): Starts with character vocabulary; iteratively merges the most frequent adjacent character/token pairs into subword units (e.g., 'unbelievable' -> 'un' + 'believ' + 'able'). Used in GPT-2/3/4.

- WordPiece: Similar to BPE, but merges pairs based on maximizing the likelihood of a language model. Used in BERT.
- SentencePiece: Language-independent subword tokenizer treating raw input as a stream of bytes without whitespace pre-tokenization.

2.3 Word Representation: Sparse vs. Dense & WordNet

1. One-Hot Encoding & Co-Occurrence Matrices

- **One-Hot Encoding:** Represents word i as a binary vector of dimension |V| with 1 at index i and 0 everywhere else. Fatal flaws: (1) Extreme high dimensionality (|V| = 500,000+), (2) Severe sparsity, (3) Semantic Orthogonality (dot product <w_cat, w_dog> = 0; cannot capture semantic similarity).
- **Distributional Semantics (J.R. Firth, 1957):** 'You shall know a word by the company it keeps.' Words occurring in similar linguistic contexts share similar semantic meanings.
- **Word-Context Co-Occurrence Matrix:** Constructs a matrix C of size (|V| x |V|) where C[i, j] counts how many times word i co-occurs within a sliding context window of word j.

2. Sparse vs. Dense Vector Representations

Comparison Dimension	Sparse Representations (One-Hot, BoW, DTM)	Dense Distributed Embeddings (Word2Vec, GloVe)
Dimensionality	Massive: dimension equals vocabulary size |V| (50,000 to 1,000,000+).	Low: compact continuous vector space (typically 100 to 300 dimensions).
Vector Density	Extremely sparse (> 99.9% elements are zeros).	100% dense (all elements are real-valued continuous floats).
Semantic Relationships	Zero semantic encoding; all word vector pairs are orthogonal (distance = 0).	Captures rich syntactic & semantic analogies (Cosine similarity, vector arithmetic).
Generalization & Out-of-Domain	Poor generalization; vulnerable to the curse of dimensionality.	Superior generalization; maps similar unseen contexts into nearby vector clusters.

3. Lexical Semantic Resources: WordNet

WordNet (Fellbaum, Miller, 1998) is a large curated lexical database of English organized into Synsets (Sets of Cognitive Synonyms):

- **Core Semantic Relations in WordNet:**
  - • Hypernymy (IS-A / Super-concept): 'Canine' is a hypernym of 'Dog'.

  - • Hyponymy (Sub-concept): 'Dog' is a hyponym of 'Animal'.
  - • Meronymy (Part-Of): 'Wheel' is a meronym of 'Car'.
  - • Holonymy (Whole-Entity): 'Car' is a holonym of 'Wheel'.
  - • Antonymy (Opposite): 'Hot' vs 'Cold'.
- **Semantic Similarity Metrics:** Path Similarity (shortest path in hypernym taxonomy tree), Wu-Palmer Similarity (depth of least common subsumer LCS).

2.4 Feature Extraction (BoW, TF-IDF) & Selection

1. Bag of Words (BoW) & Document-Term Matrix (DTM)

Bag of Words (BoW) models a document purely as an unordered multiset of word frequencies, disregarding grammatical syntax and word order. A Document-Term Matrix (DTM) of size (N x |V|) stores word occurrence counts across N corpus documents.

2. Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF Mathematical Formulation

TF-IDF evaluates the relative importance of term t inside document d within corpus D of N documents:

*TF-IDF(t, d, D) = TF(t, d) * IDF(t, D)*

• *Term Frequency (TF): TF(t, d) = count(t, d) / [total words in document d]*

• *Inverse Document Frequency (IDF): IDF(t, D) = ln(N / DF(t)) (where DF(t) is count of documents containing term t)*

Intuition: High weight is given to words frequent in document d, but rare across the entire corpus (informative discriminative keywords!).

3. Feature Selection: Chi-Square (X^2) & Mutual Information (MI)

- **Chi-Square (X^2) Test:** Measures statistical independence between the presence of term t and document class category c: X^2(t, c) = [N * (AD - BC)^2] / [(A+B)(C+D)(A+C)(B+D)]. High X^2 indicates strong class dependence.
- **Mutual Information (MI):** Quantifies the reduction in uncertainty about class c upon observing term t: I(t; c) = sum sum P(t, c) * log_2 [P(t, c) / (P(t) * P(c))].

2.5 Word Embeddings: Word2Vec (CBOW, Skip-Gram) & GloVe

1. Word2Vec (Mikolov et al., Google 2013)

Word2Vec trains a 2-layer shallow neural network on raw unlabelled text to learn dense, continuous vector embeddings (dimension d = 100 to 300) that capture semantic and syntactic analogies.

- **Two Architectures of Word2Vec:**
  - **• Continuous Bag of Words (CBOW):** Predicts the Target center word w_t given the surrounding Context words [w_{t-c}, ..., w_{t+c}]. Fast training; superior for frequent words.
  - **• Skip-Gram Architecture:** Predicts the surrounding Context words given a single Input center word w_t. Objective: max sum_{t=1}^T sum_{-c <= j <= c, j != 0} log P(w_{t+j} | w_t). Superior for rare words and small datasets.
- **Optimization (Negative Sampling - SGNS):** Replaces the computationally prohibitive full-vocabulary softmax with binary logistic regression distinguishing true context pairs from k randomly drawn noise words: L = log sigma(v'_{w_O}^T v_{w_I}) + sum_{i=1}^k E_{w_i ~ P_n(w)} [log sigma(-v'_{w_i}^T v_{w_I})].
- **Geometric Vector Arithmetic:** Vector('King') - Vector('Man') + Vector('Woman') approx Vector('Queen').

2. GloVe (Global Vectors for Word Representation — Pennington et al., 2014)

GloVe combines the advantages of global matrix factorization (LSA) and local context window methods (Word2Vec). It optimizes a log-bilinear model directly over the non-zero cells of the Global Co-occurrence Matrix X:

- **GloVe Objective Function:** J = sum_{i, j=1}^{|V|} f(X_{ij}) * [w_i^T w_j_tilde + b_i + b_j_tilde - ln(X_{ij})]^2, where f(X_{ij}) = (X_{ij} / x_max)^alpha is a weighting function preventing frequent stopwords from dominating loss.

2.6 High-Yield University Exam Question Bank

Part A: Short Answer Questions (2 to 5 Marks)

Q1. Differentiate between Stemming and Lemmatization with examples. [3 Marks]

- **Answer:** Stemming uses crude heuristic suffix-stripping rules (e.g., Porter stemmer: 'caring' -> 'car'), often resulting in non-words. Lemmatization uses morphological dictionaries and POS tags to return true canonical base forms (e.g., 'caring' [Verb] -> 'care').

Q2. Define TF-IDF with formulas and explain its intuition. [4 Marks]

- **Answer:** TF-IDF(t, d) = TF(t, d) * IDF(t, D), where TF = count(t,d)/total_words, and IDF = ln(N/DF(t)). High weight is assigned to terms frequent in a specific document but rare across the corpus, isolating keywords.

Q3. Differentiate between CBOW and Skip-Gram in Word2Vec. [3 Marks]

- **Answer:** CBOW predicts a target word from its context words (fast, good for frequent words). Skip-Gram predicts surrounding context words given a target center word (better for rare words and semantic analogies).

Part B: Long Answer Questions (10 to 15 Marks Outline Guides)

- **Q4. Explain WordNet architecture, Synsets, and semantic relations (Hypernymy, Hyponymy, Meronymy, Holonymy) with examples. [10 Marks]**
- **Q5. Compare Sparse vs. Dense representations. Derive the Word2Vec Skip-Gram objective function with Negative Sampling. [12 Marks]**
- **Q6. Explain Subword Tokenization techniques (Byte-Pair Encoding, WordPiece, SentencePiece) and how they solve the Out-of-Vocabulary (OOV) problem. [10 Marks]**
- **Q7. Explain GloVe embedding derivation from the global co-occurrence matrix and compare GloVe vs. Word2Vec. [10 Marks]**

2.7 Unit Summary & Key Takeaways

- Corpora provide representative linguistic text; annotated via POS (Penn Treebank) and NER (IOB/BIO schemes).
- Preprocessing pipeline involves tokenization, normalization, cleaning, true-casing, stemming (Porter rules), and lemmatization (WordNet).
- Out-of-vocabulary (OOV) tokens are handled using subword algorithms (BPE, WordPiece, SentencePiece).
- Sparse vectors (One-Hot, BoW) suffer from orthogonality and high dimensionality; dense embeddings (Word2Vec, GloVe) capture continuous distributed semantics.
- TF-IDF isolates discriminative document keywords: TF(t,d) * ln(N/DF(t)).
- Word2Vec (CBOW & Skip-Gram SGNS) and GloVe log-bilinear co-occurrence factorization learn semantic vector geometry.