Lecture 02 - Contents

An overview of the main sections in this lecture.

Part 1
Clinical Text Processing

Part 2
Medical Ontologies

Part 3
Multimodal Processing

Hands-on
Preprocessing Hands-on

This outline is for guidance. Navigate the slides with the left/right arrow keys.

Lecture 2:

Medical Data Preprocessing and Curation

데이터 품질이 모델 성능의 80%를 결정합니다

Ho-min Park

homin.park@ghent.ac.kr

powersimmani@gmail.com

Lecture Overview

Part 1: Clinical Text Processing Pipeline

Part 2: Medical Ontologies and Coding Systems

Part 3: Multimodal Data Integration

Part 1/3:

Clinical Text Processing Pipeline

1. De-identification Techniques
2. PHI Detection and Removal
3. Clinical Text Normalization
4. Abbreviation Expansion
5. Negation Detection
6. Temporal Expression Extraction
7. Section Segmentation

De-identification Techniques

Safe Harbor Method

HIPAA standard method that removes 18 identifiers

- Removes specified items such as names, addresses, dates
- Relatively simple implementation
- Easy regulatory compliance

Expert Determination

Expert judges re-identification risk to be very low

- Utilizes statistical methods
- Allows use of more data
- Requires expert verification

Rule-based Pattern Matching

Automatic detection using regular expressions

- Date pattern: \d{2}/\d{2}/\d{4}
- Phone number: \d{3}-\d{3}-\d{4}
- Fast processing speed

ML-based Detection

Utilizing machine learning-based NER models

- BiLSTM-CRF, BERT models
- Context-based detection possible
- Achieves F1 score of 95%+

Accuracy Metrics Comparison

Precision

Precision

Ratio of actual PHI among detected

Recall

Recall

Ratio of detected among actual PHI

F1 Score

Harmonic Mean

Harmonic mean of Precision and Recall

FP Rate

False Positive Rate

Non-PHI incorrectly detected as PHI

PHI Detection and Removal

1 Name	**2** Address	**3** Date	**4** Phone Number	**5** Fax Number	**6** Email
7 SSN	**8** Medical Record Number	**9** Health Plan Number	**10** Account Number	**11** License Number	**12** Vehicle Number
13 Device ID	**14** Web URL	**15** IP Address	**16** Biometric ID	**17** Photo	**18** Other Unique Identifiers

Hybrid Approach: Rule-based + Machine Learning

Rule-based

Detect clear patterns with regular expressions

- Date: MM/DD/YYYY
- Phone: (123) 456-7890
- High precision, low recall

Machine Learning (ML-based)

Context-based detection with NER models

- BiLSTM-CRF, BERT
- spaCy, MedCAT
- High recall, contextual understanding

STEP 1
Original Text
Input Data

→

STEP 2
PHI Detection
NER + Regex

→

STEP 3
Masking/Removal
Anonymize

→

STEP 4
Validation
Validate

Clinical Text Normalization

Abbreviation Expansion

50,000+

Medical Abbreviation Dictionary Entries

85%

Context-Based Expansion Accuracy

Abbreviation Types

- General abbreviations: BP → blood pressure
- Drug abbreviations: ASA → aspirin
- Diagnostic abbreviations: MI → myocardial infarction
- Test abbreviations: CBC → complete blood count

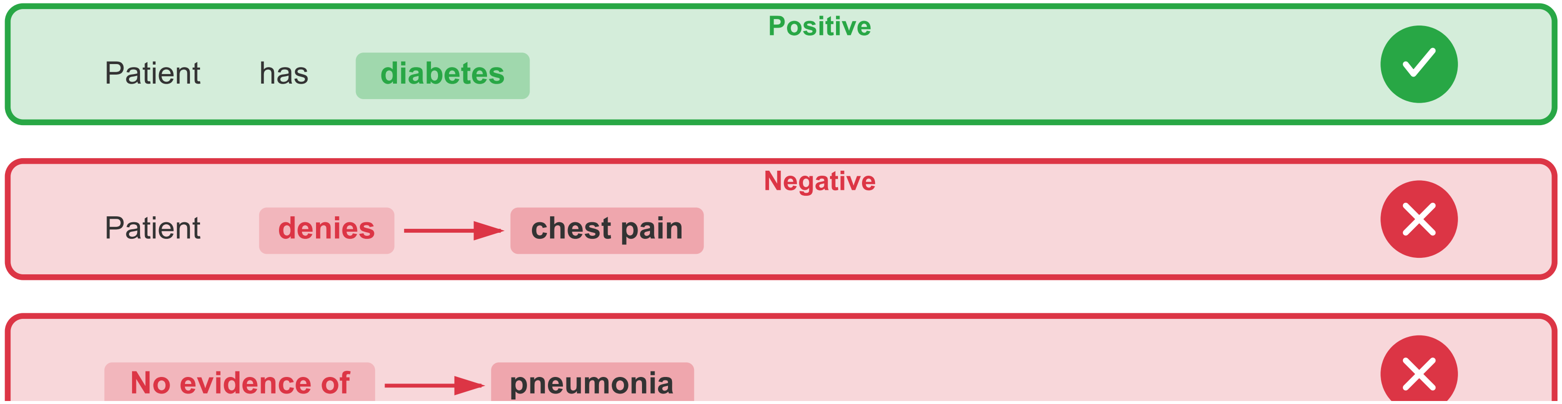
Ambiguity Resolution

- MS → multiple sclerosis vs. mitral stenosis
- RA → rheumatoid arthritis vs. right atrium
- Context analysis required
- UMLS utilization

Negation Detection

NegEx Algorithm

A rule-based algorithm that determines negation expressions and their scope of influence



Negation Triggers

no, not, denies, without, absent, negative, rule out, free of

Possibility Triggers

possible, probable, likely, suspected, questionable, consider

Temporal Expression Extraction

Date

- 2023-03-15
- March 15, 2023
- 03/15/2023

Duration

- 3 weeks
- for 2 months
- since 2020

Frequency

- twice daily
- every 6 hours
- once a week

HeidelTime

Rule-based temporal expression extraction

SUTime

Stanford temporal expression recognizer

Section Segmentation

Chief Complaint
Main reason for patient's visit

HPI
History of Present Illness - progression of current disease

Past Medical History
Previous illnesses and surgical history

Physical Exam
Vital signs and examination findings

Assessment
Diagnosis and clinical judgment

Plan
Treatment plan and follow-up strategy

Boundary Detection Methods
- Header keyword matching (HISTORY:, ASSESSMENT:)
- Machine learning-based segmentation
- F1 Score: 92-96%

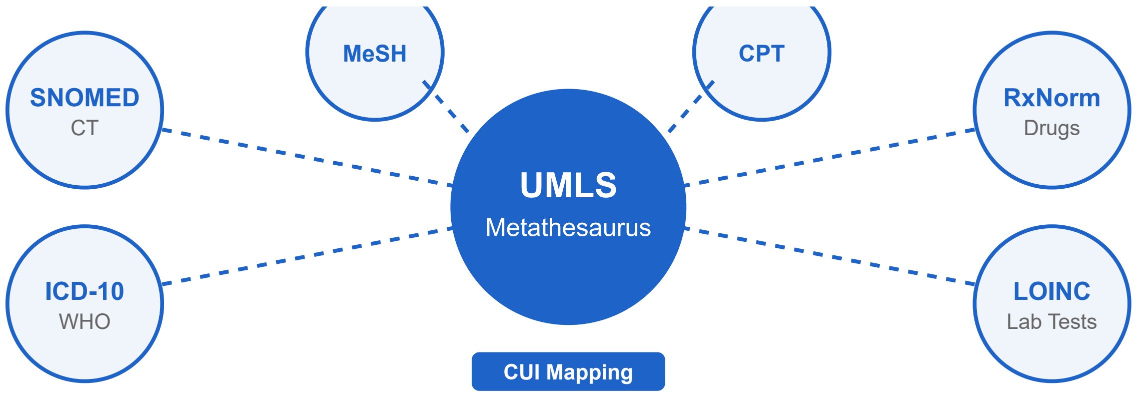
Part 2/3:

Medical Ontologies and Coding Systems

1. UMLS Metathesaurus
2. SNOMED CT Hierarchy
3. ICD-10/11 Coding
4. RxNorm Drug Normalization
5. LOINC Lab Values
6. Entity Linking Techniques

UMLS Metathesaurus

200+	3.8M	14M
Source Vocabularies	Concepts (CUI)	Unique Names



Metathesaurus

A concept database integrating various medical terminology systems

- CUI (Concept Unique Identifier)
- Synonyms and translations
- Cross-source mapping

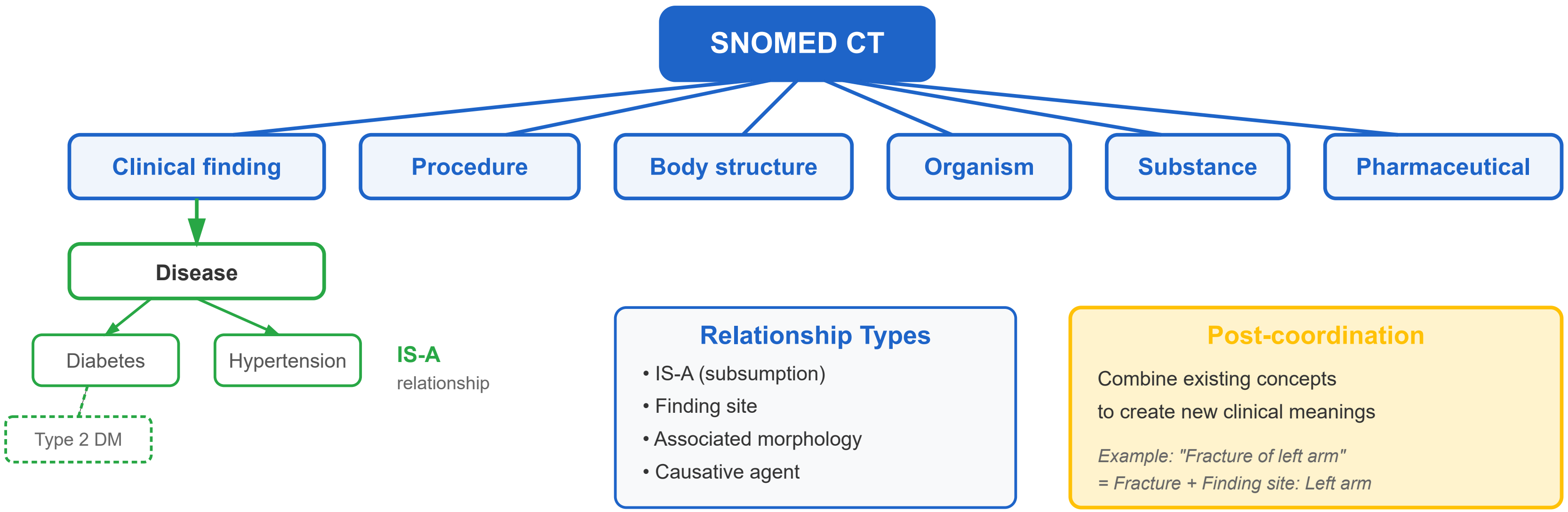
Semantic Network

Composed of 135 semantic types and 54 relationships

- is_a relationship
- associated_with
- treats, causes, etc.

SNOMED CT Hierarchy

350,000+ concepts | 19 top-level hierarchies | 1.5M+ relationships



ICD-10/11 Coding

ICD-10

- 70,000+ codes
- 21 chapters
- A00-Z99 range
- 7-digit detailed codes

ICD-11

- 55,000 codes (simplified)
- 26 chapters
- Online search optimized
- Improved scalability

Automatic Coding Algorithm

- NLP-based clinical note analysis
- 85%+ accuracy
- Rule-based + ML hybrid
- BERT-based coding model

RxNorm Drug Normalization

RxNorm Concept Model

- **Ingredient**: Active ingredient (aspirin)
- **Clinical Drug**: Ingredient + strength (aspirin 81 MG)
- **Branded Drug**: Brand name (Bayer Aspirin 81 MG)
- **RxCUI**: Unique identifier

NDC Mapping

Linked with National Drug Code

Includes manufacturer and package information

Interaction Check

Drug-drug interaction data

Verify contraindications

Normalization Process

Transforms drug data into standardized RxCUI codes to enable consistent data management and analysis.

RxNorm Hierarchy Structure

Ingredient

(Active component)

↓

Clinical Drug
(Ingredient + strength)

↓

Branded Drug
(Brand product name)

Example 1

Ingredient: aspirin
Clinical: aspirin 81 MG
Branded: Bayer Aspirin 81 MG

Example 2

Ingredient: metformin
Clinical: metformin 500 MG
Branded: Glucophage 500 MG

Example 3

Ingredient: lisinopril
Clinical: lisinopril 10 MG
Branded: Prinivil 10 MG

Key Applications of RxNorm

✓ Data Integration

Integrate drug data from different systems

✓ Clinical Decision Support

Provide drug information and alerts during prescribing

✓ Research Analysis

Analyze drug utilization patterns and outcomes

✓ Claims Management

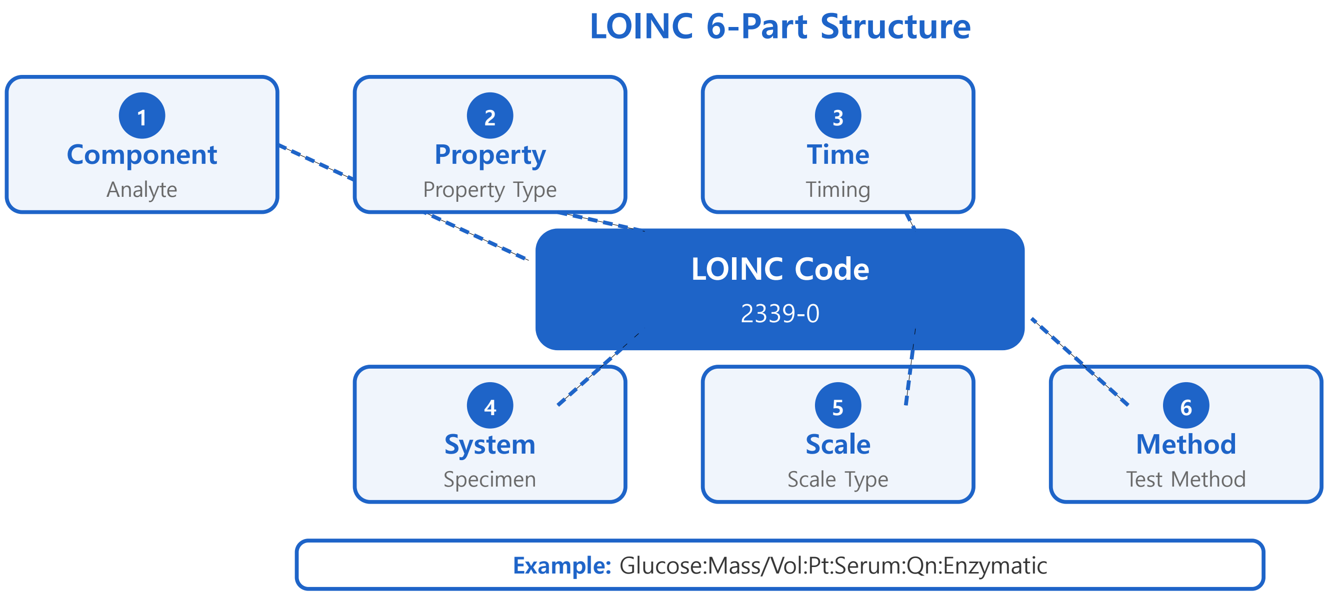
Use standardized codes for insurance claims

💡 Key Points

> RxNorm is a standardized drug nomenclature system provided by the U.S. National Library of Medicine (NLM), ensuring interoperability of drug information across different healthcare systems.

LOINC Lab Values

96,000+ Test Codes | **6-Part** Structure



1. Component (Analyte)

The substance or entity being measured or observed in the test.

Examples:

- Glucose
- Hemoglobin
- Creatinine
- Sodium

2. Property (Property Type)

The characteristic or type of the measured value.

Key Properties:

- **Mass/Vol (MCnc)** - Mass Concentration
- **Substance/Vol (SCnc)** - Substance Concentration
- **Arbitrary/Vol (ACnc)** - Arbitrary Concentration
- **Presence (Prid)** - Presence/Absence

3. Time (Timing)

The temporal characteristic of when the test was performed.

Time Types:

- **Pt (Point in time)** - Single point in time
- **24H** - 24-hour collection
- **8H** - 8-hour collection
- **Random** - Random time point

4. System (Specimen)

The type of biological sample being tested.

Specimen Types:

- **Serum** - Blood serum
- **Plasma** - Blood plasma
- **Blood** - Whole blood
- **Urine** - Urine

- **CSF** - Cerebrospinal fluid

5. Scale (Scale Type)

The data type of the measurement result.

Scale Types:

- **Qn (Quantitative)** - Numeric values
- **Ord (Ordinal)** - Ordered categories
- **Nom (Nominal)** - Named categories
- **Nar (Narrative)** - Text description

6. Method (Test Method)

The specific method or technique used to perform the test. (Optional)

Method Examples:

- **Enzymatic** - Enzymatic method
- **Immunoassay** - Immunoassay method
- **Chromatography** - Chromatography
- **Electrophoresis** - Electrophoresis

Real LOINC Code Examples

2339-0

Glucose:MCnc:Pt:Ser:Qn:Enzymatic

→ Quantitative measurement of glucose mass concentration in serum at a point in time using enzymatic method

718-7

Hemoglobin:MCnc:Pt:Bld:Qn

→ Quantitative measurement of hemoglobin mass concentration in whole blood at a point in time

2160-0

Creatinine:MCnc:Pt:Ser/Plas:Qn

→ Quantitative measurement of creatinine mass concentration in serum/plasma at a point in time

💡 Benefits of Using LOINC

- **Standardization:** Consistent interpretation of test results across healthcare institutions
- **Interoperability:** Easy data exchange between different systems
- **Accuracy:** Clear definition of test items prevents misunderstandings
- **Efficiency:** Enables automated data processing and analysis

Entity Linking Techniques

String Matching

- Exact matching
- Fuzzy matching
- Levenshtein distance
- Soundex, Metaphone

Semantic Similarity

- Word embeddings
- BERT embeddings
- Cosine similarity
- Semantic distance

Context-based Linking (Ensemble)

Combining rule-based + string matching + semantic similarity
Considering surrounding words and context
Can achieve 90%+ accuracy

📊 Entity Linking Process

Text Input → Entity Recognition → Candidate Generation → Ranking → KB Linking

🔍 Detailed Technique Descriptions

1. String Matching

Exact Matching: Perfect match search
Example: "Apple" → "Apple Inc."

Fuzzy Matching: Allows typos
Example: "Microsft" → "Microsoft"

Levenshtein Distance: Edit distance calculation
Measures insert/delete/replace operations

2. Semantic Similarity

Word Embeddings: Word vectorization
Utilizing Word2Vec, GloVe

BERT Embeddings: Context-based embeddings
Can distinguish homonyms

Cosine Similarity: Vector similarity
Range -1 to 1, closer to 1 means more similar

3. Rule-based

Naming Rules: Proper noun patterns
Capital letter start, special formats

Domain Knowledge: Field-specific rules
Medical, legal, technical terms

Context Rules: Surrounding word patterns
"CEO of", "located in", etc.

4. Ensemble Methods

Weighted Combination: Summing scores from each technique
α·string + β·semantic + γ·rule

Voting Approach: Majority decision
Combining predictions from multiple models

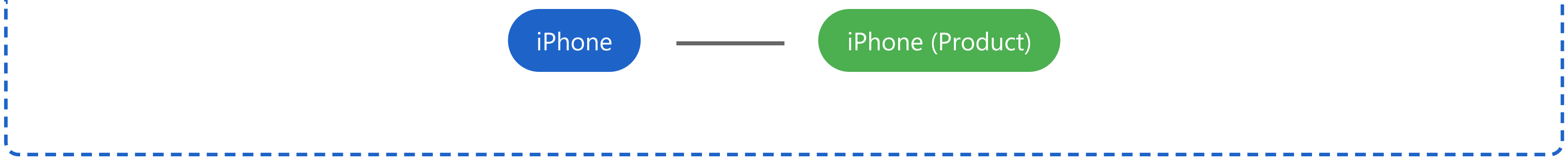
Sequential Application: Stepwise filtering
High confidence → low confidence order

💡 Real-world Application Example

Sentence: "Apple's CEO Tim Cook announced a new iPhone"

Apple — Apple Inc. (Company)

Tim Cook — Tim Cook (Person)



⚙️ Step-by-Step Processing

1. **Entity Extraction**
   Identify entities from text through NER → Person names, organization names, place names, etc.

2. **Candidate Generation**
   Select similar entities from knowledge base as candidates → Maximum 10-20 candidates

3. **Context Analysis**
   Analyze surrounding words and sentence structure to understand meaning

4. **Ranking**
   Calculate confidence score for each candidate by combining multiple techniques

5. **Final Linking**
   Select the candidate with the highest score and link to KB entity

📈 Technique Comparison

Technique	Advantages	Disadvantages	Use Cases
String Matching	Fast processing speed Simple implementation	Cannot distinguish homonyms No context consideration	Proper noun search Initial filtering

Technique	Advantages	Disadvantages	Use Cases
Semantic Similarity	Context understanding Excellent synonym handling	High computational cost Requires embedding training	Natural language understanding Semantic search
Rule-based	Domain specialization High interpretability	Difficult maintenance Limited scalability	Specialized domains Structured data
Ensemble	High accuracy Excellent robustness	Increased complexity Requires tuning	Production systems High accuracy requirements

💡 **Practical Tips:**

- Initial Prototype: Start with string matching
- Accuracy Improvement: Add semantic similarity
- Optimization: Apply domain-specific rules
- Production: Integrate with ensemble methods
- Continuous monitoring and feedback incorporation essential

🎯 Key Summary

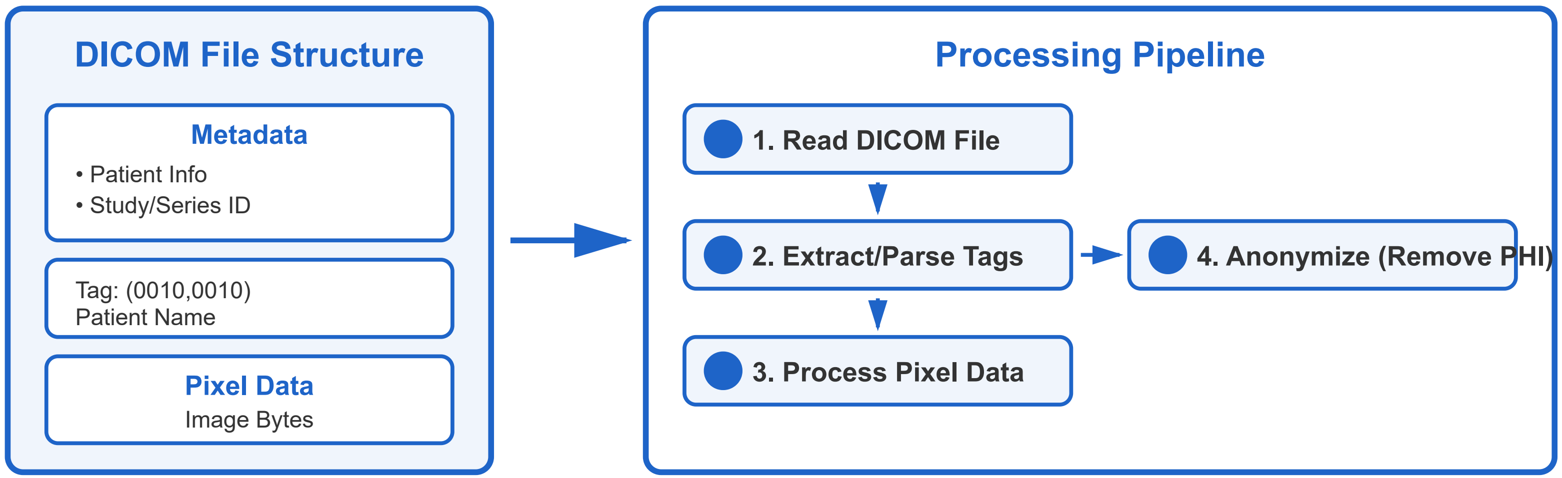
Entity Linking is the process of connecting entities in text with structured entities in a knowledge base. It uses various techniques from simple string matching to deep learning-based semantic analysis, and in real systems, multiple methods are ensembled to achieve accuracy rates over 90%.

Part 3/3:

Multimodal Data Integration

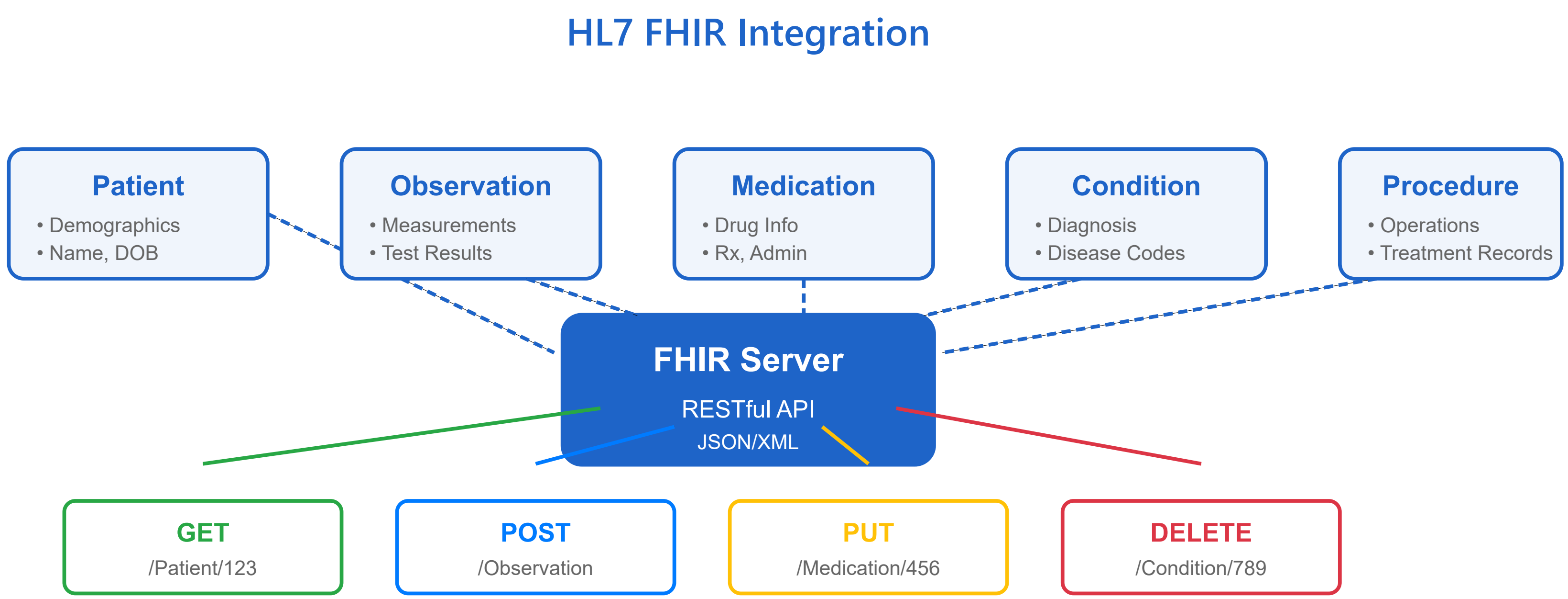
1. DICOM Image Handling
2. HL7 FHIR Integration
3. Waveform Signal Processing
4. Lab Value Normalization
5. Data Quality Assessment
6. Bias Detection & Mitigation

DICOM Image Handling



Python Library: pydicom

```
import pydicom
ds = pydicom.dcmread('image.dcm')
pixel_array = ds.pixel_array
patient_name = ds.PatientName
```



RESTful API Features

- **GET** /Patient/123 - Retrieve patient information
- **POST** /Observation - Create observation data
- Data exchange in **JSON format**
- Ensure interoperability with standard resource structure

Waveform Signal Processing

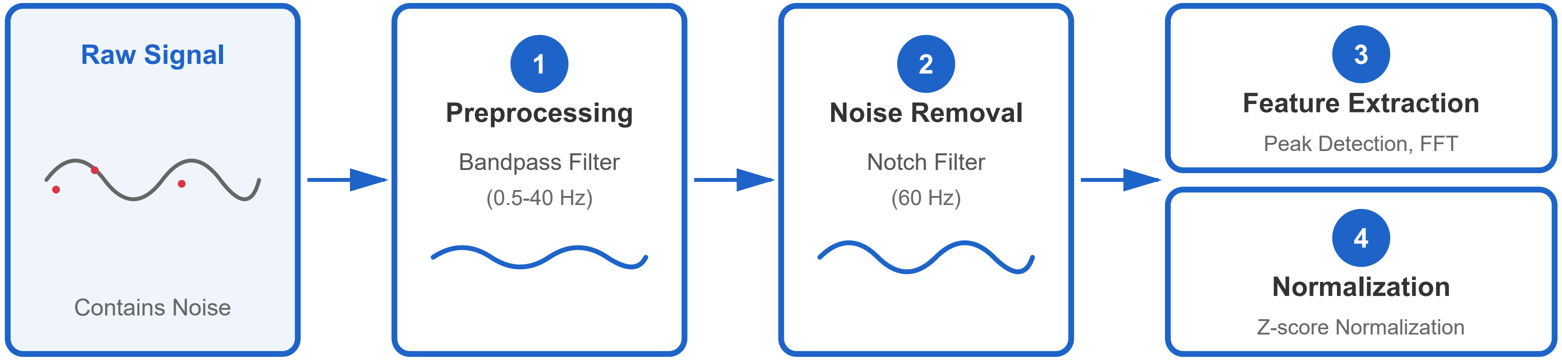
ECG (Electrocardiogram)

- Sampling: 250-500 Hz
- P, QRS, T wave detection
- Arrhythmia classification
- R-R interval calculation

EEG (Electroencephalogram)

- Sampling: 256-512 Hz
- Frequency band analysis
- Artifact removal
- Seizure detection

Signal Processing Pipeline



Lab Value Normalization

Unit Standardization

- **Glucose**: mg/dL ↔ mmol/L
- **Hemoglobin**: g/dL ↔ g/L
- **Creatinine**: mg/dL ↔ µmol/L
- SI units vs US conventional units

Reference Range Standardization

- Age-specific reference values
- Sex-specific reference values
- Pregnancy reference values
- Z-score calculation

Outlier Detection & Time Series Alignment

- IQR (Interquartile Range) method
- 3-sigma rule
- Time synchronization and missing value handling

1. Unit Conversion

Conversion Formulas for Key Lab Tests:

- **Glucose**: mg/dL × 0.0555 = mmol/L
- **Hemoglobin**: g/dL × 10 = g/L
- **Creatinine**: mg/dL × 88.4 = µmol/L
- **Cholesterol**: mg/dL × 0.0259 = mmol/L

When integrating multinational data, standardization to SI units is recommended.

2. Z-Score Normalization

$$Z = (X - \mu) / \sigma$$

X: measured value, μ: mean, σ: standard deviation

Application Examples:

- $Z > 2$ or $Z < -2$: Suspected outlier
- Apply age/sex-specific reference ranges
- Enable comparison across multiple lab tests

3. Outlier Detection Methods

IQR Method

Q1 = 25th percentile

Q3 = 75th percentile

IQR = Q3 - Q1

Outliers: < Q1-1.5×IQR or > Q3+1.5×IQR

3-Sigma Rule

$\mu \pm \sigma$: Contains 68.3%

$\mu \pm 2\sigma$: Contains 95.4%

$\mu \pm 3\sigma$: Contains 99.7%

Outliers: $|X - \mu| > 3\sigma$

4. Time Series Data Processing

Missing Value Handling Strategies:

Forward Fill	Interpolation	Mean/Median
Fill with previous value	Linear interpolation	Replace with mean/median

For clinical data, expert review is recommended rather than simple imputation.

Normalization Process Visualization

Raw Data → **Unit Conversion** → **Normalization** → **Quality Control**

- **Raw Data**: Various units, Non-uniform range
- **Unit Conversion**: Standard units (SI units)
- **Normalization**: Z-score, Min-Max
- **Quality Control**: Outlier removal, Missing value handling

💡 Practical Application Checklist

✓ Verify units by data source

✓ Preserve reference range metadata

✓ Check distribution before normalization

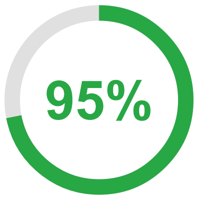
✓ Document conversion history

✓ Specify outlier handling criteria

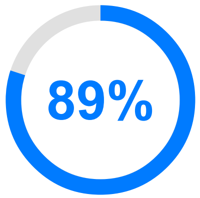
✓ Ensure reversibility of conversions

Data Quality Assessment

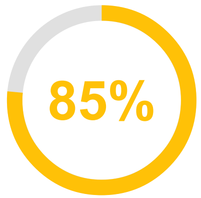
Completeness



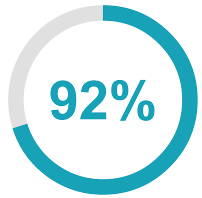
Accuracy



Consistency



Timeliness



Quality Score Calculation

- Field-level completeness measurement
- Data type validation
- Range checking
- Overall quality score (0-100)

Improvement Strategy

- Automated validation rules
- Anomaly flagging
- Data profiling
- Quality dashboard

Quality Trend Monitoring

Time series quality metrics tracking, periodic report generation, alert system implementation

Bias Detection & Mitigation

Bias Types

- **Demographic Bias**: Underrepresentation of specific race/gender
- **Selection Bias**: Non-random sampling
- **Measurement Bias**: Differences in measurement tools
- **Label Bias**: Annotator prejudice

Fairness Metrics

- **Demographic Parity**
- **Equalized Odds**
- **Disparate Impact**
- **Individual Fairness**

Mitigation Techniques

- Resampling (over/under sampling)
- Weight adjustment
- Adding fairness constraints
- Post-processing for bias mitigation

Bias Detection Process

Step 1: Data Collection and Analysis

- Identify protected attributes (gender, race, age, etc.)
- Check data distribution by group
- Measure degree of imbalance

Step 2: Model Training and Evaluation

- Train baseline model
- Measure performance metrics by group
- Fairness Metrics 계산

Step 3: Apply Bias Mitigation

- 적절한 Mitigation Techniques 선택
- Mitigation Techniques 적용 및 재Evaluation
- Analyze performance-fairness trade-offs

Fairness Metrics 상세

Demographic Parity

$P(\hat{Y}=1|A=0) = P(\hat{Y}=1|A=1)$

Equal positive prediction rates across all groups

Disparate Impact

Ratio = $P(\hat{Y}=1|A=0) / P(\hat{Y}=1|A=1)$

Generally considered fair if ≥ 0.8

Equalized Odds

Equal TPR and FPR across all groups

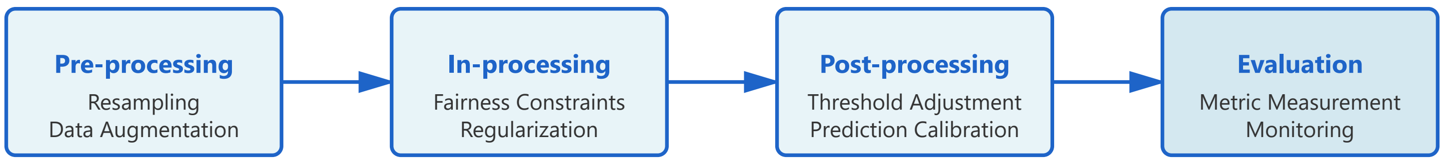
$P(\hat{Y}=1|Y=y, A=0) = P(\hat{Y}=1|Y=y, A=1)$

Individual Fairness

Similar individuals should receive similar predictions

$d(x_1, x_2) \approx 0 \rightarrow d(f(x_1), f(x_2)) \approx 0$

Bias Mitigation Pipeline



Real-World Application Examples

Hiring System

Problem: 특정 성별이 과소Evaluation됨

Solution: Remove gender + Equalized Odds

Result: Group acceptance rate gap: 15% → 3%

Loan Approval

Problem: Racial approval rate disparity

Solution: Resampling + Threshold Adjustment

Result: Disparate Impact 0.65 → 0.85

Implementation Considerations

Trade-off

Balance between performance and fairness

Transparency

Explainability of decision process

Monitoring

Continuous bias monitoring

Legal Compliance

Regulatory and ethical standards

Tools and Libraries

Python Libraries:

- **Fairlearn**: Microsoft의 공정성 Evaluation 및 완화 도구
- **AIF360**: IBM's AI Fairness 360 toolkit
- **What-If Tool**: Google's visual analysis tool
- **Themis-ML**: Fairness-aware machine learning

Evaluation 프레임워크:

- Fairness Indicators (TensorFlow)
- FairTest
- Aequitas

Missing Data Strategies

Missing Patterns

- **MCAR**: Missing Completely At Random
- **MAR**: Missing At Random
- **MNAR**: Missing Not At Random

Visualize patterns with missing data heatmap

Imputation Methods

- **Mean/Median** Imputation
- **KNN** Imputation
- **MICE**: Multiple Imputation
- **Deep Learning** Based

Impact Analysis

Compare model performance before and after imputation, sensitivity analysis, impact assessment by missing rate

Missing Patterns in Detail

MCAR (Missing Completely At Random)

Missingness occurs completely randomly, independent of other variables. Data loss exists but no bias.

Example: Random non-responses in a survey

MAR (Missing At Random)

Missingness depends on observed variables but not on the missing value itself. Most common pattern.

Example: Older people are more likely to omit income information

MNAR (Missing Not At Random)

Missingness is related to the missing value itself. Most difficult pattern to handle.

Example: People with very high or low income omit income information

Imputation Methods Comparison

Method	Advantages	Disadvantages	Suitable Situations
Mean/Median	Fast and simple	Reduces variance, distorts relationships	MCAR, low missing rate
KNN	Utilizes similar cases	High computational cost	MAR, moderate missing rate
MICE	Reflects uncertainty	Complex and slow	MAR, high missing rate
Deep Learning	Learns complex patterns	Requires large data	Large-scale datasets

Missing Data Handling Process

Step 1
Explore Missing Patterns

→

Step 2
Identify Missing Mechanism

→

Step 3
Select Appropriate Method

→

Step 4
Perform Imputation and Validate

Performance Comparison Example

Original Accuracy
94.2%

Mean Imputation
89.5%

KNN Imputation
92.8%

MICE Imputation
93.6%

Key Considerations

⚠️ **Cautions**

- Consider variable removal over imputation if missing rate exceeds 40%
- Check impact of imputation method on target variable
- Compare multiple methods for optimal selection

✓ **Recommendations**

- Derive insights through missing pattern visualization
- Evaluate imputation methods with cross-validation
- Leverage domain knowledge for imputation

Python Implementation Example

```
# Check missing patterns

import missingno as msno
msno.matrix(df)

# Mean imputation
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
df_filled = imputer.fit_transform(df)

# KNN imputation
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=5)
df_filled = imputer.fit_transform(df)

# MICE imputation
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
imputer = IterativeImputer()
df_filled = imputer.fit_transform(df)
```

Recommended Strategy by Missing Rate

≤ 5%

5-20%

20-40%

> 40%


Simple Imputation
(Mean/Median)
KNN or
Regression Imputation
MICE or
Advanced Methods
Consider Variable
Removal

Data Augmentation Techniques

텍스트 증강

- **역번역**: EN→KO→EN
- **동의어 치환**: WordNet
- **패러프레이징**: T5, GPT
- **임의 삽입/삭제**

합성 데이터

- **GPT-4** 기반 생성
- **템플릿** 기반 생성
- **SMOTE**: 소수 클래스
- **GAN**: 이미지 생성

증강 효과

데이터 다양성 증가, 과적합 방지, 소수 클래스 성능 향상, F1 score +5-15%

Data Augmentation Techniques

Text Augmentation

- **Back Translation**: EN→KO→EN
- **Synonym Replacement**: WordNet
- **Paraphrasing**: T5, GPT
- **Random Insertion/Deletion**

Synthetic Data

- **GPT-4** based generation
- **Template** based generation
- **SMOTE**: Minority classes
- **GAN**: Image generation

Augmentation Effects

Increased data diversity, overfitting prevention, improved minority class performance, F1 score +5-15%

Text Augmentation Techniques Details

1. Back Translation

Translates the original text into another language and then back to the original language to maintain meaning while diversifying expressions.

Example: "The model performs excellently" → "모델 성능이 우수합니다" → "The model works well"

2. Synonym Replacement

Replaces specific words in a sentence with synonyms using dictionaries like WordNet.

Example: "fast execution" → "quick execution"

3. Paraphrasing

Uses language models like T5 and GPT to express sentences in different ways while maintaining their meaning.

4. Random Insertion/Deletion

Randomly adds or removes words from sentences to generate variations.

Synthetic Data Generation Techniques

1. GPT-4 Based Generation

Generates high-quality synthetic data for specific domains through prompt engineering.

2. Template-Based Generation

Creates structured data by inserting various entities into predefined templates.

3. SMOTE (Synthetic Minority Over-sampling Technique)

Generates new synthetic samples through interpolation between minority class samples to address class imbalance.

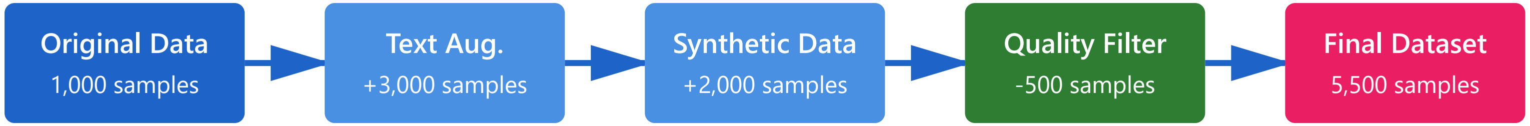
4. GAN (Generative Adversarial Network)

Generates realistic images or text through competitive learning between generator and discriminator.

Performance Comparison

Baseline Model	Text Augmentation	Synthetic Data	Mixed Techniques
75%	**82%**	**85%**	**90%**

Augmentation Process Flow



Implementation Considerations

✓ Advantages

- Improve model performance with limited data
- Resolve class imbalance issues
- Enhance model generalization capability
- Provide opportunities to learn new patterns

✗ Precautions

- Excessive augmentation may increase noise
- Need to select techniques considering domain characteristics
- Quality validation of augmented data is essential
- Consider computational cost and time consumption

Hands-on: Preprocessing with MIMIC-III

Python Code Example

```
import pandas as pd
from presidio_analyzer import AnalyzerEngine
from presidio_anonymizer import AnonymizerEngine

# PHI Removal
analyzer = AnalyzerEngine()
anonymizer = AnonymizerEngine()

text = "Patient John Doe, MRN 123456"
results = analyzer.analyze(text, language='en')
anonymized = anonymizer.anonymize(text, results)

# Abbreviation Expansion
abbrev_dict = {'BP': 'blood pressure', 'HR': 'heart rate'}
text = text.replace('BP', abbrev_dict['BP'])

# LOINC Mapping
loinc_code = '2339-0' # Glucose [Mass/volume] in Blood
```

- ✓ PHI Detection and Removal
- ✓ Text Normalization
- ✓ Abbreviation Expansion
- ✓ Negation Detection

✓ Concept Mapping

✓ Quality Validation

Best Practices Checklist

Preprocessing Checklist

- ✓ Confirm complete PHI removal
- ✓ Verify abbreviation consistency
- ✓ Standardize dates/units
- ✓ Handle negative expressions
- ✓ Ontology mapping
- ✓ Handle missing values
- ✓ Detect outliers
- ✓ Evaluate bias
- ✓ Calculate quality metrics
- ✓ Complete documentation

Quality Assurance Process

1. Sample Validation: Manual review of 100 random cases
2. Automated Testing: Unit tests and integration tests
3. Performance Benchmark: Processing speed and accuracy
4. Documentation: Record processing steps and decision-making

Thank you

다음 강의 예고: Lecture 3 - Advanced LLM Training

Ho-min Park

homin.park@ghent.ac.kr
powersimmani@gmail.com