CEAL 2025 Annual Meeting
Committee on Technical Processing (CTP) Session

The Current Methods of Romanization in Japanese Cataloging and Internationalization

Takashi Harada
(Doshisha University/
Yashima Gakuen University,
Gifu Women's University)

Outline

1. Introduction: Why romanization is essential
2. Overview of existing romanization methods
3. Evaluation and comparison of methods
4. Realistic solutions for implementation
5. Comparative study: MeCab & NDL API
6. AI and future prospects
7. Conclusion

Why Romanization is Important?

- **Compliance with International Standards**

  Romanization is required by international cataloging frameworks such as OCLC, PCC, and BIBFRAME to ensure interoperability among library systems worldwide.

- **Metadata Consistency**

  Standardized Romanization improves uniformity in bibliographic records, preventing discrepancies from multiple transliteration rules

- **Multilingual Integration**

  Enables seamless integration of Japanese bibliographic data with global library networks, making Japanese resources more accessible in international catalogs.

- **Improved Searchability**

  Although not the primary focus from a library perspective, romanization still enhances discoverability, especially for non-Japanese-speaking researchers and institutions.

International Standards & PCC Guidelines

- OCLC & PCC promote Model A (original + romanized text).
- The ALA-LC Romanization Tables define standard rules for transliteration.
- Japanese libraries use TRC MARC and NDL authority files to maintain consistency.
- These standards facilitate a unified bibliographic environment worldwide.

Evolution of Romanization Methods

- Pre-digital era (1940s–1980s):
  - Manual transliteration for card catalogs with multiple inconsistent styles (Hepburn, Kunrei).
- Early Digital Era (1980s–1990s):
  - Introduction of rule-based scripts for simple conversions; OPAC systems reduced the need for full romanization.
- Modern Methods (2000s–Present):
  - Utilization of morphological analysis tools (MeCab, ChaSen), commercial systems (Happiness), NDL Search API, and AI models.

Modern Methods (2000s–Present):

- Commercial systems(Happiness, etc.)
- Morphological analysis tools

  (MeCab, ChaSen)
- WebAPI (NDL Search API, etc.)
- AI/Generative AI (BERT, etc.)

Realistic Solutions for Library Operations

- The key question is: which method best meets the needs of libraries?
- We evaluate MeCab, Happiness, and the NDL Search API.
- Our findings suggest that a hybrid approach is necessary to balance accuracy, efficiency, and cost.

Commercial Systems

- Happiness is a commercial system for romanization used in many Japanese libraries.
- It achieves high accuracy with an integrated dictionary and offers reliable support and maintenance.
- However, it is expensive and involves high licensing fees, with the risk of vendor lock-in.

Open-Source Systems (MeCab, ChaSen)

- MeCab is a widely used open-source morphological analysis tool.
- It is free and highly customizable to follow specific romanization rules (e.g., ALA-LC).
- ChaSen(older alternative), MeCab require technical expertise for setup and continuous dictionary updates.

MeCab – Advantages & Limitations

✅ Strengths

- Fast and efficient morphological analysis
- Completely free and customizable
- Can be adapted to different romanization standards

❌ Weaknesses

- Requires technical setup and ongoing maintenance
- Dictionary updates needed for accurate results
- Does not include authority-controlled romanizations

WebAPI : NDL Search API

- The NDL Search API provides official romanized names from the National Diet Library.
- Ensures high reliability and consistency in bibliographic records.
- Limited to existing records, meaning newly published books may not be covered.
- Designed for use in standardized cataloging rather than real-time applications.

NDL API – Advantages & Limitations

✅ Strengths

- Provides authoritative romanized names for standardization.
- Ensures consistency across library records.
- Eliminates ambiguity in proper names and official terms.

❌ Weaknesses

- Limited to registered NDL records, excluding new publications.
- Cannot handle dynamic romanization or unknown words.
- Not designed for real-time, on-the-fly conversion.

AI Models : BERT, ChatGPT

- Machine learning models, like BERT and ChatGPT, can be used to generate romanized text.
- Can Process words in context, helping with ambiguous readings.
- Can handle unknown words, unlike rule-based systems.
- Rely on probability-based predictions, which may vary.
- It also suffers from hallucination problems.

Key Difference:
AI does not follow predefined rules like MeCab or NDL API; instead, it learns patterns from large DBs.

AI-Based – Advantages & Limitations

✅ Strengths

- Capable of romanizing words not found in dictionaries.
- Understand context and adapt more effictivery than rule-based systems.
- Improve with more training data over time.

❌ Weaknesses

- Inconsistent results – different outputs for the same input.
- Risk of hallucination – AI may generate incorrect but plausible romanizations.
- Requires high computational power and ongoing tuning.
  Cost : very expensive

The Difficulty of Romanizing Japanese with AI

A case of Japanese: Can AI romanize 強 correctly?

強力 strong	強: kyō	伊藤強	強: tsuyoshi
強力犯 violent criminal	強: gō	岩田強	強: tsutomu
強い strong	強: tsuyo	海東強	強: takeshi
強い stubborn	強: kowa	赤土正強	強: take
強いる to force	強: shi	赤石強司	強: kyō
強か severely	強: shitata	杉浦強司	強: tsuyo
強ち (not) necessarily	強: anaga	松平貴強	強: masu
強情 stubbornness	強情: gōjō	志村弘強	強: yuki
強情る to nag	強情: neda	強谷幸雄	強: sune
強情る to extort	強情: yusu	強力敏郎	強: gō

Hyoungbae Lee & Dae Chul Son. AI and Romanization: Possibilities and Limitations. CEAL2024

Comparing Methods

Method	Pros	Cons
Happiness (Commercial)	High accuracy, robust support	Expensive, vendor lock-in
MeCab (Open-Source)	Free, customizable	Requires technical setup
NDL Search API	Reliable, official readings	Limited coverage
AI Models	Good for unknown words	Risk of errors, inconsistency, Cost !

No single method is perfect; a balanced approach is needed.

AI-based Book Data Reading

- **ChatGPT Plus (o4 model):**
  - Correctly read 193 out of 200 book entries
  - Example: “死者と生者の市” read as “shisha to Seija no ichi”
- **ChatGPT Pro (o3 mini):**
  - Correctly read 191 out of 200 book entries
  - Same title resulted in a message indicating ambiguity: “No book found, and the reading is ambiguous”
- **Key Points**
  - o4 model is not higher performance than o3 mini
  - o3 mini currently incurs higher costs and longer processing times

Practical Approaches and Future Prospects

- **Current Challenges**
  - High costs and long processing times when using the o3 mini model
- **Short-term Solution**
  - Develop a system combining MeCab for Japanese morphological analysis with the NDL API
  - Provides a stable and cost-effective method for handling library data
- **Future Prospects**
  - Gradually integrate generative AI “Deep Research” capabilities into the existing system

MeCab + NDL API

- The combination of MeCab and the NDL Search API is the most practical for Short-Term Solution.
- MeCab handles routine romanization effectively, while the NDL API provides authoritative romanized names.
- This approach offers a cost-effective and accurate solution.
- A potential issue with this approach is the inconsistency between MeCab and NDL Search API results.

Experimental Evaluation

2,078,653 Data Points

	Count	Findings
A-1. Perfect Match	432,053	API and MeCab were fully identical (20% of cases).
A-2. Minor Fixes Needed	947,907	Spaces, punctuation, or minor variations.
A-3.Rule-based Corrections	222,930	Required specific processing rules (e.g., 'ha' vs. 'wa').
B. Major Differences	118,170	MeCab and API suggested different readings.
C. Complex Cases	387,512	Unique cases needing human intervention.

Difference of MeCab & API

📌 Findings

- A-1,A-2, A-3 **cover 3/4 of the data**, meaning a hybrid approach **can be highly automated**.

📌 Challenge

- **B and C (25% of data) still require intelligent rule processing or manual review**.

- These factors make it challenging to fully automate the processing of B and C categories, requiring manual verification and refined dictionary rules.

Challenges in Handling B Category

- “B: Major Differences” category includes cases where the word itself has a different reading.
- Examples include “nihon” vs. “nippon” and proper nouns, which may be partially resolved through dictionary refinement.
- For example, there are cases such as “男(おとこ)道” where only the National Diet Library’s data encloses the reading in parentheses.

Challenges in Handling C Category

- “C: Miscellaneous Cases” category contains various discrepancies that cannot be easily classified.
- While a complete review is impractical, a manual inspection of 1,000 cases revealed that 16.4% involved different kanji readings for single-character words (polyphonic characters).
- Other issues include:
  - A unique NDL rule where "砥部町" is read as "とべまち."
  - Cases where NDL omits subtitles but provides readings
  - Foreign language transcription variations
  - Dictionary boundary mismatches

How Should We Handle Differences?

- **Why API is not always correct:**
  - ❌ Sometimes **API has errors or outdated entries.**
  - ❌ Multiple romanization styles exist
    (e.g., **Sato, Satou, Satō**).
  - ✅ API is a **useful reference**,
    but final decisions require a **customized rule set.**

- **When should MeCab be overridden?**
  - 📌 If MeCab produces **clear misinterpretations**
    (e.g., 五月雨 → 'satukiame').

Choosing a Strategy:
API First, MeCab First, or Hybrid Comparison

API First Approach:

- Use API romanization for existing records (high reliability for known names).
- Use MeCab for unknown terms (context-based processing).
- ✅ Pro: High accuracy for authority-controlled data.
- ❌ Con: API calls can be costly and not always available.

MeCab First Approach:

- Use MeCab for all cases (completely independent of external services).
- Manually verify discrepancies for proper names.
- ✅ Pro: No API dependency; fully controllable.
- ❌ Con: Errors in proper names and ambiguous readings.

Hybrid (Comparison) Approach:

- Run both MeCab and API simultaneously.
- Compare results and apply priority rules (e.g., API for names, MeCab for general terms).
- ✅ Pro: Best balance of accuracy and coverage.
- ❌ Con: Complex implementation and requires manual fine-tuning.

Extraction of Proper Names

- NDL Search API
  - Recognizes proper names only if they are present in the authority files.
- MeCab:
  - Predicts whether a word is likely a proper noun based on character patterns and surrounding context, even if it’s missing from the dictionary.
  - However, its estimation accuracy is not perfect.

Machine Learning-Based Named Entity Recognition (NER)

- Research on NER in Japanese began in the late 1990s (e.g., IREX workshops).
- Early approaches used CRF-based sequence labeling with high accuracy in the 2000s.
- Recent deep learning models (e.g., BERT) have achieved F1 scores over 90%.
- ML-based NER can detect unknown proper names from context but requires sufficient training data.
- Recent efforts use large language models (LLMs) with prompt-based techniques for flexible proper name extraction and classification.

Combine AI with MeCab & Web API

- AI-based romanization offers additional support in handling unknown or ambiguous terms.
- Advanced AI models can analyze context and provide romanized outputs where traditional methods struggle.
- Aligning AI-based romanization with bibliographic standards is crucial—as authoritative databases expand and AI models refine, consistency will improve and the burden of manual corrections will be reduced.

AI-Driven Approaches & Achievements

- By adopting AI methods (e.g., RAG, fine-tuning) and leveraging the o3-mini model available in ChatGPT Pro, we have achieved remarkably high accuracy—not only for romanization but across many bibliographic challenges.
- Prompt engineering with ChatGPT Pro appears to offer viable solutions for issues in data organization such as classification numbers and subject heading assignments.

Challenges & Future Considerations

- Experiments reveal that even nearly correct generated data require costly human verification, potentially exceeding the cost of initial human data assignment.
- What are the defined goals for generative AI usage? How much verification cost is acceptable, and can librarians foresee a future where some level of error is tolerated, possibly eliminating human assignment entirely?

Conclusion

- This research indicates that the best current solution is a hybrid model combining MeCab and the NDL Search API.
- This approach offers a balanced solution that optimizes both automation and accuracy while remaining cost-effective.
- As AI continues to evolve, its role expands—but so does the cost of human oversight for high accuracy.
- Can librarians accept an approach that delegates all verification and correction to AI without human checks? **Even if it contains errors**

CEAL 2025 Annual Meeting
Committee on Technical Processing (CTP) Session

The Current Methods of Romanization in Japanese Cataloging and Internationalization

Takashi Harada
(Doshisha University/
Yashima Gakuen University,
Gifu Women's University)

Why Combine AI with MeCab & Web API for Romanization? (1)

- AI Challenges in Romanization:
  • AI models sometimes generate incorrect romanizations ("hallucination").
  • They can produce inconsistent outputs for the same input over time.
  • Training and maintaining advanced AI systems incur high costs.
- Ensuring Reliability and Reproducibility:
  • Web API (NDL Search API) relies on official authority data, ensuring high reproducibility for known terms.
  • MeCab, using rule-based and statistical models, processes routine romanization consistently.

Why Combine AI with MeCab & Web API for Romanization? (2)

- Maintaining Standardization and Rule Adherence:
  • Bibliographic data must follow international standards (e.g., ALA-LC, Hepburn).
- Error Mitigation and Fallback Mechanisms:
  • AI's flexibility in handling ambiguous cases is valuable, but its errors can be corrected through cross-checking with MeCab and API outputs.
- System Transparency and Interpretability:
  • Rule-based methods offer clear reasons for their outputs, aiding in troubleshooting.
- Operational Flexibility and Risk Diversification:
  • Combining different approaches minimizes dependency on a single method.