Graph-Structured Trajectory Extraction from Travelogues

Aitaro Yamamoto[♣,*] **Hiroyuki Otomo**[♡]
Hiroki Ouchi[♣,♦,†] **Shohei Higashiyama**[♠,♣]
Hiroki Teranishi[♦,♣] **Hiroyuki Shindo**[♢] **Taro Watanabe**[♣]
[♣] NAIST [♠] NICT [♦] RIKEN [♡] CyberAgent, Inc. [♢] MatBrain, Inc.
yamamoto.aitaro.xv6@is.naist.jp, otomo_hiroyuki@cyberagent.co.jp,
hiroki.ouchi@is.naist.jp, shohei.higashiyama@nict.go.jp,
hiroki.teranishi@riken.jp, hshindo@matbrain.jp, taro@is.naist.jp

Abstract

Human traveling trajectories play a central role in characterizing each travelogue, and automatic trajectory extraction from travelogues is highly desired for tourism services, such as travel planning and recommendation. This work addresses the extraction of human traveling trajectories from travelogues. Previous work treated each trajectory as a *sequence* of visited locations, although locations with different granularity levels, e.g., "Kyoto City" and "Kyoto Station," should not be lined up in a sequence. In this work, we propose to represent the trajectory as a *graph* that can capture the hierarchy as well as the visiting order, and construct a benchmark dataset for the trajectory extraction. The experiments using this dataset show that even naive baseline systems can accurately predict visited locations and the visiting order between them, while it is more challenging to predict the hierarchical relations.

1 Introduction

The advancement of Web technologies facilitates people to share their travel experiences on the Web in the form of textual travelogues (Hao et al., 2010). Travelogues are vital sources for analyzing human traveling behavior in tourism informatics, geographic information science, and digital humanities, because of their rich geographical and thematic content, which gives people, e.g., a simulated experience of trip (Haris and Gan, 2021). In particular, human traveling trajectories play a central role in characterizing each travelogue, and thus, automatic trajectory extraction from travelogues is highly desired for tourism services, such as travel planning and recommendation (Pang et al., 2011).

Some studies have addressed automatic trajectory extraction from text (Ishino et al., 2012; Wagner et al., 2023; Kori et al., 2006). However, these studies suffer from two issues: (i) inadequate trajectory representation and (ii) the scarcity of benchmark datasets. First, the previous studies treated each trajectory as a *sequence* of visited locations (Ishino et al., 2012; Wagner et al., 2023; Kori et al., 2006), but a sequence is inadequate as a representation of trajectories. This is because a pair of locations where one geographically includes the other cannot be lined up in a single sequence, for example, "Kyoto City" and "Kyoto Station." This necessitates more appropriate trajectory representations other than sequences, as we discuss in detail in §4.1. Second, the previous studies constructed and used their in-house datasets for evaluating their systems, and no public text datasets

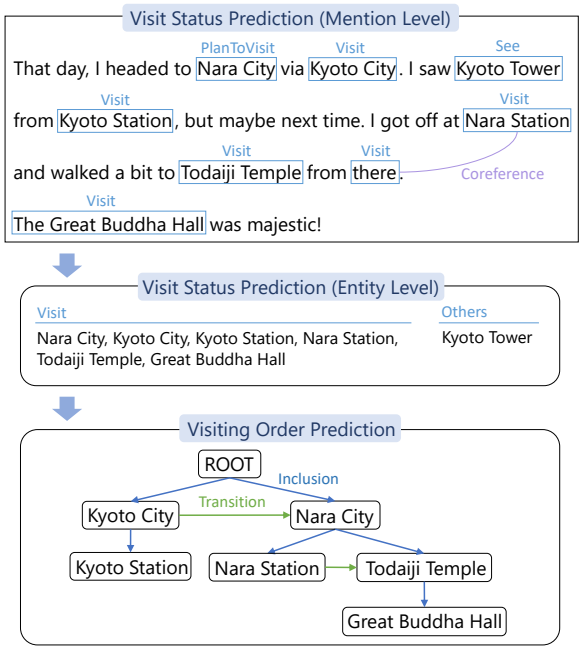

Figure 1: Illustration of our proposed tasks: visit status prediction (VSP) and visiting order prediction (VOP). VSP assigns visit status labels to mentions for mention level (top) and to entities for entity level (middle). VOP outputs a visiting order graph by assigning inclusion and transition relations to entity pairs (bottom).

[*]This work was done while he belonged to NAIST, and he is currently working at a company.
[†]Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14116–14132

annotated with trajectory information have been released. However, shared benchmark datasets are necessary for facilitating fair comparisons with other studies and accelerating the accumulation of research findings (Ohsuga and Oyama, 2021).

For the first issue, we propose a *visiting order graph* illustrated at the bottom of Figure 1. This graph has nodes of locations or geo-entities and edges of relations between geo-entities. It can represent not only temporal *transition relations* but also geographical *inclusion relations* between visited locations. For enabling automatic construction of the graph for each travelogue, we introduce trajectory extraction subtasks: *Visit Status Prediction* (VSP) and *Visiting Order Prediction* (VOP), as shown in Figure 1. VSP requires to assign *visit status labels* to mentions and entities. Then, VOP requires to identify inclusion and transition relations between nodes of the "visited" entities.

For the second issue, we have constructed a dataset for training and evaluating trajectory extraction systems: Arukikata Travelogue Dataset with Visit Status and Visiting Order Annotation (ATD-VSO).[1] Our dataset comprises 100 travelogue documents annotated with the corresponding visiting order graphs, totally including 3,354 geo-entities (nodes) and 3,369 relations (edges).

Using this dataset, we have trained and evaluated masked and causal language model-based systems. Notable findings through the experiments are (i) that the systems can achieve relatively high accuracy for predicting visit status labels and transition relations, and (ii) that the systems failed to accurately predict inclusion relations. The latter implies an important future issue, i.e., how to inject the knowledge of geographic hierarchical structure into the systems.

Contributions For the purpose of building a foundation for future studies, we have made two main contributions: (i) the proposal of visiting order graph and (ii) the construction of a benchmark dataset for the trajectory extraction.[2] We will release our code and dataset for research purposes. We expect that our dataset will foster continued growth in the trajectory extraction research.

[1]We will release our dataset at https://github.com/naist-nlp/atd-vso.

[2]Our contributions are in the data resource direction, not the technical one such as algorithm and model sophistication. On top of the resource, we will make technical contributions in the future.

2 Preliminaries for Data Construction

Our dataset, ATD-VSO, has been constructed on the basis of Arukikata Travelogue Dataset with geographic entity Mention, Coreference, and Link annotation (ATD-MCL) (Higashiyama et al., 2024).[3] ATD-MCL is a Japanese travelogue dataset annotated with three types of geo-entity information, namely, mentions, coreference relations, and links to geo-database entries, to a collection of the original travelogues, the Arukikata Travelogue Dataset (ATD) (Arukikata. Co., Ltd., 2022; Ouchi et al., 2023).[4]

Annotated mentions in ATD-MCL include proper nouns (e.g., "Nara station"), general noun phrases (e.g., "the station"), and deictic expressions (e.g., "there") that refer to various types of locations, such as geographic regions, facilities, and landmarks. Moreover, a set of mentions that refer to the same location constitutes a coreference cluster or geo-entity. Given such annotated travelogues, we focus on annotating the visit status and visiting order of candidate geo-entities.

3 Visit Status Prediction

We propose a task comprising the two subtasks: Visit Status Prediction (VSP) and Visiting Order Prediction (VOP). This section describes the task of VSP, where a visit status is predicted for each location. For example, it can be judged that the traveler visited the station from the description of the real experience: "Arrived at Kintetsu Nara Station!" In contrast, the factual statement, "JR Nara Station is a little far from Kintetsu Nara Station." does not indicate that the traveler visited these locations. In this task, we aim to distinguish such differences and identify locations visited by travelers.

3.1 Annotation Data Construction

We defined two types of visit status labels in Table 1 for entities and six types of visit status labels in Table 2 for mentions. The mention labels serve to distinguish detailed status of the mentioned location based on the context, i.e., the sentence where the mention occurs. The entity labels serve to determine whether the traveler eventually visited the location, considering the entire document. As annotation work, native Japanese annotators at a data annotation company assigned visit status labels to each mention and entity in ATD-MCL travelogues

[3]http://github.com/naist-nlp/atd-mcl

[4]https://www.nii.ac.jp/dsc/idr/arukikata/

1	`Visit`	A visit to the location is stated or implied.
2	`Other`	Not 1.

Table 1: Visit status labels for entities.

1	`Visit`	The same as the entity label 1.
2	`PlanToVisit`	It mentions a plan to visit the location during this trip (described in the travelogue).
3	`See`	Not any of 1–2, and that the traveler saw the location can be identified.
4	`Visit-Past`	Not any of 1–3, and it mentions having visited the location before this trip.
5	`Visit-Future`	Not any of 1–3, and it mentions the intention to visit the location after this trip.
6	`UnkOrNotVisit`	The visit to the locations cannot be identified from the descriptions, or the non-visit can be identified.

Table 2: Visit status labels for mentions.

according to the label definitions and annotation guideline.[5]

Inter-Annotator Agreement We requested two annotators to independently annotate five documents. We then measured the inter-annotator agreement (IAA) using F1-score and Cohen's Kappa κ. The obtained scores suggest the high agreement: F1 score of 0.80 and κ of 0.68 for 180 mentions, and F1-score of 0.89 and κ of 0.81 for 124 entities.

Data Statistics The annotators annotated additional 95 documents (one annotator per document); the total became 100 documents, including the aforementioned five documents, as shown in Table 3 and Table 4.

3.2 Task Definition

Entity-level and mention-level VSP are defined as follows. Given a set of entities $\mathcal{E}$ in an input document, entity-level VSP requires a system to assign an appropriate visit status label $y \in \mathcal{L}_e$ for each entity $e_q \in \mathcal{E}$. Similarly, given an entity (or coreference cluster) $e_q = \{m_1^{(q)}, \ldots, m_{|e_q|}^{(q)}\}$, which consists of one or more mentions, mention-level VSP requires a system to assign an appropriate visit status label $y \in \mathcal{L}_m$ for each mention $m_i^{(q)} \in e_q$.

3.3 VSP System Framework

As the framework for VSP, we employ a two-step method that first predicts mention labels and then predicts entity labels based on the mention labels. Specific systems under this framework are described in §5.4. Specifically, we calculate the label probability distribution $P(y|m_i^{(q)})$ for each mention $m_i^{(q)} \in e_q$, and select the most probable label $\hat{y}_i^{(q)}$:

$$\hat{y}_i^{(q)} = \arg\max_{y \in \mathcal{L}_m} P(y|m_i^{(q)}).$$

Then, we select a label for each entity e_q according to the following mention label aggregation (MLA) rules.

1. If `Visit` or `PlanToVisit` has been assigned to at least one mention in e_q, then `Visit` is assigned to e_q.
2. Otherwise, `Other` is assigned to e_q.

[5]The annotators used the brat annotation tool (Stenetorp et al., 2012) (`https://github.com/nlplab/brat`).

Set	#Doc	#Sent	#Men	#Ent	#Inc&Tra
Train	70	4,254	3,782	2,339	2,343
Dev	10	601	505	316	329
Test	20	1,469	1,102	699	697
Total	100	6,324	5,389	3,354	3,369

Table 3: Statistics of the ATD-VSO.

Set	`Visit`	`Plan`	`See`	`Past`	`Future`	`UN/O`
Mention						
Train	2,577	358	212	10	6	619
Dev	332	48	46	1	4	74
Test	748	121	59	10	4	160
Entity						
Train	1,942	–	–	–	–	397
Dev	252	–	–	–	–	64
Test	575	–	–	–	–	124

Table 4: Numbers of visit status labels for mention level (top) and entity level (bottom). `Plan`, `Past`, and `Future` indicate `PlanToVisit`, `Visit-Past`, and `Visit-Future`, respectively. `UN/O` indicates `UnkOrNotVisit` for mention level and `Other` for entity level.

4 Visiting Order Prediction

This section describes the task of VOP, where geographical and temporal relations between visited locations are predicted.

4.1 Visiting Order Graph

We introduce a visiting order graph that can represent non-linear relations of visited locations. In a graph, nodes correspond to entities, i.e., locations, and edges correspond to relations between

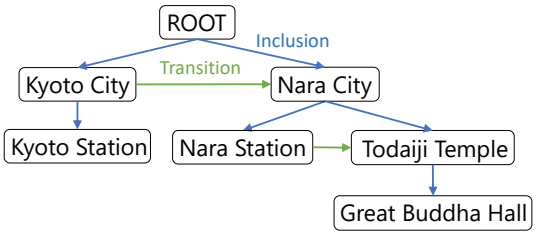

Figure 2: Example of a visiting order graph, the same example at the bottom of Figure 1.

entities, as shown in the example in Figure 2. A directed edge ($\rightarrow$) of inclusion relation represents that the starting node geographically includes the ending node. A directed edge ($\rightarrow$) of transition relation indicates that the traveler visited the starting node entity and then visited the ending node entity, without visiting any other entities in between. We describe further details on these relations in the following paragraphs.

Inclusion Relation Consider the example document in Figure 1, which describes that the traveler visited both "Nara City" and "Todaiji Temple." Based on the geographical fact that the region of "Nara City" includes that of "Todaiji Temple," it is reasonable to interpret that the traveler visited the temple and thereby also visited the city simultaneously. We introduce inclusion relation $\langle e_1, e_2 \rangle$, where an entity e_1 geographically includes another e_2. From Figure 2, we describe two examples:

$$p_1 = \langle \text{Nara City}, \text{Todaiji Temple} \rangle,$$
$$p_2 = \langle \text{Todaiji Temple}, \text{Great Buddha Hall} \rangle.$$

Here, p_1 represents "Nara City" includes "Todaiji Temple", and p_2 represents "Todaiji Temple" includes "Great Buddha Hall." Also, these two relations imply a hierarchical relation: "Nara City" is a grand parent of "Great Buddha Hall."

Transition Relation Given a set of entities for a document and inclusion relations among them, we assign transition relation to each pair of preceding and subsequent visited entities. Notably, we restrict an entity pair with transition relation to two entities with the same parent entity. In Figure 2, while "Nara Station" and "Todaiji Temple" have the same parent node, "Kyoto Station" and "Nara Station" does not. Therefore, the transition relation can be assigned to $\langle$Nara Station, Todaiji Temple$\rangle$, but cannot be assigned to $\langle$Kyoto Station, Nara Station$\rangle$. This restriction enables determining the order of visits for any entity pairs by traversing transition and inclusion relations, even if entity pairs are not directly related to each other. For example, although "Kyoto Station" does not have transition relation to "Nara City," you can interpret "Kyoto Station" was visited before "Nara City" because the parent "Kyoto City" has transition relation to "Nara City."[6]

Set	Inclusion	Transition
Train	1,302	1,041
Dev	186	143
Test	375	322

Table 5: Statistics for visiting order annotation.

4.2 Annotation Data Construction

After the annotation step of visit status, we left only the entities with the `Visit` or `VisitPossibly` label as the nodes of a visiting order graph. In the annotation step of the relations, annotators assigned the visiting relations between the entities.[7]

Inter-Annotator Agreement We requested two annotators to independently annotate the same five documents as those used for visit status annotation. We then measured the IAA using F1-score. The obtained F1 scores suggest the moderate or high agreement: 0.94 for inclusion, 0.74 for transition, and 0.85 for both.

Data Statistics The 95 documents assigned visit status were divided among multiple annotators, and each annotator annotated each document. The total became 100 documents with 1,856 inclusion relations and 1,494 transition relations, including the five aforementioned documents (Table 5).

4.3 Task Definition

The task of VOP can be divided into two subtasks: Inclusion Relation Prediction (IRP) and Transition Relation Prediction (TRP).

Inclusion Relation Prediction Given a set of entities $\mathcal{E}$ in a document, IRP requires a system to determine the parent entity for each entity $e_q \in \mathcal{E}$ from the set of candidate entities $\mathcal{P}^{(q)}_{\text{cand}} = \mathcal{E} \setminus$

[6]The two relations cover most of trajectories in the dataset, but not all. We introduce a few other criteria described in Appendix A.1.

[7]As the annotation tool for entity relations, we adopted the online whiteboard service, Miro (`https://miro.com/`), and the annotators drew arrows representing relation edges between boxes representing entity nodes using the graphical interface.

$\{e_q\} \cup \{\texttt{ROOT}\}$. In other words, if $e \in \mathcal{P}^{(q)}_{\text{cand}}$ is predicted as the parent entity for e_q, it represents that e includes e_q. The pseudo parent node ROOT should be predicted when the entity of interest has no parent entities.

Transition Relation Prediction Given a set of entities $\mathcal{E}$ in a document, TRP requires a system to determine the entity subsequently visited for each entity $e_q \in \mathcal{E}$ from the candidate set $\mathcal{S}^{(q)}_{\text{cand}}$ with the same parent as e_q:

$$\mathcal{S}^{(q)}_{\text{cand}} = \{e_k \in \mathcal{E} \mid \text{Par}(e_k) = \text{Par}(e_q)\} \cup \{\texttt{EOS}\}.$$

Here, $\text{Par}(e)$ represents the parent entity of e, and the pseudo subsequent node EOS represents that the entity of interest has no subsequent entities.

4.4 VOP System Framework

For the two VOP subtasks, we adopt the following framework. Specific systems under this framework are described in §5.4. Specifically, for IRP and TRP, we select the most probable entity as the parent entity $\hat{e_p}$ or the subsequent entity $\hat{e_s}$ from the corresponding candidate set based on score function $\text{score}_{\text{par}}$ or $\text{score}_{\text{sub}}$, respectively:

$$\hat{e_p} = \arg\max_{e' \in \mathcal{P}^{(q)}_{\text{cand}}} \text{score}_{\text{par}}(e_q, e'), \quad (1)$$

$$\hat{e_s} = \arg\max_{e' \in \mathcal{S}^{(q)}_{\text{cand}}} \text{score}_{\text{sub}}(e_q, e'). \quad (2)$$

Sequence Sorting Decoding In TRP, all nodes under the same parent node (i.e., in the same hierarchy) should be arranged in a single sequence. However, Equation 2 does not always generate a single sequence. To address this issue, we propose a sequence sorting decoding, which has the constraint that all nodes in the same hierarchy result in a single sequence. We describe the details in Appendix B.1.

5 Experimental Setup

For the visit status prediction (VSP) task (§3.2) and the visiting order prediction (VOP) subtasks (§4.3)—inclusion relation prediction (IRP) and transition relation prediction (TRP)—we evaluated the performance of three types of systems: rule-based systems, classification-based Masked Language Models (MLM), and generation-based Causal Language Models (CLM).

5.1 Data Split

As shown in Table 3, we split the 100 documents in ATD-VSO into training, development, and test sets at a ratio of 7:1:2.

5.2 Task Settings

We adopted the settings where gold standard labels of preceding tasks were given, and evaluated systems for each task independently. That is, systems take as input gold entities for VSP and IRP, and gold visited entities (that have `Visit` or `VisitPossibly` labels) and gold inclusion relations for TRP.

5.3 Evaluation Metrics

For VSP, we measured the accuracy of predicted labels for input entities. For IRP, we measured the F1 score for extracting inclusion entity pairs from input entities. For TRP, we measured the F1 score for extracting transition entity pairs, excluding pairs where the subsequent entity is EOS, from input entities.

5.4 System Implementations and Model Training

We constructed MLM and CLM-based systems under the frameworks described in §3.3 and §4.4. As the backbones for the MLM-based systems, we used Japanese LUKE[8] (Yamada et al., 2020) and multilingual LUKE[9] (Ri et al., 2022). As the backbones for the CLM-based systems, we used two pretrained models—Llama-3-ELYZA-JP-8B (ELYZA)[10] (Hirakawa et al., 2024) and Llama-3-Swallow-8B-v0.1 (Swallow)[11] (Fujii et al., 2024; Okazaki et al., 2024)—both of which were continually pretrained from Llama 3 (Grattafiori et al., 2024).

LUKE We constructed our LUKE-based systems by fine-tuning a pretrained LUKE model with each task's training set, using multilingual LUKE with `LukeForEntityClassification`[12] for VSP (mention-level) and Japanese LUKE with `LukeForEntityPairClassification` for the VOP subtasks (IRP and TRP). For VSP, the input comprises a sentence containing a mention of interest and the mention's position (character offsets). For IRP and TRP, the input comprises, for an entity of interest and a candidate entity, the context

[8] https://huggingface.co/studio-ousia/luke-japanese-base

[9] https://huggingface.co/studio-ousia/mluke-large-lite

[10] https://huggingface.co/elyza/Llama-3-ELYZA-JP-8B

[11] https://huggingface.co/tokyotech-llm/Llama-3-Swallow-8B-v0.1

[12] https://huggingface.co/docs/transformers/model_doc/luke

and the positions of each entity's representative mention; the context is formed by concatenating the sentences containing each of the two representative mentions and all intervening sentences in their original order.[13] Representative mentions are selected as follows. For IRP, proper noun mentions are prioritized over other mentions. For TRP, mentions with visit status label of higher confidence (`Visit` > `See` > other labels) are prioritized. Unless otherwise specified, we report the mean accuracy or F1 score on the test set of five runs with different random seed values for the baseline system for each task.

ELYZA and Swallow We constructed our CLM-based systems by zero-shot in-context learning (ICL) and Low-Rank Adaptation (LoRA) (Hu et al., 2021) fine-tuning[14] with the training set. Specifically, for each task the system performs either multi-class classification over mentions or binary classification over entity pairs. To do that, we designed task-specific prompts. For VSP (mention-level), a system is prompted to answer whether an entity was visited. For IRP, a system is prompted to answer whether a location is within another. For TRP, a system is prompted to answer whether one location was visited immediately after another. We used the same prompt for the zero-shot ICL and LoRA fine-tuned systems in each task. The full prompt texts are described in Table 16 in Appendix B.2. Representative mentions for entities were selected in the same way as in LUKE. The fine-tuning was conducted on top of their base models using the Hugging Face PEFT library with a QLoRA-style configuration (Dettmers et al., 2023). We describe more detailed settings, including hyperparameters, in Appendix B.2.

6 Experimental Results

6.1 Results for Visit Status Prediction

Systems We evaluated a rule-based system (ML: Majority Label), an MLM-based system (LUKE), and two CLM systems (ELYZA and Swallow). For entity-level prediction, MLA (§3.3) was applied to each system's mention-level output. The ML rule always outputs the most frequent label, `Visit`, for both mention and entity levels.

Method	Mention	Entity
ML rule	0.629	0.790
ELYZA (0-shot ICL)	0.633	0.810
Swallow (0-shot ICL)	0.661	0.828
LUKE (fine-tuned)	0.750	0.838
ELYZA (fine-tuned)	0.813	0.881
Swallow (fine-tuned)	**0.823**	**0.896**

Table 6: System performance (accuracy) for visit status prediction (left: mention-level, right: entity-level).

Tag	Mention			Entity		
	P	R	F1	P	R	F1
LUKE (fine-tuned)						
`Vis`	0.785	0.924	0.849	0.869	0.950	0.908
`Pla`	0.706	0.688	0.696	–	–	–
`See`	0.655	0.661	0.657	–	–	–
`Pas`	0	0	0	–	–	–
`Fut`	0	0	0	–	–	–
`U/O`	0.611	0.403	0.482	0.650	0.495	0.561
Swallow (fine-tuned)						
`Vis`	0.868	0.919	0.893	0.912	0.942	0.926
`Pla`	0.814	0.727	0.768	–	–	–
`See`	0.672	0.694	0.683	–	–	–
`Pas`	0.625	0.500	0.555	–	–	–
`Fut`	0	0	0	–	–	–
`U/O`	0.639	0.531	0.580	0.666	0.693	0.679

Table 7: Precision (P), recall (R), and F1 scores of LUKE and Swallow (mention-level) and LUKE+MLA and Swallow+MLA (entity-level) for each label of visit status prediction. The tag column indicates the following labels: `Vis` (`Visit`), `Pla` (`PlanToVisit`), `See`, `Pas` (`Visit-Past`), `Fut` (`Visit-Future`), and `U/O` (`UnkOrNotVisit` for mention level and `Other`).

Main Results Table 6 shows the performance of the evaluated systems for mention-level and entity-level VSP. The ML rule that always outputs the `Visit` label for every mention achieved good accuracy. This indicates the imbalance in label distribution with a majority of `Visit` instances, which aligns with the intuition that visited locations are often mentioned in travelogues. Under ICL, the CLMs yielded slightly better accuracy than the ML rule, but the fine-tuned models achieved even higher accuracy. In particular, both fine-tuned CLMs outperformed LUKE, reaching mention-level accuracy above 0.8 and entity-level accuracy near 0.9.

[13] For example, when "Nara City" and "Todaiji Temple" in the Japanese translations of the first three sentences in Figure 1 are entities of interest, the input text is as follows: "<s>That day, I headed to <ent>Nara City<ent> via Kyoto City.</s><s>. . . </s><s>I got off at Nara Station and walked a bit to <ent2>Todaiji Temple<ent2> from there.</s>".

[14] LoRA is a parameter-efficient fine-tuning method that introduces low-rank matrices into selected layers of a pretrained model, allowing efficient adaptation with a small number of trainable parameters.

Method	All	Par=ROOT	Par≠ROOT
Random	0.043	0.057	0.038
Flat	0.244	1	0
ELYZA (0-shot ICL)	0.183	0.443	0.143
Swallow (0-shot ICL)	0.129	0.268	0.119
LUKE (fine-tuned)	0.355	0.058	0.425
ELYZA (fine-tuned)	0.497	**0.561**	0.454
Swallow (fine-tuned)	**0.514**	0.547	**0.474**

Table 8: System performance (F1 score) for inclusion relation prediction. All indicates the performance for all entities. "Par=ROOT" and "Par≠ROOT" indicate the performance for entities whose gold parent are or are not ROOT.

Method	All	Fwd.	Rev.
Random	0.190	0.247	0.061
Occurrence Order (EM)	0.730	0.773	0
Occurrence Order (VS)	0.758	0.794	**0.386**
ELYZA (0-shot ICL)	0.208	0.275	0.034
Swallow (0-shot ICL)	0.227	0.294	0.079
LUKE (fine-tuned)	0.748	0.796	0.366
ELYZA (fine-tuned)	**0.763**	**0.828**	0.346
Swallow (fine-tuned)	0.742	0.811	0.329

Table 9: System performance (F1 score) for transition relation prediction. All indicates the performance for all entities. Fwd. and Rev. indicate the performance for entities whose gold subsequent entities occurred after or before the entities of interest in documents, respectively, regarding their earliest mentions.

Label-Wise Performance Table 7 shows the performance of the LUKE-based and Swallow-based systems for each label. The results can be summarized as follows. First, both systems achieved high performance for the Visit label, with F1 scores ranging from approximately 0.85 to 0.9 at both the mention and entity levels. Second, both systems showed moderate performance for the UnkOrNotVisit/Other label. Notably, Swallow outperformed LUKE on this label. At the entity level, Swallow achieved an F1 score of around 0.68. Nevertheless, there remains room for improvement, as predicting this label is challenging due to limited context and the frequent absence of explicit cues indicating whether a location was visited.

6.2 Results for Inclusion Relation Prediction

Systems We evaluated two rule-based systems (Random and Flat), an MLM-based system (LUKE), and two CLM systems (ELYZA and Swallow). Random indicates a method that randomly selects the parent entity from the candidate set for each entity. Flat indicates a rule-based method that always selects ROOT as the parent entity for an arbitrary entity.

Main Results Table 8 shows the performance (F1 score) of the evaluated systems for IRP. Flat, which is a rule always predicting ROOT as a parent, exhibited the better performance than Random (F1 of 0.244 vs 0.043), suggesting that predicting ROOT can be a reasonable strategy when systems do not have knowledge for specific entities. The zero-shot CLMs yielded poor performance, with F1 scores below 0.2. LUKE outperformed the zero-shot CLM systems, but its performance was highly unbalanced: while it performed relatively well for Par≠ROOT cases to some extent, it almost failed on Par=ROOT cases (F1 of about 0.1). The fine-tuned CLM systems achieved even higher performance. These models demonstrated more balanced performance on both Par=ROOT and Par≠ROOT cases. However, the absolute performance remains modest (F1 of around 0.5), highlighting the difficulty of the task and the need for additional modeling strategies.

Discussion The current MLM- and CLM-based systems have a limitation: their absolute overall performance (maximum F1 of 0.5 at best) has not yet reached a practical level. Probable reasons are that (1) the backbone models pretrained on general infilling or generation tasks did not learn geographic relations among specific geo-entities, and (2) it was difficult to obtain generalized knowledge on geographic relations between entities from fine-tuning only with text-based features. Possible solutions include (a) pretraining with geospatial information like GeoLM (Li et al., 2023), (b) fine-tuning a model with geocoding-based features, such as coordinates and shapes of entities, or (c) incorporating structured knowledge from external resources such as Wikidata.[15] Specifically, Wikidata encodes hierarchical geographic relations—such as city-district and district-landmark—via properties like P131 ("located in the administrative territorial entity").[16] These relations can be leveraged to further pretrain models or to provide explicit features during downstream task fine-tuning or ICL.

[15] https://www.wikidata.org/

[16] https://www.wikidata.org/wiki/Property:P131

6.3 Results for Transition Relation Prediction

Systems We evaluated three rule-based systems (Random and two variants of Occurrence Order), an MLM-based system (LUKE), and two CLM systems (ELYZA and Swallow). Random is a rule-based system that randomly lines up candidate entities for each set of entities with the same parent entity. Occurrence Order arranges candidate entities in the order of occurrence of each representative mention in their document; whereas the *early mention* (EM) strategy uses the earliest occurrence mention as the representative mention, the *visit status* (VS) strategy prioritizes mentions based on visit status label similarly to LUKE (§5.4).

Main Results Table 9 presents the performance of the evaluated systems for TRP. The Occurrence Order variants achieved strong results, with F1 scores of 0.730 (EM) and 0.758 (VS). This aligns with the intuition that the order of location mentions in the text often corresponds to the visiting order. The zero-shot CLM systems, similar to their performance in the IRP task, exhibited poor performance. In contrast, the fine-tuned models, including ELYZA, LUKE, and Swallow, demonstrated similar levels of overall performance, with F1 scores around 0.75. These results were comparable to the Occurrence Order (VS) baseline, which achieved an F1 of 0.758. Notably, the fine-tuned systems were able to correctly identify some reverse pairs, where the subsequent entity appeared earlier in the text than the preceding one. However, their performance on these challenging cases remained limited, with F1 scores around 0.3, suggesting that there is still room for improvement.

Discussion While the three fine-tuned systems yielded promising results, they still have room for improvement. First, entities' contexts are limited. For LUKE, the vector representation of an entity is constructed from a single representative mention selected by the heuristic rule (§5.4). For CLMs, any intervening text between the two entities' representative mentions that exceeds the context size (512 tokens) is truncated. This would be improved by extending the context to include all mentions for two entities of interest, although an effective method may be necessary to grasp complicated relations among many mentions. Second, the current systems uniformly treat all entity pairs without transition relation as negative instances. However, entity pairs with indirect transition relations, where one entity is visited before the other through one or more intermediate entities, could also be exploited as positive instances for an additional auxiliary task. Incorporating such higher-order dependencies, similar to relative event time prediction (Wen and Ji, 2021), may enable the models to capture more complex visiting patterns and improve the overall accuracy of visit order prediction.

7 Qualitative Analysis

We provide a qualitative analysis based on predictions of LUKE-based systems for the three subtasks: VSP, IRP, and TRP.[17]

7.1 Visit Status Prediction

As Table 7 shows, LUKE tends to fail to correctly predict the `UnkOrNotVisit/Other` label. Our analysis indicates two error tendencies. For the first, consider the following example.

Matsue Shinjiko Onsen Station $^{\text{G:UnkOrNotVisit}}_{\text{S:Visit}}$ *is the final station.*

The gold label for *Matsue Shinjiko Onsen Station* is `UnkOrNotVisit` because this sentence is a factual statement and does not indicate the traveler visited the location, but the system assigned `Visit`. As this example shows, it is sometimes difficult to distinguish a factual statement from the one indicating traveler's visitation. For the second, consider the following example.

This time, I skipped Matsue $^{\text{G:UnkOrNotVisit}}_{\text{S:Visit}}$ *and Yonago* $^{\text{G:UnkOrNotVisit}}_{\text{S:Visit}}$.

This sentence clearly indicates that the traveler did not visit *Masue* and *Yonago* by the verb "*skipped*," but the system assigned `Visit`. As this example shows, the system sometimes fails to correctly understand the meaning of some motion verbs, such as "skip" and "pass on."

7.2 Visiting Order Prediction

Inclusion Relation Prediction The results shown in Table 8 (§6.2) have indicated that IRP is a challenging task. Our analysis reveals that LUKE learned the tendency that prefectures and cities often become parents of some entities, but

[17] In the camera-ready version, we added experimental results for fine-tuned CLMs in response to reviewers' comments; although the fine-tuned CLMs exhibited strong performance, the original submission focused its analysis on LUKE, which was the best-performing model at that time. Qualitative analysis of the CLMs remains future work.

LUKE also sometimes made incorrect predictions, such as a prefecture/city being the parent of another prefecture/city. Consider the following example.

I planned to stay one night in Nagoya $^{\text{G:Plan}}_{\text{P:Plan}}$*, so I left Ise* $^{\text{G:Vis}}_{\text{P:Vis}}$ *even though it was still early.*

LUKE predicted "Nagoya" as the parent of "Ise," although both are cities. This suggests that the model lacks geographic commonsense.

Transition Relation Prediction The results shown in Table 9 (§6.3) have indicated difficulty in predicting reverse-order entity pairs. Consider the following example.

Here is Daiouji Temple $^{\text{G:Vis}}_{\text{P:Vis}}$ *with its mausoleum. I took a taxi because it was far from the station* $^{\text{G:Vis}}_{\text{P:Vis}}$.

While "Daiouji Temple" precedes "the station," these sentences describe that the traveler moved from the station to the temple. Although LUKE tended to predict the correct order of reverse pairs when there were some clues, such as temporal expressions like "before" and "after," the system made incorrect predictions for reverse pairs without salient clues, including the above example.

8 Related Work

8.1 Visit Status Prediction

"Visiting" is one type of human actions or movements, thus our Visiting Status Prediction falls into the category of the NLP research that analyzes actions or movements in text. One major stream of such research is the predicate-centric approach (described in detail in Appendix D). Here, we focus on another stream: the location-centric approach.

Li and Sun (2014) and Matsuda et al. (2018) specified visit status of location-referring expressions in each tweet. In a similar manner, Peterson et al. (2021) specified it in clinical documents. While they focused on the "mention-level" prediction, we focus on the "entity-level" prediction as well. In travelogues, multiple expressions referring to the same location (belonging to the same geo-entity) appear in a document. Some of the mentions referring to the same location could appear with the contexts that indicate the writer actually visited, and the others not. By aggregating such various visit status of the different mentions, you can conclude the visit status of the location (geo-entity).

8.2 Visiting Order Prediction

Many studies have addressed the extraction of location-referring expressions, such as toponyms and place names, and the grounding of them onto a map (Lieberman et al., 2010; Matsuda et al., 2017; Kamalloo and Rafiei, 2018; Wallgrün et al., 2018; Weissenbacher et al., 2019; Gritta et al., 2020; Higashiyama et al., 2024). However, very few studies have focused on geographic *trajectories*, i.e., a temporal-ordered sequence of multiple locations.

There are three exceptional studies on trajectory extraction from text. Ishino et al. (2012) proposed a task to extract the origin, destination and its transportation method, from each disaster-related tweet. Wagner et al. (2023) proposed a task to extract a trajectory from each transcribed testimony. Each one-minute speech was transcribed and categorized into one of the coarse-grained location categories, e.g., "`cities in Austria`" and "`ghettos in Hungary`." Their trajectory is not a detailed movement trajectory of specific locations. Kori et al. (2006) proposed to extract visiters' representative trajectories from blogs. Each trajectory is defined as a sequence of location-referring mentions. The visiting order is defined as the one in which the mentions appear in the text. Beyond the mention-appearing order, we have adopted the faithful visiting order, which aligns with written intentions.

The crucial difference between the three studies and ours is the trajectory representation; while the four studies assumed trajectories as *sequences*, we define them as *graphs*. As discussed in §4.1, because trajectories often cannot be represented as sequences, we adopt graphs to appropriately represent geographic hierarchical relations.

9 Conclusion

In this study, we introduced a visiting order graph to represent non-linear relations among visited locations and constructed an annotated travelogue dataset for extracting graph-structured trajectories. The experiments on our dataset demonstrated the performance of current MLM and CLM-based systems and suggested possible directions for improvement. In the future, we will develop an end-to-end system that extracts the visiting trajectories from each source document and grounds them on a map by linking each location to the corresponding point or area.

Limitations

Language Our ATD-VSO dataset was constructed from the original ATD, which consists of Japanese travelogues. Therefore, our experiments are limited to the Japanese language. We plan to extend our dataset to other languages through manual translation. While our annotation scheme is based on visiting status, visiting order, and geographical hierarchy and is designed to be language-agnostic, creating datasets for other languages and regions may require additional steps. These include collecting travelogues in the target language, adapting label definitions to reflect cultural differences, and ensuring access to suitable map databases.

Geographical Coverage Our ATD-VSO dataset includes locations from all prefectures in Japan, as it was created using travelogues of domestic travels within Japan. We plan to extend our dataset to include locations from various countries and areas around the world by using travelogues of overseas travels in the original ATD.

Dataset Size Our ATD-VSO dataset consists of 100 annotated travelogues, which is relatively small due to the high annotation cost and effort required to ensure data quality. Despite its size, the dataset is valuable for its novel graph-based trajectory representation and public availability, promoting reproducibility and fair benchmarking. We plan to expand the dataset in the future to cover more regions and languages.

Source Diversity and Generalizability Our dataset was built entirely from travelogues on the "Arukikata Travelogue" website. While this limits source diversity, it includes various authors and covers all prefectures in Japan, offering a range of narrative styles. Including data from multiple sources could further enhance diversity, though copyright and licensing issues present challenges.

Causal Language Models The CLMs used in our experiments have three limitations: prompt engineering, learning method, and model size. We used only one prompt per task, leaving a full investigation of prompt effects for future work. We focused on zero-shot ICL but plan to explore few-shot ICL and fine-tuning. Our models had eight billion parameters, and using larger models could improve performance. The comprehensive investigation of performance differences among possible prompts is left for future work.

Optimization of System Performance We performed minimum hyperparameter search for the models due to time and resource limitations. Thus, performing optimized experiments has potential for further performance improvement in these models.

Ethical Considerations

License of Used Resources As for our annotated dataset ATD-VSO, its intended use is for academic research purposes related to information science, similarly to that of the original ATD. The text in our dataset is a subset of the original ATD, and the original data does not contain any information about the travelogue authors. The Arukikata Travelogue Dataset is available via the Informatics Research Data Repository, National Institute of Informatics under specific terms of use.[18] The pretrained mLUKE model is available under the Apache License 2.0. The pretrained Japanese BERT model is available under CC BY-SA 4.0. Llama3-ELYZA and Llama3-Swallow are both available under Meta Llama 3 Community License.[19]

Human Annotation Effort The annotation work was performed by annotators at a professional data annotation company, which determined the number of annotators and payment based on its own estimates. The work involved three annotators, all native Japanese speakers, with an annotation manager overseeing the process. We informed the annotators that the data would be used for future NLP research. The visit status annotation involved four annotators (three men and one woman), and visiting order annotation involved three annotators (two men and one woman), all supported by the same manager. All had prior experience with Japanese text annotation and were in their 30s to 50s. The total annotation time amounted to approximately 150 hours for visit status annotation and 75 hours for visiting order annotation. While more annotators could have been ideal, we followed the company's recommendation to balance budget and coordination effort.

Predicted Results for Real-World Applications Models trained on our dataset may predict incorrect visit status and order, which can lead to inaccurate trajectories. Users should be cautious when applying these models in real-world applications, as such errors may impact outcomes.

[18] https://www.nii.ac.jp/dsc/idr/arukikata/documents/arukikata-policy.html (in Japanese)

[19] https://llama.meta.com/llama3/license/

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This study was supported by JSPS KAKENHI Grant Number JP23K24904.

References

Arukikata. Co., Ltd. 2022. Arukikata travelogue dataset. Informatics Research Data Repository, National Institute of Informatics. https://doi.org/10.32130/idr.18.1.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized llms. arXiv:2305.14314.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual LLM adaptation: Enhancing Japanese language capabilities. In *Proceedings of the First Conference on Language Modeling*, COLM, University of Pennsylvania, USA.

Aaron Grattafiori et al. 2024. The llama 3 herd of models. arXiv:2407.21783.

Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. 2020. A pragmatic guide to geoparsing evaluation: Toponyms, named entity recognition and pragmatics. *Language Resources and Evaluation*, 54:683–712.

Qiang Hao, Rui Cai, Changhu Wang, Rong Xiao, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. 2010. Equip tourists with knowledge mined from travelogues. In *Proceedings of the 19th International Conference on World Wide Web*, pages 401–410. Association for Computing Machinery.

Erum Haris and Keng Hoon Gan. 2021. Extraction and visualization of tourist attraction semantics from travel blogs. *ISPRS International Journal of Geo-Information*, 10(10):710.

Shohei Higashiyama, Hiroki Ouchi, Hiroki Teranishi, Hiroyuki Otomo, Yusuke Ide, Aitaro Yamamoto, Hiroyuki Shindo, Yuki Matsuda, Shoko Wakamiya, Naoya Inoue, Ikuya Yamada, and Taro Watanabe. 2024. Arukikata travelogue dataset with geographic entity mention, coreference, and link annotation. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 513–532, St. Julian's, Malta. Association for Computational Linguistics.

Masato Hirakawa, Shintaro Horie, Tomoaki Nakamura, Daisuke Oba, Sam Passaglia, and Akira Sasaki. 2024. elyza/llama-3-elyza-jp-8b.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. arXiv:2106.09685.

Aya Ishino, Shuhei Odawara, Hidetsugu Nanba, and Toshiyuki Takezawa. 2012. Extracting transportation information and traffic problems from tweets during a disaster. In *The Second International Conference on Advances in Information Mining and Management (IMMM 2012)*.

Ehsan Kamalloo and Davood Rafiei. 2018. A coherent unsupervised model for toponym resolution. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1287–1296, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Hiroshi Kori, Shun Hattori, Taro Tezuka, Keishi Tajima, and Katsumi Tanaka. 2006. Extraction of visitors' typical route and its context from local blogs. *IPSJ SIG Technical Report*, 78 (2006-DBS-140):35–42.

Chenliang Li and Aixin Sun. 2014. Fine-grained location extraction from tweets with temporal awareness. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR'14, page 43–52, New York, NY, USA. Association for Computing Machinery.

Zekun Li, Wenxuan Zhou, Yao-Yi Chiang, and Muhao Chen. 2023. GeoLM: Empowering language models for geospatially grounded language understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5227–5240, Singapore. Association for Computational Linguistics.

Michael D. Lieberman, Hanan Samet, and Jagan Sankaranarayanan. 2010. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *2010 IEEE 26th International Conference on Data Engineering*, pages 201–212. IEEE.

Koji Matsuda, Mizuki Sango, Naoaki Okazaki, and Kentaro Inui. 2018. Monitoring geographical entities with temporal awareness in tweets. In *Computational Linguistics and Intelligent Text Processing: 18th International Conference (CICLing 2017, Revised Selected Papers, Part II 18)*, pages 379–390, Budapest, Hungary.

Koji Matsuda, Akira Sasaki, Naoaki Okazaki, and Kentaro Inui. 2017. Geographical entity annotated corpus of Japanese microblogs. *Journal of Information Processing*, 25:121–130.

Tomoko Ohsuga and Keizo Oyama. 2021. Sharing datasets for informatics research through informatics research data repository (IDR) (in japanese). *IPSJ Transactions on digital practices*, 2(2):47–56.

Naoaki Okazaki, Kakeru Hattori, Hirai Shota, Hiroki Iida, Masanari Ohi, Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Rio Yokota, and Sakae Mizuki. 2024.

Building a large Japanese web corpus for large language models. In *Proceedings of the First Conference on Language Modeling*, COLM, page (to appear), University of Pennsylvania, USA.

Hiroki Ouchi, Hiroyuki Shindo, Shoko Wakamiya, Yuki Matsuda, Naoya Inoue, Shohei Higashiyama, Satoshi Nakamura, and Taro Watanabe. 2023. Arukikata travelogue dataset. arXiv:2305.11444.

Yanwei Pang, Qiang Hao, Yuan Yuan, Tanji Hu, Rui Cai, and Lei Zhang. 2011. Summarizing tourist destinations by mining user-generated travelogues and photos. *Computer Vision and Image Understanding*, 115(3):352–363.

Kelly S Peterson, Julia Lewis, Olga V Patterson, Alec B Chapman, Daniel W Denhalter, Patricia A Lye, Vanessa W Stevens, Shantini D Gamage, Gary A Roselle, Katherine S Wallace, et al. 2021. Automated travel history extraction from clinical notes for informing the detection of emergent infectious disease events: Algorithm development and validation. *JMIR public health and surveillance*, 7(3):e26719.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

James Pustejovsky, Parisa Kordjamshidi, Marie-Francine Moens, Aaron Levine, Seth Dworman, and Zachary Yocum. 2015. SemEval-2015 task 8: SpaceEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 884–894, Denver, Colorado. Association for Computational Linguistics.

James Pustejovsky, Jessica Moszkowicz, and Marc Verhagen. 2012. A linguistically grounded annotation language for spatial information. *Traitement Automatique des Langues*, 53(2):87–113.

James Pustejovsky and Zachary Yocum. 2013. Capturing motion in ISO-SpaceBank. In *Proceedings of the 9th Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation*, pages 25–34, Potsdam, Germany. Association for Computational Linguistics.

Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. 2022. mLUKE: The power of entity representations in multilingual pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7316–7330, Dublin, Ireland. Association for Computational Linguistics.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43:227–268.

Miloš Stanojević and Shay B. Cohen. 2021. A root of a problem: Optimizing single-root dependency parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10540–10557, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Eitan Wagner, Renana Keydar, and Omri Abend. 2023. Event-location tracking in narratives: A case study on holocaust testimonies. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8789–8805, Singapore. Association for Computational Linguistics.

Jan Oliver Wallgrün, Morteza Karimzadeh, Alan M MacEachren, and Scott Pezanowski. 2018. GeoCorpora: Building a corpus to test and train microblog geoparsers. *International Journal of Geographical Information Science*, 32(1):1–29.

Davy Weissenbacher, Arjun Magge, Karen O'Connor, Matthew Scotch, and Graciela Gonzalez-Hernandez. 2019. SemEval-2019 task 12: Toponym resolution in scientific papers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 907–916, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Haoyang Wen and Heng Ji. 2021. Utilizing relative event time to enhance event-event temporal relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10431–10437, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

A Details on Annotation Dataset

A.1 Other Criteria of Visiting Order Graphs

Visiting order graphs defined by the above two types of relations can represent many trajectories, but not all. We further introduce the following criteria.

- Multiple Visits: There may be cases where an entity is revisited after passing through other entities. In such cases, the entity should be split into sub-entities that include the corresponding mentions for each visit, and sub-entities are regarded as nodes in the visited order graph instead of the original entity.
- `UnknownTime`: There may be cases where the timing of the visit to an entity is not specified. In such cases, the entity should be assigned the `UnknownTime` label, and it is excluded from nodes in the visited order graph.
- `Overlap`: There may be cases where two entities are geographically overlapping, but one does not include the other, e.g., "Tokyo Prefecture" and "Honshu" (the main island of Japan). In such cases, the two entities should be assigned the `Overlap` relation, and either entity can be selected as a representative node to be assigned `Inclusion` and `Transition` relations between it and other entities.

A.2 Detailed Dataset Statistics

Detailed statistics for visiting order annotation are shown in Table 10.

Set	Inc	Trans	Overlap	UnkTime	MV
Train	1,302	1,041	38	35	95
Dev	186	143	8	8	16
Test	375	322	5	10	32

Table 10: Detailed statistics for visiting order annotation. `Inc` (`Inclusion`), `Trans` (`Transition`), and `Overlap` indicate the numbers of entity pairs with each relation type. `UnkTime` (`UnknownTime`) indicates the number of entities with the label. MV indicates the number of entities with multiple visits.

B Details on Evaluated Systems

B.1 Sequence Sorting Decoding for the Baseline System

In TRP, all nodes under the same parent node (i.e., in the same hierarchy) should be arranged in a single sequence. However, Equation 2 does not always generate a single sequence. To address this issue, we propose a sequence sorting decoding, which has the constraint that all nodes in the same hierarchy result in a single sequence, as follows.

1. $\mathcal{P}$ is a set of all possible pairs whose nodes are in the same hierarchy.
2. The highest scoring pair $\langle e_a, e_b \rangle$ is selected from $\mathcal{P}$.
3. From $\mathcal{P}$, we exclude the pairs applicable to any of the followings: (i) the order-swapped pair $\langle e_b, e_a \rangle$, (ii) the pair $\langle *, e_b \rangle$, which consists of an arbitrary preceding node and the subsequent node e_b, and (iii) the pair $\langle e_a, * \rangle$, which consists of a preceding node e_a and an arbitrary subsequent node.
4. If transition relations among all the nodes have been determined, terminate the decoding. Otherwise, return to the procedure 2. above.

B.2 Detailed Settings for CLM Systems

We ran the two CLMs on a single GPU server of NVIDIA A100 80GB. In the zero-shot ICL setting, it took less than two hours to complete each task. In the LoRA tuning setting, it took about 1 hour for visit status prediction, about 12 hours for inclusion relation prediction, and about 3 hours for transition relation prediction, respectively. Table 16 shows the prompts for the CLM systems in each task.

At inference time, we adopted a logit-based classification approach, rather than relying on string generation. In this approach, the model is prompted to predict the next token following the input prompt, and we extracted the unnormalized output scores (logits) $\mathbf{z} \in \mathbb{R}^V$ over the vocabulary V for the generated token. These logits represent the model's confidence for each possible output token. We will explain this approach for each task in more detail.

Visit Status Prediction Let z_l denote the logit corresponding to the predefined token ID for label $l \in \mathcal{L}$, where

$$\mathcal{L} = \{\texttt{Visit}, \texttt{PlanToVisit}, \texttt{See}, \texttt{Visit-Past}, \texttt{Visit-Future}, \texttt{NotOrUnkVisit}\}.$$

The unnormalized score z_l is obtained directly from the model's logits for the first generated token. The predicted label $\hat{y}$ is then determined by selecting the label with the highest logit:

$$\hat{y} = \arg\max_{l \in \mathcal{L}} z_l$$

This logit-based decision procedure avoids ambiguity in string decoding.

Inclusion/Transition Relation Prediction Let z_{pos} and z_{neg} denote the logits corresponding to the predefined token IDs for the positive and negative classes, respectively. For example, in IRP, the positive class means that there exists the inclusion relation between `child` and `parent`, and in TRP, the positive class means that there exists the transition relation between `entity` and `candidate_entity`. Based on the logits, we computed the score as:

$$\hat{z} = \begin{cases} 1 & \text{if } z_{\text{pos}} > z_{\text{neg}} \\ 0 & \text{otherwise.} \end{cases}$$

For IRP, based on the score $\hat{z}$, we generated the tree with the highest score as the final result by using the Maximum Spanning Tree algorithm (Stanojević and Cohen, 2021)[20]. For TRP, based on the logit z_pos, we greedily determined the order from first to last.

B.3 Hyperparameters

Table 11 shows the hyperparameter values used in the experiments using LUKE. We specifically selected batch size for each task, but we followed Yamada et al. (2020) and Ri et al. (2022) for the other hyperparameters. We saved the models at the training epoch when the models achieved the best scores on the development sets. The sizes of the models for visit status prediction (VSP), inclusion relation prediction (IRP) and transition relation prediction (TRP) are 253M, 561M and 561M, respectively. Table 12 shows the hyperparameter values used in the zero-shot in-context learning experiments with Llama3-ELYZA and Llama3-Swallow. Table 13 shows the hyperparameter values used in the supervised fine-tuning experiments with LoRA-tuned Llama3-ELYZA and Llama3-Swallow.

Task	Name	Value
VSP	Learning rate	5e-6
	Batch size	16
	Training epochs	10
IRP	Learning rate	5e-6
	Batch size	4
	Training epochs	10
TRP	Learning rate	5e-6
	Batch size	4
	Training epochs	10

Table 11: Hyperparameter values for the LUKE models.

Name	Value
Max new tokens	10
Batch size	1
Decoding	Multinomial Sampling
Temperature	0.6
Top_p	0.9

Table 12: Hyperparameter values for Llama3-ELYZA and Llama3-Swallow.

C Additional Experimental Results

C.1 Analysis on LUKE-based System Variants

To investigate the influence of surface text on learning and prediction of the baseline model for mention-level VSP, we evaluated two additional variants of the LUKE-based system trained with edited input text. That is, (1) mention masking model trained with input text where mention tokens are replaced by `[MASK]` tokens, and (2) mention only model trained with input text where context tokens other than mention tokens are removed. Table 14 shows the performance of the model variants on the development set. Compared to the original baseline, the mention masking model remained slightly lower in accuracy, and the mention only model, while even lower in accuracy, was still able to predict correct labels to some extent. This suggests that the model mainly relied on context information and also used mention information together.

C.2 Pipeline Prediction

We performed pipeline prediction on documents in the development set using the current baseline systems: LUKE+MLA for VSP, LUKE for IRP, and LUKE with sequence sorting decoding for TRP (we simply refer to these systems as "LUKE" in this section). Figure 3 shows gold and predicted visiting order graphs for a document (ID: 00019).

For VSP, LUKE correctly assigned `Visit` or `Other` to 10 out of 13 entities, but misclassified three entities with the gold label `Visit` as predicted label `Other`. These misclassified entities resulted from predictions for three mentions in sentence 009 in Table 15; the MLA rule determined the entity label `Other` according to LUKE's prediction of the mention label `See` for the three mentions. This suggests that the trained model did not grasp the nuanced context, which describes a photo of the facilities ("five-storied pagoda" and "kofukuji Tem-

[20] https://github.com/stanojevic/Fast-MST-Algorithm

Name	Value
LoRA rank (r)	8
LoRA alpha	16
LoRA dropout	0.1
Target modules	`q_proj`, `k_proj`, `v_proj`
Quantization	4-bit (NF4)
Batch size	4
Learning rate	2e-4
Training epochs	5
Optimizer	AdamW
Gradient accumulation steps	2

Table 13: LoRA fine-tuning hyperparameters for Llama3-ELYZA and Llama3-Swallow.

Method	Acc.	Macro F1
LUKE	0.750	0.383
LUKE (mention masking)	0.738	0.373
LUKE (mention only)	0.634	0.151

Table 14: Performance of LUKE variants for mention-level visit status prediction (on the development set).

ple") taken by the traveler and the nearby location ("Sarusawaike Pond").

For IRP, LUKE predicted correct parents for four out of seven entities with the predicted label `Visit` and incorrect parents for the remaining three entities. Two of the failed entities are written with general noun mentions ("bamboo grove" in sentence 019 and "shop" in sentence 021); it is necessary for correct prediction to understand that the geographic relations among these and other entities are not explicitly described, except the context on the traveler's trip to Nara. For correct prediction for another failed entity regarding the mention "Great Buddha" in sentence 005, which refers to Birushana Buddha at Todaiji Temple, geographic knowledge that Todaiji Temple is located in Nara Park is also necessary.

For TRP, LUKE was able to identify no exact entity pairs with correct transition relation. The gold transition sequences are those arranged in the order of occurrence in the document for each hierarchy level (except for entities with `UnknownTime` or `Overlap`), and LUKE also arranged entities in the same manner within the given inclusion hierarchy. This result indicates that accurate prediction of inclusion relation is crucial for accurate prediction of transition sequences.

D Supplementary Related Work

Predicate-Centric Approach to Visit Status Prediction A line of work on spatial information in natural language, such as SPACEBANK, seeks to develop computational models that can recognize, generate and reason about spatial information in natural language, including place names, topological relations, and human movement (Pustejovsky et al., 2012; Pustejovsky and Yocum, 2013; Pustejovsky et al., 2015). Basically, they regarded verbs as the expressions that represent movement and defined MOVELINK for encoding movement information, such as the mover, the goal location, and the goal reachability of the movement. Also, previous work on event and temporal expressions, such as TIMEML (Pustejovsky et al., 2003), and event factuality, such as FACTBANK (Saurí and Pustejovsky, 2009), regarded verbs (predicates) as a trigger of each event and specified attribution information on verbs. Instead of predicates, we specify visit status information on location-referring expressions and geo-entities because it is not rare that movement is expressed without verbs. Consider the following example.

Todaiji Temple. In the main hall, I saw the Great Buddha of Nara. What a majestic statue!
Next, Nara National Museum. I had lunch in the restaurant and looked around the exhibits.

Here, the geographic movement from *Todaiji Temple* to *Nara National Museum* is expressed as scene transition by changing paragraphs. Because this kind of example is not rare in travelogues, we specify necessary information on geographic entities and mentions, instead of predicates.

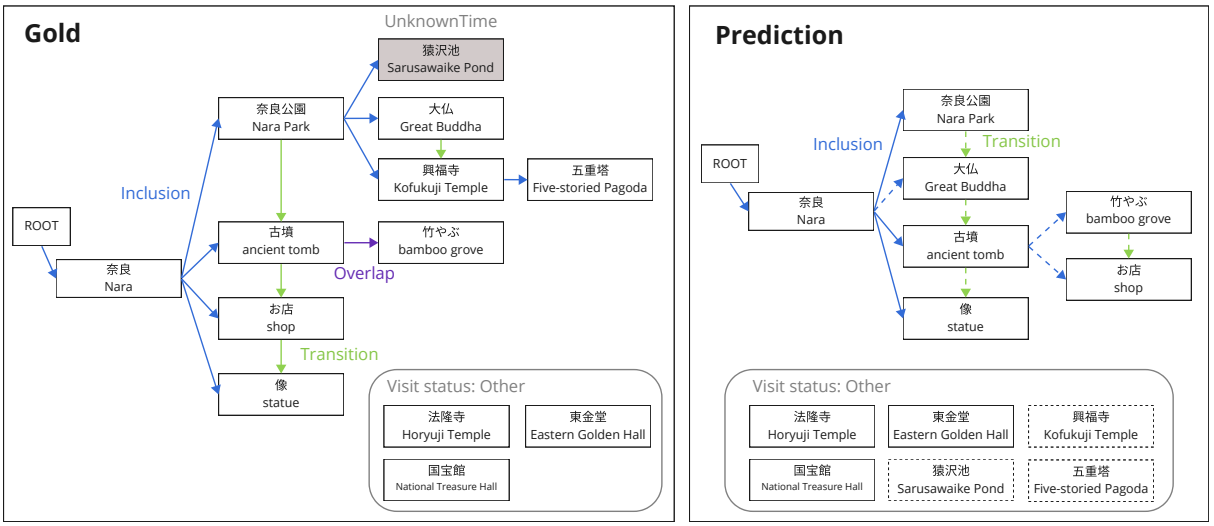

Figure 3: Gold and predicted visiting order graphs for an actual document. The nodes with dashed frames and edges with dashed arrows represent prediction errors.

SentID	Text	English Translation
005	大仏 Visit→Visit 様はとっても大きかったなぁ~	The Great Buddha was really huge.
009	写真は猿沢池 UnkOrNotVisit→See から見える興福寺 Visit→See の五重塔 Visit→See です。	It's a photo of the five-storied pagoda at Kofukuji Temple visible from Sarusawaike Pond.
017,018	写真だとわかりづらいけど、とっても大きな石が使われています。古墳 Visit→Visit の中に入ると、さらに大きさを感じることができます。	
019	竹やぶ Visit→Visit の中にひっそりとあります。	
021	「柿の葉寿司」で有名なお店 Visit→Visit です。	

Table 15: Actual sentences in a document (ID: 00019) and its English translation. Gold mentions are highlighted with blue underline.

Task	Prompt	English Translation
VSP	指示: 文章読解問題です。次の旅行記の文章を読んで、「{mention}」についての質問に回答してください。 文章: {input_text} 質問: 旅行記の著者は「{mention}」を訪れましたか？次の選択肢から1つ選んで、選択肢の番号のみを回答してください。 選択肢: 1 訪問した 2 訪問予定だ 3 その場所を見た 4 前に訪問したことがある 5 将来的に訪問したい 6 その他 回答:	Instruction: This is a reading comprehension test. Read the following travelogue and answer the question on "{mention}." Document: {input_text} Question: Did the author of the travelogue visit "{mention}?" Select one of the following options and answer only its option number. Options: 1 The author visited the place 2 The author plans to visit the place 3 The author saw the place 4 The author had visited the place 5 The author will visit the place in the future 6 Other Answer:
IRP	指示: 文章読解問題です。次の旅行記の文章を読んで、「{child}」と「{parent}」についての質問に回答してください。 文章: {input_text} 質問: 旅行記中の「{child}」は「{parent}」の領域内にありますか？次の選択肢から1つ選んで、選択肢の番号のみを回答してください。 選択肢: 1 はい、領域内にあります 2 いいえ、領域内にはありません 回答:	Instruction: This is a reading comprehension test. Read the following travelogue and answer the question on "{child}" and "{parent}." Document: {input_text} Question: Is the location "child" within the area of "parent" in the travelogue? Select one of the following options and answer only its option number. Options: 1 Yes, it is within the area 2 No, it is not within the area Answer:
TRP	指示: 文章読解問題です。次の旅行記の文章を読んで、「{entity}」と「{candidate_entity}」についての質問に回答してください。 文章: {input_text} 質問: 旅行記の著者は「{entity}」を訪れ、その次に「{candidate_entity}」を訪れましたか？次の選択肢から1つ選んで、選択肢の番号のみを回答してください。 選択肢: 1 はい、次に訪れました 2 いいえ、次に訪れてはいません 回答:	Instruction: This is a reading comprehension test. Read the following travelogue and answer the question on "{entity}" and "{candidate_entity}." Document: {input_text} Question: Did the author of the travelogue visit "{entity}" and then visit "{candidate_entity}" next? Select one of the following options and answer only its option number. Options: 1 Yes, visited next 2 No, did not visit next Answer:

Table 16: Prompts for the CLM systems. "VSP" stands for visit status prediction, "IRP" stands for inclusion relation prediction, and "TRP" stands for transition relation prediction. The phrases {xxx} are variables (place holders).