A Text Embedding Model with Contrastive Example Mining for Point-of-Interest Geocoding

Hibiki Nakatani[♠] **Hiroki Teranishi**[♡,♠] **Shohei Higashiyama**[♣,♠]
Yuya Sawada[♠] **Hiroki Ouchi**[♠,♡] **Taro Watanabe**[♠]
[♠]Nara Institute of Science and Technology [♡]RIKEN
[♣]National Institute of Information and Communications Technology
{nakatani.hibiki.ni4,yuya.sawada.sr7,hiroki.ouchi,taro}@is.naist.jp
hiroki.teranishi@riken.jp,shohei.higashiyama@nict.go.jp

Abstract

Geocoding is a fundamental technique that links location mentions to their geographic positions, which is important for understanding texts in terms of where the described events occurred. Unlike most geocoding studies that targeted coarse-grained locations, we focus on geocoding at a fine-grained point-of-interest (POI) level. To address the challenge of finding appropriate geo-database entries from among many candidates with similar POI names, we develop a text embedding-based geocoding model and investigate (1) entry encoding representations and (2) hard negative mining approaches suitable for enhancing the model's disambiguation ability. Our experiments show that the second factor significantly impact the geocoding accuracy of the model.[1]

1 Introduction

Geocoding is a fundamental technique that identifies the geographic positions, typically, coordinates, of real-world locations from reference expressions (*mentions*) written in natural languages. Geocoding results are useful for accurately understanding where the events in the texts occurred, thereby paving the way for various applications, including tourism management, disaster management, and disease surveillance (Hu et al., 2022).

Geocoding approaches can be classified into two types: (i) *direct positioning* approach and (ii) *linking-based* approach. The direct positioning approach directly identifies the geographic coordinate (or tile) of a location of interest (Gritta et al., 2018; Kulkarni et al., 2021; Huang et al., 2022). The linking-based approach searches in the geographic database (geo-DB) and identifies an entry with its coordinate corresponding to a location (Li et al., 2022, 2023; Zhang and Bethard, 2023; Halterman, 2023; Zhang et al., 2024; Gomes et al., 2024). Most existing studies, regardless of the type of approach, have focused on coarse-grained locations, such as administrative areas, not fine-grained points-of-interest (POIs), such as facilities and landmarks.[2] One probable reason is the limited availability of facility mentions and entries in public geocoding resources. Recent geocoders (Li et al., 2023; Halterman, 2023; Zhang and Bethard, 2023; Zhang et al., 2024; Gomes et al., 2024) have often been developed using popular geocoding datasets, for example, the LGL corpus (Lieberman et al., 2010), TR-News (Kamalloo and Rafiei, 2018), and GeoWebNews (Gritta et al., 2020), which are news text corpora annotated with entries in the GeoNames[3] database. However, news texts typically mention coarse-grained locations more frequently than fine-grained ones, and GeoNames contains a

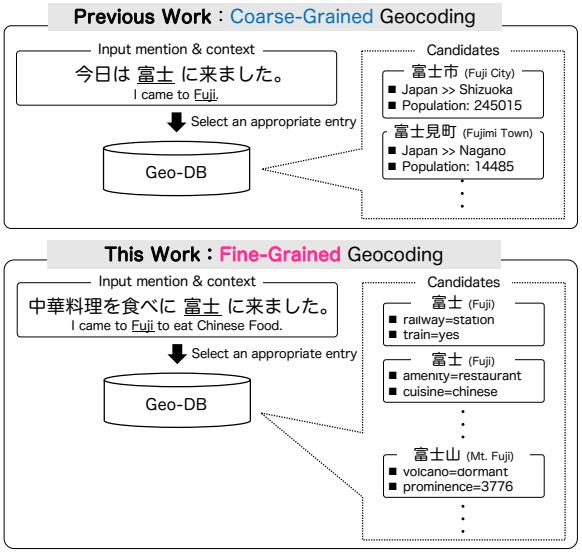

Figure 1: The difference in focus between previous studies and ours: coarse-grained locations for the former and fine-grained POIs, which have many candidates with similar names, for the latter.

[1]We will release our code at https://github.com/naist-nlp/poi-geocoding

[2]We sorely refer to both artificial facilities and natural/historic landmarks as "facilities."

[3]https://www.geonames.org/

Proceedings of the 31st International Conference on Computational Linguistics, pages 7279–7291
January 19–24, 2025.

relatively small number of facility entries.[4]

Detailed activities and events are described in text along with fine-grained POI mentions, thus predicting the geographic positions of such mentions is crucial for achieving practical applications, e.g., tourist spot recommendation. Motivated by this observation, our study addresses POI-level geocoding, which targets facility mentions as well as location mentions. The difference in focus between previous studies and ours is illustrated in Figure 1.

A major challenge in fine-grained POI geocoding is to identify appropriate entries from many facility names with similar strings, containing the same place names, e.g., 富士駅 (Fuji Station), 富士山駅 (Mt. Fuji Station), and 富士山下駅 (Fujiyama-shita Station, meaning Station at the Foot of Mt. Fuji). This also implies that simple approaches based on string match and population heuristics,[5] both of which have been used in many geocoding studies for coarse-grained locations, would not be able to effectively disambiguate candidate entries of fine-grained POIs. Addressing this issue necessitates the development of a geocoder that can distinguish subtle differences among similar entries while carefully considering mention contexts. In this study, we explore a text embedding-based geocoder, focusing on two technical points: (1) how to encode key-value style attributes of geo-DB entries, which typically have no textual descriptions, and (2) how to penalize similar but incorrect entry predictions during model training.

Our experimental findings regarding the geocoder's accuracy include the following: (1) entry vector representations with explicit information of all attributes improved the accuracy to some extent, and (2) hard negative mining approaches significantly impact the accuracy; training with negative examples sampled based on POI name similarity was most effective among the aspects we investigated.

[4]Our preliminary investigation using an existing travelogue dataset (Higashiyama et al., 2024b) shows that GeoNames only covered 40% of the gold entries for randomly-sampled 50 facility mentions, whereas another geo-DB, OpenStreetMap, covered 86% of them.

[5]It is difficult to obtain comparable population indicators for various types of facilities.

2 Text Embedding-Based Geocoder

2.1 Task Definition

We treat geocoding as a task of identifying an appropriate geo-DB entry for each input mention. Formally, given a tokenized document $x = (x_1, \ldots, x_n)$ with a mention span $m = (i_s, i_e)$ within it, where i_s and i_e ($1 \leq i_s \leq i_e \leq n$) indicate the first and last token indices within the span, a geocoding system is required to select an entry e from a geo-DB $\mathcal{E} = \{e_j\}_{j=1}^{|\mathcal{E}|}$.

2.2 Input Representations

We use multilingual E5[6] (Wang et al., 2024a,b) as our backbone text embedding model. This model has been pretrained with contrastive learning on massive multilingual text pairs and has demonstrated strong performance. Notably, multilingual E5 has achieved state-of-the-art results in the Japanese text embedding benchmark.[7] Furthermore, its contrastive learning-based pretraining aligns well with our fine-tuning approach, which is expected to improve performance.

Mention Vector As the text representation of a mention of interest, we use a tokenized text with document context concatenated on both sides of the mention up to the input length limit, $x = (x_1, \ldots, x_n)$,[8] instead of the entire document. Then, the text of n tokens is converted into a hidden vector sequence $\boldsymbol{H}^{(\mathrm{t})} = (\boldsymbol{h}_1^{(\mathrm{t})}, \ldots, \boldsymbol{h}_n^{(\mathrm{t})})$ via a Transformer encoder, and the mention vector is obtained using either of the following ways:

- Average pooling of token vectors over the entire text: $\boldsymbol{h}_m = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{h}_i^{(\mathrm{t})}$,
- Average pooling of token vectors within the mention span: $\boldsymbol{h}_m = \frac{1}{i_e - i_s + 1}\sum_{i=i_s}^{i_e} \boldsymbol{h}_i^{(\mathrm{t})}$.

Geo-Database Entry Vector We assume that geo-DB entries have attributes, each of which corresponds to a key-value pair, e.g., name=興福寺 (Kofukuji temple), prefecture=奈良県 (Nara), and building=temple. Because how to represent entries as a text is non-trivial, we use the following two types of text representations:

[6]https://huggingface.co/intfloat/multilingual-e5-base

[7]https://github.com/sbintuitions/JMTEB/tree/main

[8]Following Wang et al. (2024a), we added the prefix text "query: " and "passage: " to the beginning of the mention and entry text representations, respectively. We set the length limit as 512 in our experiments. Note that x_1 and x_n are special tokens of the beginning and end of the context, respectively.

- Attribute key-value string concatenation with the special separate token [SEP]: e.g., "name=興福寺 [SEP] prefecture=奈良県 [SEP] building=temple",[9]
- Natural language template filled with attribute values: e.g., "興福寺は奈良県に位置しています。 [SEP] building=temple".[10]

Then, the text of ℓ tokens is converted into a hidden vector sequence $\boldsymbol{H}^{(\mathrm{d})} = (\boldsymbol{h}_1^{(\mathrm{d})}, \ldots, \boldsymbol{h}_\ell^{(\mathrm{d})})$ via the encoder, and the entry vector $\boldsymbol{h}_e$ is obtained in similar ways to those for mention vectors, namely, the average pooling of token vectors over the entire text or those within the span of the name value text.

2.3 Candidate Entry Ranking

Given a mention of interest m and the set of candidate entries $\mathcal{E}_m \subseteq \mathcal{E}$,[11] the score s for each candidate entry $e \in \mathcal{E}_m$ is calculated as the inner product between the mention and entry vectors:

$$s(m, e) = \boldsymbol{h}_m \cdot \boldsymbol{h}_e. \quad (1)$$

When predicting top-k entries for a mention, the entries with the top-k scores are selected.

2.4 Training with Negative Examples

Positive training examples are pairs of mentions and their gold entries. For negative examples, several approaches can be used. In this study, we use two types of negatives: in-batch random negatives and hard negatives (Gillick et al., 2019).[12]

In-Batch Random Negatives Assume a training mini-batch $B = \{(m_b, e_b)\}_{b=1}^{|B|} \subset \mathcal{M}_B \times \mathcal{E}_B$, where $\mathcal{M}_B$ and $\mathcal{E}_B$ indicate the set of all mentions and their gold entries in B, respectively. For each mention $m_b \in \mathcal{M}_B$, we use pairs with the gold entries of other in-batch mentions $(m_b, e_{b'})$ $(e_{b'} \in \mathcal{E}_B \setminus \{e_b\})$ as in-batch random negatives.

Hard Negatives As illustrated in Figure 2, we generate hard negatives using a popular sparse lexical search algorithm, BM25 (Robertson et al., 1995),[13] as follows. First, we represent each geo-DB entry as a certain string explained later and create a search index from these "documents." Next, for each training mention, we search and store the top-N results prior to training using the same format string of its gold entry as the "query." During model training, for each epoch, we randomly select an entry from the N saved entries (we call them a *hard negative pool*) for each mention m_i and add the entry as a hard negative e_i^{hard} for the mention. Specifically, we use the following three criteria for hard negative search,[14] focusing on one or more specific attribute types:

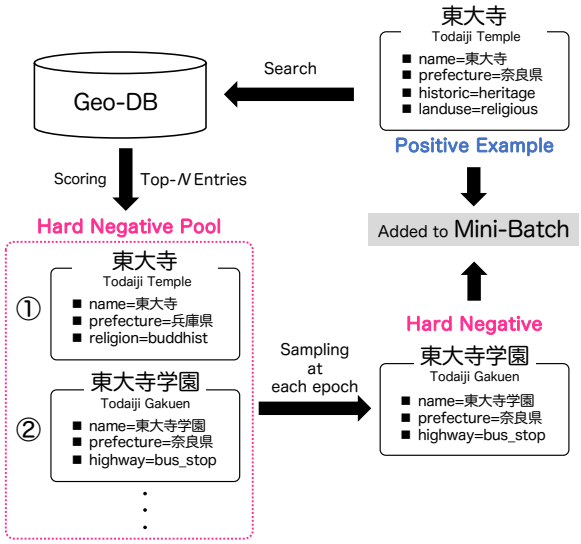

Figure 2: Generation flow of hard negatives.

- Name: A query/document entry is represented as a string where its name value is tokenized by the encoder's tokenizer. For example, an entry whose name is 興福寺 (Kofukuji temple) is converted into a token sequence ["_", "興", "福", "寺"].[15]
- Address: A query/document entry is represented as a string where its address-related attribute[16] values are independently tokenized by the tokenizer and concatenated. For example, an entry with address-related attributes prefecture=奈良県 (Nara Prefecture) and city=奈良市 (Nara City) is converted into

[9] In practice, we replace [SEP] with </s>, which is defined as the separator symbol in the multilingual E5 tokenizer.

[10] The template "{x}は{y}に位置しています。" means "{x} is in {y}." We applied this template only to name and address-related attributes and adopted key-value-style strings for the remaining attributes.

[11] In the experiments, we used the full geo-DB $\mathcal{E}$ as the set of candidate entries $\mathcal{E}_m$ for all mentions.

[12] The purpose of introducing hard negatives is similar to Gillick et al., but our sampling method is different from theirs.

[13] We used an implementation by Lù (2024), BM25-Sparse (https://github.com/xhluca/bm25s).

[14] Regardless of the string representation used here, the entry text representations explained in §2.2 are used for calculating embedding vectors.

[15] The character "_" (U+2581) is the meta symbol that represents a whitespace.

[16] We regard the following attributes as address-related ones: prefecture, city, suburb, quarter, neighbourhood, and road.

Set	Doc	Sent	Mention			
			All	FAC	LOC	LINE
Train	70	4,254	1,544	835	559	150
Dev	10	601	223	133	79	11
Test	20	1,469	457	235	200	22

Table 1: The numbers of documents (Doc), sentences (Sent), and mentions in ATD-MCL.

[“_”, “奈良”, “県”, “_”, “奈良”, “市”].

- Misc: A query/document entry is converted into a token sequence where the key-value strings of its miscellaneous attributes (attributes rather than name and address-related ones) are concatenated. For example, an entry with miscellaneous attributes building=temple and amenity=place_of_worship is converted into [“building=temple”, “amenity=place_of_worship”].
- Mixture of attribute types: A query/document entry is converted into a token sequence that concatenates two or three of name, address, and misc-style token sequences.

The use of a specific type of string representation indicates that entries similar to the gold entries in terms of a targeted attribute type are penalized as negative examples, so that the geocoder does not predict such entries. Thus, hard negatives based on different types of string representations would train the model’s discriminative ability in different directions.

Training Loss During training, we update the encoder’s parameters for each mini-batch B by minimizing the following loss $\mathcal{L}_B$ based on the score s in Eq. (1):

$$\mathcal{L}_B = \frac{1}{|B|}\sum_{b=1}^{|B|}\{-s(m_b, e_b) + \log\sum_{b'=1}^{|B|}\{\exp(s(m_b, e_{b'})) + \exp(s(m_b, e_{b'}^{\text{hard}}))\}\}.$$

3 Experimental Settings

We performed geocoding experiments to investigate the accuracy of our text embedding-based model. In this section, we describe the common setups of our experiments in §3.1–3.4 and present the specific experimental scenarios in §3.5.

3.1 Dataset

We used ATD-MCL[17] (Higashiyama et al., 2024b), which is a Japanese travelogue dataset[18] (Arukikata. Co., Ltd., 2022; Ouchi et al., 2023) annotated with geographic mentions and their corresponding entries of a geo-DB, OpenStreetMap (OSM).[19] We converted the original coreference cluster-level examples[20] with `best_ref_type=OSM` into mention-level examples and targeted only mentions of proper nouns (e.g., “Nara station”) with location (`LOC`), facility (`FAC`), and line (`LINE`) types, excluding mentions of general noun phrases (e.g., “the station”) and deictic expressions (e.g., “there”). As shown in Table 1, this dataset is suitable for a POI geocoding task because it contains the large number of facility mentions.

We adopted the geo-DB preprocessing to group together entries that refer to almost the same real-world locations by assigning the same group ID string, which consists of attribute key-value pairs, following Higashiyama et al. (2024b). This resulted in 1.8M entry groups. Thus, we adopted a setting where entry groups should be predicted as linking units rather than individual entries for given mentions.[21]

3.2 Metrics

We used Mean Reciprocal Rank (MRR) and recall@k (R@k) as the evaluation metrics for the geocoding task, by treating it as an entry ranking problem targeting all entries (entry groups, to be precise) in the geo-DB. The MRR score for q examples is calculated as follows:

$$\text{MRR} = \frac{1}{q}\sum_{i=1}^{q}\frac{1}{\text{rank}(m_i, e_i)},$$

where m_i, e_i, and $\text{rank}(m_i, e_i)$ indicate a mention, its gold entry, and the rank of e_i among all entries based on the model’s prediction scores regarding m_i, respectively. For recall@k, the prediction is regarded as correct if one of the predicted k entries contains the gold entry for each mention.

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

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

[19] https://www.openstreetmap.org/

[20] In the dataset, a set of mentions that refer to the same location constitutes a coreference cluster.

[21] An example of entry group ID: “奈良県|city=奈良市|quarter=樽井町|road=猿沢遊歩道|amenity=cafe” (Starbucks Coffee at Sarusawa pathway, Tarui-cho, Nara City, Nara Prefecture).

	EntRep	NegCrite	R@1	R@5	R@10	MRR
Leven	-	-	0.317	0.588	0.667	0.443
BM25	-	-	0.338	0.618	0.700	0.465
E5	KV	Random	0.465 (±0.031)	0.777 (±0.050)	0.844 (±0.037)	0.602 (±0.033)
	KV	Name & Addr & Misc	<u>0.570</u> (±0.024)	<u>0.827</u> (±0.004)	<u>0.875</u> (±0.004)	<u>0.683</u> (±0.014)
	Template	Random	0.478 (±0.022)	0.788 (±0.026)	0.850 (±0.011)	0.613 (±0.021)
	Template	Name & Addr & Misc	**0.573** (±0.050)	**0.828** (±0.052)	**0.877** (±0.030)	**0.685** (±0.045)

Table 2: Performance of the string similarity baselines and E5-based model with different settings for two representative aspects, i.e., entry text representations (EntRep) and hard negative mining criteria (NegCrite), on the test set. The best scores are indicated in bold, and second-best scores are underlined.

3.3 String Similarity Baselines

We use two types of baseline systems that rank entries based on string similarities: BM25 and the Levenshtein distance. The BM25 baseline scores candidate entries based on each query mention string, where entries are represented by only name values and tokenized by the encoder's tokenizer, in the same manner as that for name-style hard negative sampling in §2.4. The Levenshtein baseline calculates the score based on the Levenshtein distance (Levenshtein, 1966) $d_{\text{Levenshtein}}$ between each mention text and a candidate entry, which is represented by only name value. The normalized Levenshtein distance between mention m and entry e is calculated as follows:

$$\frac{d_{\text{Levenshtein}}(\text{str}(m), \text{str}(e))}{\max(\text{len}(\text{str}(m)), \text{len}(\text{str}(e)))},$$

where $\text{str}(\cdot)$ and $\text{len}(\cdot)$ are functions that return the string representation and length of an argument, respectively.

As the evaluation metrics for these baselines, we calculate the expected recall value following Higashiyama et al. (2024a) and the expected MRR value based on mention-level MRR for mention m_i (MRR_i) as follows:

$$\begin{aligned} \text{MRR}_i &= \frac{1}{|E_i|}\sum_{j=1}^{|E_i|}\frac{1}{\text{rank}(m_i, e_j)}, \\ E_i &= \{e_j \mid s(m_i, e_j) = s(m_i, e_i)\}. \end{aligned}$$

3.4 Model Training Settings

We fine-tuned the pretrained model, multilingual E5 (base), with the hyperparameter settings in Appendix A, and selected the model checkpoint with the best MRR score on the development set. We performed model fine-tuning three times with different random seeds for each setting and report mean scores for the three runs, unless otherwise specified.

3.5 Experimental Scenarios

In our experiments with the text embedding-based geocoder, we focused on two aspects, input representations and hard negative mining, each of which consists of two sub-aspects, as follows:

1. Input representations (§2.2):
   (a) entry text representations, and
   (b) mention/entry vector pooling method.
2. Hard negative mining (§2.4):
   (a) mining criteria, and
   (b) pool size.

Evaluating all combinations of different settings for these aspects is computationally expensive. Thus, we evaluated our model for each aspect on the development set, by varying the settings of one aspect at a time while using the fixed default settings of the other aspects (shown in Appendix B), which were determined based on preliminary experiments. The experiments in §4 show the results of the model with each possible settings for the focusing aspect and default settings for the other aspects, except for the main experiments (§4.1).

4 Results

4.1 Main Results

Table 2 shows the main experimental results (mean ± standard deviation for each metric) on the test set, which includes results of the E5-based model with different settings for representative aspects, i.e., entry text representation (EntRep) and hard negative mining criteria (NegCrite),[22] as well as the results of string similarity baselines. The evaluated settings for the E5-based model include key-value string concatenation (KV) and natural language

[22]For the remaining aspects, we used the best settings on the development set: the mention span average pooling for mention vectors, the entire text average pooling for entry vectors, and 40 for hard negative pool size N.

	EntRep	NegCrite	R@1 FAC	R@1 LOC	R@1 LINE
Leven	-	-	0.335	0.311	0.184
BM25	-	-	0.361	0.302	0.417
E5	KV	Random	0.477	0.470	0.303
	KV	N&A&M	**0.594**	0.552	**0.485**
	Template	Random	0.467	0.490	**0.485**
	Template	N&A&M	0.591	**0.568**	0.424

Table 3: Performance of the baselines and E5-based model for each entity type on the test set. "N&A&M" indicates the mixture criterion of three attribute types, i.e., name, address, and misc.

	Name	Addr	Misc	R@1	R@5	MRR
Random	-	-	-	0.425	0.827	0.598
(a)	✓	-	-	0.601	0.885	0.701
(b)	-	✓	-	0.447	0.813	0.613
(c)	-	-	✓	0.389	0.786	0.566
(d)	✓	✓	-	0.575	0.843	0.700
(e)	✓	-	✓	0.571	0.861	0.701
(f)	-	✓	✓	0.393	0.770	0.554
(g)	✓	✓	✓	**0.649**	**0.891**	**0.755**

Table 4: Performance of the E5-based model with different hard negative mining criteria on the dev set.

template (Template) for EntRep, and the mixture criterion of three attribute types (Name & Address & Misc) for NegCrite. For comparison, we also evaluated the random criterion, which stores random N entries from all candidate entries for each training mention as a hard negative pool.

The results indicate the following three findings. (1) The E5-based model outperformed the string similarity baselines by up to 0.235 points in R@1 and 0.220 points for MRR, demonstrating the strong representational capability of the text embedding model. (2) The selection criteria for hard negatives had a significant impact on performance, for example, approximately 0.1 point improvements in R@1, considering that the random criterion used the same number of negatives as the mixture criterion. (3) Using either of two entry text representations had little impact on performance, which is unsurprising given that both contain equivalent information.

Results of Another Dense Retrieval Model The aspects investigated in this study have room for further exploration using alternative modeling frameworks other than the text embedding model. Thus, we also evaluated another modeling framework: bi-encoder model (Wu et al., 2020) with BERT (Devlin et al., 2019). [23] As described in Appendix D, we observed that the findings for the BERT-based model were almost consistent with those for the E5-based model, particularly demonstrating the importance of the selection criteria for hard negatives.

Results for Each Entity Type Table 3 shows the performance of the same geocoders as those in Table 2 for each entity type. A similar trend in accuracy across methods was observed, as in the overall results in Table 2, except for line, which has only a small number of development examples (i.e., 22). Additionally, the results indicate the following findings. (1) The string similarity baselines achieved better accuracy for facility mentions than for location mentions, indicating there is a higher ratio of gold entries with names similar to mention texts for facility than location. (2) Despite this fact and the large number of facility training examples, the E5-based model trained based on the random criterion yielded similar or worse accuracy for facility mentions than for location mentions, suggesting insufficient learning for facility mentions to identify the correct candidate from among many candidates with similar names.[24] (3) However, the E5-based model trained based on the mixture criterion achieved larger improvements over the random counterpart for facility mentions (0.117–0.124 points in R@1) than for location mentions (0.078-0.082 points), suggesting that the mixture criterion successfully selected hard negatives useful for learning facility examples.

4.2 Comparison of Hard Negative Mining Criteria

We compared possible hard negative mining criteria, that is, all combinations where one or more attribute types were selected from the three attribute types: name, address, and misc. The results on the development set are shown in Table 4.

The results indicate the following three findings. (1) Compared to the random criterion, the criteria without name attributes yielded slight improvements or degradations. (2) The criteria with name attributes (highlighted in pale purple background) achieved significant performance improvements, specifically, 0.146–0.224 point improvements in R@1 over the random criterion and 0.128–0.256

[23] We leave the introduction of the re-ranking step with the cross-encoder for future work.

[24] Actually, gold facility entities have more candidate entries with the same or similar names than gold location entries as discussed in Appendix §C.

Mention	Entry	R@1	R@5	R@10	MRR
Entire	Entire	0.617	0.877	0.921	0.729
Span	Entire	**0.620**	**0.883**	**0.927**	**0.734**
Entire	Span	0.586	0.861	0.906	0.707
Span	Span	0.577	0.877	0.919	0.712

Table 5: Performance of the E5-based model with different vector pooling methods, i.e., the *entire* text average pooling or mention/name *span* average pooling, on the dev set.

point improvements in R@1 over the counterpart criteria without name attributes. (3) Although the criteria with address and misc attributes did not necessarily bring improvements, the criterion of the mixture of three attribute types (g) achieved the best performance.

The key finding (2) suggests that by using hard negatives whose names are similar to those of gold entries, the model had come to focus on mention contexts and other entry attributes while avoiding excessive dependence on mention text and entry names. This is also supported by our observations of the prediction examples on the development set. For example, for a mention 広島駅 (Hiroshima Station), the model trained based on the name criteria correctly predicted the gold entry representing the Hiroshima railway station, with attributes name=広島 (Hiroshima) and railway=station, but the model trained based on the other criteria predicted incorrect entries, e.g., those with attributes name=広島駅 and highway=bus_stop.

4.3 Effect of Hard Negative Pool Size

We investigated the impact of varying the size N of hard negative pools among 10, 20, 40, and 80, which correspond to the expected number of times that each entry in a pool is actually selected as a negative example is 2, 1, 0.5, and 0.25. The results on the development set are shown in Figure 3. Compared to the case where $N = 0$, which corresponds to the model trained only with in-batch random negatives, the model's performance significantly improved in all cases where $N > 0$. Additionally, the best performance was achieved when $N = 40$, and the degraded performance was observed when $N = 80$. This suggests the importance of balancing the diversity of hard negatives and the reasonably high similarity between them and gold entries.

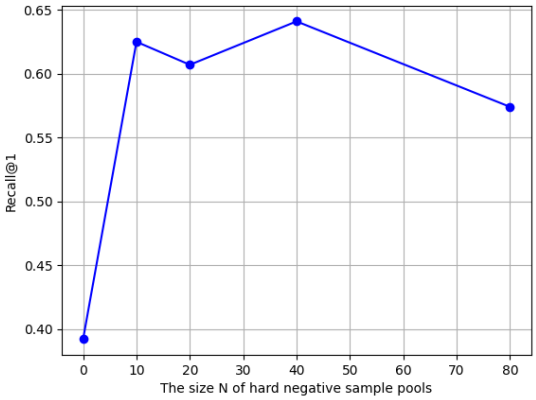

Figure 3: Performance (Recall@1) of the E5-based model trained with hard negatives sampled from hard negative pools with different size N on the dev set.

4.4 Comparison of Vector Pooling Methods

We evaluated the model with possible combinations of vector pooling methods, where average pooling over the entire text or that over mention/name span was used for mention and entry vectors, respectively. The results on the development set are shown in Table 5. The results indicate that the two pooling methods for mention vectors yielded similar performance when those for entry vectors were fixed. In contrast, for entry vectors, pooling over the entire text achieved better performance than the counterpart by 0.031–0.043 points in R@1 and 0.022 points in MRR. This result for entry vectors further indicates that not only name but also other attributes, such as address, are important for distinguishing multiple entries with the same or similar names. For the different trends in results between the pooling methods for mention and entry vectors, we present the following possible explanation. Whereas token vectors within a mention span may already contain information on useful context via Transformer's attention over natural language text, token vectors within an entry name span may not sufficiently contain information on attributes outside the name because of non-optimal attention over attribute key-value pair sequence.

4.5 Qualitative Analysis

We conducted an error analysis on the prediction results for the four model variants in Table 2. Table 6 shows these results for the development examples.

For example [1] of mention 夫婦岩 (Meoto Iwa, meaning the Wedded Rocks), the model trained with hard negatives identified the correct entry in

	Gold Entry	EntRep/NegCrite	Top Predicted Entry	Gold Entry Rank
[1]	夫婦岩 (Ise City, Mie Pref.)	KV/Random	夫婦岩 (Ichinoseki City, Iwate Pref.)	11
		KV/N&A&M	夫婦岩 (Ise City, Mie Pref.)	1
		Template/Random	夫婦岩 (Ichinoseki City, Iwate Pref.)	12
		Template/N&A&M	夫婦岩 (Ise City, Mie Pref.)	1
[2]	大三島 (Imabari City, Ehime Pref.)	KV/Random	大三島環状線	12
		KV/N&A&M	三島市 (Shizuoka Pref.)	13
		Template/Random	大三島橋 (Imabari City, Ehime Pref.)	13
		Template/N&A&M	大島大橋	3
[3]	来島海峡大橋; しまなみ海道 サイクリングロード (Imabari City, Ehime Pref.)	KV/Random	来島海峡大橋 (Imabari City, Ehime Pref.)	2
		KV/N&A&M	来島海峡大橋 (Imabari City, Ehime Pref.)	4
		Template/Random	来島海峡大橋 (Imabari City, Ehime Pref.)	13
		Template/N&A&M	来島海峡大橋 (Imabari City, Ehime Pref.)	89

Table 6: Prediction examples by the E5-based model on the dev set. The mentions are 夫婦岩 (Meoto Iwa), 大三島 (Omishima), and 来島海峡大橋 (Kurushima Kaikyo Bridges) for examples [1], [2] and [3], respectively. The entries without prefecture or municipality information originally lacked that information.

Ise City, Mie Pref by leveraging the surround context about Ise whereas the model trained only with random negatives predicted an incorrect entry with the same name in the different prefecture. This indicates that training with hard negatives enabled the model to focus not only name but also other attributes, such as address-related ones.

For example [2] of mention 大三島 (Omishima) in Ehime Prefecture, all model variants failed to predict the correct entry and some model variants predicted entries in different prefectures. This example is difficult because the input document did not describe any prefectures and municipalities, which is often the case in personal travelogues.

For example [3] of mention 来島海峡大橋 (Kurusima Kaikyo Bridges), all model variants predicted an almost correct entry that represents a *regular* bridge POI. However, the mention is annotated with the gold entry that refers to the same bridge but represents a *bicycle path*, しまなみ海道サイクリングロード (Shimanami Kaido Cycling Road). This example is challenging but not impossible to resolve because the context for the mention includes descriptions related to cycling, which are useful to identify the gold entry, as shown in Figure 6 (Appendix E).

5 Related Work

5.1 Geocoding

There exist two main approaches to geocoding: *direct positioning* and *linking-based* approaches.

Direct positioning approach The direct positioning approach predicts geographic coordinates or grids for given mentions. Gritta et al. (2018) classify mentions into geodesic tiles using neural networks that encode lexical features, such as mentions and their surrounding words, and population information taken from an ontology. Kulkarni et al. (2021) adopt a similar approach but predict hierarchical multi-level regions without relying on gazetteer metadata. For predicting fine-grained POIs, the direct positioning approach necessitates an overwhelming number of classes for small grids.[25] To tackle this problem, Huang et al. (2022) propose a pretraining method for incorporating toponym and spatial knowledge and attempt to predict a character sequence that efficiently encodes the multi-level cells for a POI.[26]

Linking-based approach The linking-based approach identifies the location for a mention by choosing the most suitable entry from a geo-DB. Many prior systems first perform lexical search to collect candidates from a DB and then rank them by using textual features for machine learning models, such as LightGBM (Wang et al., 2019) and neural networks (Zhang and Bethard, 2023; Halterman, 2023; Zhang et al., 2024). Notably, Li et al. (2023) perform contrastive learning to relate linguistic and geospatial contexts for language model pretraining. Gomes et al. (2024) employ a Transformer-based sentence encoder to sort candidate entries by cosine similarity. Our method is similar to those of Li et al. (2023) and Gomes et al. (2024) but differs in (i) not performing candidate generation, (ii)

[25] For instance, the Earth's surface is divided into 105 trillion S2 cells at level 22 (https://s2geometry.io/resources/s2cell_statistics).

[26] Their experiments were conducted on proprietary data.

exploiting geographic attributes rather than geospatial contexts to represent geo-DB entries, and (iii) mining hard negatives using entry attributes as well as names. Particularly regarding (i), our geocoder is not affected by the performance of candidate generation,[27] and vector-based retrieval during inference can be accelerated by pre-indexing entries and using an efficient search algorithm, such as Hierarchical Navigable Small World.

5.2 Negative Sampling for Entity Linking

Linking-based geocoding can be regarded as a special case of Entity Linking (EL). EL typically involves two steps: (i) candidate generation and (ii) reranking, where candidate generation is mainly performed using frequency-based similarity calculations, such as BM25 (Logeswaran et al., 2019) and TF-IDF (Angell et al., 2021).

Bi-encoder models (Gillick et al., 2019; Wu et al., 2020; Humeau et al., 2020; Agarwal et al., 2022), a well-known architecture for EL, often use entries within the batch as in-batch negatives to improve memory efficiency. However, Gillick et al. (2019) demonstrated that incorporating not only in-batch negatives but also hard negatives helps the model learn by leveraging the context of the entry description. Hard negatives are often sampled based on the predictions of models trained only with in-batch negatives (Gillick et al., 2019; Wu et al., 2020) or using the mention-mention similarity graph (Agarwal et al., 2022).

In this study, we employed a shared E5 encoder but adopted a framework similar to that of bi-encoder EL models to encode representations of mentions and entries and calculate the similarity between mention-entry pairs. Our model, trained with hard negatives sampled using BM25 in a manner similar to Logeswaran et al. (2019), exhibited better performance than the counterpart model trained only with in-batch random negatives. The model's performance is expected to improve further with additional training through hard negatives sampled based on the model's own predictions, as demonstrated by Gillick et al. (2019) and Wu et al. (2020). We leave this for future work.

6 Conclusion and Discussion

This paper has presented a text embedding-based geocoding model designed for the POI geocoding task. We explored entry encoding representations and hard negative mining approaches for the model through the extensive experiments, and our model with the best configuration achieved a recall@1 of 0.573 and a recall@10 of 0.877 on the ATD-MCL test set. The recall@1 score indicates that approximately 40% of the top prediction results are incorrect, suggesting room for further improvement. However, the recall@10 score indicates that the top 10 predictions contain the majority of the correct entries. Introducing a detailed reranking step of candidate entries, such as the cross-encoder mechanism (Wu et al., 2020), could potentially lead to more accurate geocoding.

Limitations

Our evaluation is based on a single dataset, which consists of Japanese language travelogues and includes only annotated mentions referring to POIs in Japan. Evaluating geocoding methods on more diverse datasets, with a large number of facility mentions, is necessary to ensure that such methods are applicable across various domains, languages, and POI areas.

Acknowledgments

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

References

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4644–4658, Seattle, United States. Association for Computational Linguistics.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2021. Clustering-based inference for biomedical entity linking. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2598–2608, Online. Association for Computational Linguistics.

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

[27] Gomes et al. (2024) report that the recall rate of candidate generation for toponym resolution was 90.2% for the GeoWebNews dataset.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 528–537, Hong Kong, China.

Diego Gomes, Ross S Purves, and Michele Volpi. 2024. Fine-tuning Transformers for toponym resolution: A contextual embedding approach to candidate ranking. In Proceedings of The GeoExT 2024: Geographic Information Extraction from Texts Workshop, pages 43–51.

Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Which Melbourne? Augmenting geocoding with maps. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1285–1296.

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.

Andrew Halterman. 2023. Mordecai 3: A neural geoparser and event geocoder. arXiv:2303.13675.

Shohei Higashiyama, Masao Ideuchi, and Masao Utiyama. 2024a. Construction of the administrative agency web document corpus for Japanese entity linking [in Japanese]. IPSJ SIG Technical Report, 2024-NL-260(10):1–15.

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. 2024b. 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.

Xuke Hu, Zhiyong Zhou, Hao Li, Yingjie Hu, Fuqiang Gu, Jens Kersten, Hongchao Fan, and Friederike Klan. 2022. Location reference recognition from texts: A survey and comparison. arXiv:2207.01683.

Jizhou Huang, Haifeng Wang, Yibo Sun, Yunsheng Shi, Zhengjie Huang, An Zhuo, and Shikun Feng. 2022. ERNIE-GeoL: A geography-and-language pre-trained model and its applications in Baidu maps. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, pages 3029–3039, New York, NY, USA. Association for Computing Machinery.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In International Conference on Learning Representations.

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.

Sayali Kulkarni, Shailee Jain, Mohammad Javad Hosseini, Jason Baldridge, Eugene Ie, and Li Zhang. 2021. Multi-level gazetteer-free geocoding. In Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics, pages 79–88, Online. Association for Computational Linguistics.

Xing Han Lù. 2024. BM25S: Orders of magnitude faster lexical search via eager sparse scoring. arXiv:2407.03618.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. Soviet physics. Doklady, 10:707–710.

Zekun Li, Jina Kim, Yao-Yi Chiang, and Muhao Chen. 2022. SpaBERT: A pretrained language model from geographic data for geo-entity representation. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2757–2769.

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.

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.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Hiroki Ouchi, Hiroyuki Shindo, Shoko Wakamiya, Yuki Matsuda, Naoya Inoue, Shohei Higashiyama, Satoshi

Nakamura, and Taro Watanabe. 2023. Arukikata travelogue dataset. arXiv:2305.11444.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. Nist Special Publication Sp, 109:109.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Text embeddings by weakly-supervised contrastive pre-training. arXiv:2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Multilingual E5 text embeddings: A technical report. arXiv:2402.05672.

Xiaobin Wang, Chunping Ma, Huafei Zheng, Chu Liu, Pengjun Xie, Linlin Li, and Luo Si. 2019. DM_NLP at SemEval-2018 task 12: A pipeline system for toponym resolution. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 917–923.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6397–6407, Online. Association for Computational Linguistics.

Zeyu Zhang and Steven Bethard. 2023. Improving toponym resolution with better candidate generation, Transformer-based reranking, and two-stage resolution. In Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023), pages 48–60.

Zeyu Zhang, Egoitz Laparra, and Steven Bethard. 2024. Improving toponym resolution by predicting attributes to constrain geographical ontology entries. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 35–44, Mexico City, Mexico. Association for Computational Linguistics.

A Model Hyperparameters

Table 7 shows the hyperparameter values used for fine-tuning the E5-based model.

Hyperparameter	Value
Training epochs	20
Batch size	16
Weight decay	0.01
Adam β_1	0.9
Adam β_2	0.98
Adam ϵ	1e-6
Learning rate	1e-5
Learning rate scheduler	linear
Warmup ratio	0.06
Optimizer	AdamW

Table 7: The hyperparameter values used for the E5-based model.

B Default Model Settings

In the experiments focusing on specific aspects in §4.2–4.4, we used the default settings in Table 8, which were determined based on preliminary experiments, except for settings of the focusing aspects.

Aspect	Setting
Entry text representation	Key-value string concat
Mention vector	Mention span average pooling
Entry vector	Entire text average pooling
Hard negative criterion	Name
Hard negative pool size	20

Table 8: The default settings used for the E5-based model in the experiments in §4.2–4.4.

C Candidate Entry Statistics

Figure 4 shows the distribution over the development examples (mention-entry pairs) of the average number of candidate entries (y-axis values) whose normalized Levenshtein distance to the gold entries' names is less than or equal to the distance of x-axis values (from 0 to 1, in increments of 0.1). This figure can be interpreted as the *generalized degree of ambiguity* of gold entries. Specifically, the y-axis value indicates the number of candidates with the same name as the gold entry when the x-axis value is 0, and the number of candidates with names similar to the gold entry when the x-axis value is greater than 0. We observe that gold facility entries have more candidate entries with the same or similar names than gold location entries.

Similarly to Figure 4, Figure 5 shows the distribution over the development examples of the average number of candidate entries (y-axis values) whose normalized Levenshtein distance to the mention text is less than or equal to the distance of x-axis values (from 0 to 1, in increments of 0.1). From this figure, we observe that, to some extent, there are more candidate entries with names similar to the mention texts for facility examples than location examples when the distance is between approximately 0.3 and 0.9.

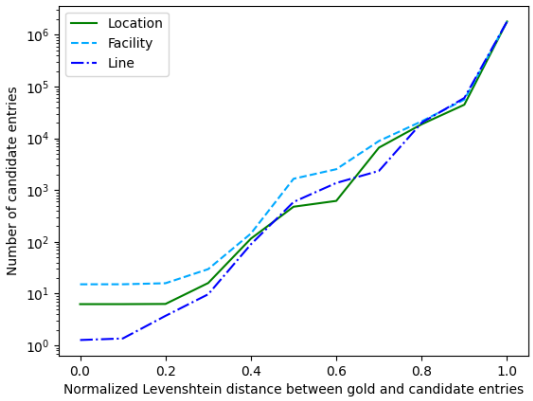

Figure 4: The distribution over the development examples of candidate entries in terms of the normalized Levenshtein distance to the gold entries' names.

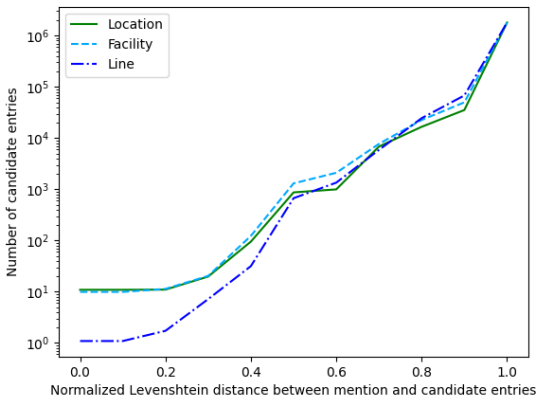

Figure 5: The distribution over the development examples of candidate entries in terms of the normalized Levenshtein distance to the mention texts.

D Experiments with BERT Bi-Encoder

As another dense retrieval model, we developed a geocoding model with Japanese BERT[28] based on the bi-encoder framework (Wu et al., 2020). Unlike

[28] https://huggingface.co/tohoku-nlp/bert-base-japanese-whole-word-masking

	EntRep	NegCrite	R@1	R@5	R@10	MRR
Leven	-	-	0.317	0.588	0.667	0.443
BM25	-	-	0.338	0.618	0.700	0.465
BERT	KV	Random	0.416 (±0.043)	0.729 (±0.070)	0.794 (±0.048)	0.552 (±0.051)
	KV	Name & Addr & Misc	0.596 (±0.037)	**0.842** (±0.007)	0.872 (±0.007)	0.708 (±0.021)
	Template	Random	0.456 (±0.032)	0.774 (±0.011)	0.842 (±0.006)	0.595 (±0.023)
	Template	Name & Addr & Misc	**0.607** (±0.015)	**0.842** (±0.004)	**0.877** (±0.006)	**0.714** (±0.008)
E5	KV	Random	0.465 (±0.031)	0.777 (±0.050)	0.844 (±0.037)	0.602 (±0.033)
	KV	Name & Addr & Misc	0.570 (±0.024)	0.827 (±0.004)	0.875 (±0.004)	0.683 (±0.014)
	Template	Random	0.478 (±0.022)	0.788 (±0.026)	0.850 (±0.011)	0.613 (±0.021)
	Template	Name & Addr & Misc	**0.573** (±0.050)	**0.828** (±0.052)	**0.877** (±0.030)	**0.685** (±0.045)

Table 9: Performance of the string similarity baselines, BERT-based model, and E5-based model with different settings for two representative aspects, i.e., entry text representations (EntRep) and hard negative mining criteria (NegCrite), on the test set. For both BERT- and E5-based models, the **best scores** and second-best scores across the four settings are highlighted, respectively. Note that the results except for the BERT-based model are identical to those in Table 2.

the E5-based model, two separate BERT encoders (initialized from the same pretrained model checkpoint) were used to model mention vectors and entry vectors. Furthermore, the `[CLS]` token vectors $\boldsymbol{h}_1^{(t)}$ and $\boldsymbol{h}_1^{(d)}$ were used as the mention vector $\boldsymbol{h}_m$ and the entry vector $\boldsymbol{h}_e$, respectively, instead of applying average pooling of token vectors over a specific span or the entire text.

Table 9 shows the experimental results on the test set, where the results for Leven, BM25, and E5 are identical to those in Table 2. Almost consistent with the discussion of the E5-based model in §4.1, we observed the following: (1) the BERT-based model outperformed the string similarity baselines, (2) the selection criteria for hard negatives had a significant impact on performance, and (3) the choice between the two entry text representations had a limited impact on performance. In the BERT-based model, however, the template representation outperformed the KV representation when the random criterion was applied.

E Travelogue Example

Figure 6 shows fragment text in a travelogue in the development set. Mentions are underlined and contexts related to cycling is dashed underlined, which are useful to identify the gold entries. As explained in §4.5, for mention 来島海峡大橋 (Kurushima Kaikyo Bridges), all variants of the E5-based model in Table 6 failed to predict the correct entry "来島海峡大橋;しまなみ海道サイクリングロード" (Kurusima Kaikyo Bridges; Shimanami Kaido Cycring Road). For mention 伯方・大島大橋 (Hakata-Oshima Bridge), a model variant predicted the correct entry because there

展望台から見下ろす来島海峡大橋、とその歩行者・自転車レーン。
橋に至るスロープも相当な登りだったけど、橋も微妙に登っているのか、皆漕ぎがゆ〜っくり。
今回は多々羅大橋〜伯方・大島大橋間のみなので、次回ぜひ通ってみたいです。
遠くで見ると細かい吊が、近くだとかなり幅広。
すぐ隣では高速で車がビュンビュン、眼下には海が広がり、わくわくしてしまいます。
人もまばらで、バーイシコー　バーイシコー♪(クイーン)と歌いながら上機嫌。

Figure 6: Example of travelogue in the dev set. Mentions are underlined and contexts related to cycling is dashed underlined, for example, "自転車レーン" (the bicycle lane) and "皆漕ぎがゆ〜っくり" (everyone is pedaling (their bicycles) slowly).

are no candidates similar to the gold entry "伯方・大島大橋;しまなみ海道サイクリングロード" (Hakata-Oshima Bridge; Shimanami Kaido Cycring Road).