Low-resource Taxonomy Enrichment with Pretrained Language Models

Kunihiro Takeoka, Kosuke Akimoto, Masafumi Oyamada
Data Science Research Laboratories, NEC Corporation
{k_takeoka, kosuke_a, oyamada}@nec.com

Abstract

Taxonomies are symbolic representations of hierarchical relationships between terms or entities. While taxonomies are useful in broad applications, manually updating or maintaining them is labor-intensive and difficult to scale in practice. Conventional supervised methods for this enrichment task fail to find optimal parents of new terms in low-resource settings where only small taxonomies are available because of overfitting to hierarchical relationships in the taxonomies.

To tackle the problem of *low-resource taxonomy enrichment*, we propose Musubu, an efficient framework for taxonomy enrichment in low-resource settings with pretrained language models (LMs) as knowledge bases to compensate for the shortage of information. Musubu leverages an LM-based classifier to determine whether or not inputted term pairs have hierarchical relationships. Musubu also utilizes Hearst patterns to generate queries to leverage implicit knowledge from the LM efficiently for more accurate prediction. We empirically demonstrate the effectiveness of our method in extensive experiments on taxonomies from both a SemEval task and real-world retailer datasets.

1 Introduction

Taxonomies, which represent the hierarchical relationships between terms, have been widely utilized in tasks related to information retrieval, recommendation, and classification (Agrawal et al., 2009; Huang et al., 2019; Babbar et al., 2016). Because the target domain of a specific taxonomy changes over time, taxonomies must be kept up to date so that newly introduced categories and hierarchical relationships can be properly integrated. However, manually constructing and maintaining taxonomies is a costly task due to their labor-intensive and domain-specific nature (Gao et al., 2018; Shen et al., 2018).

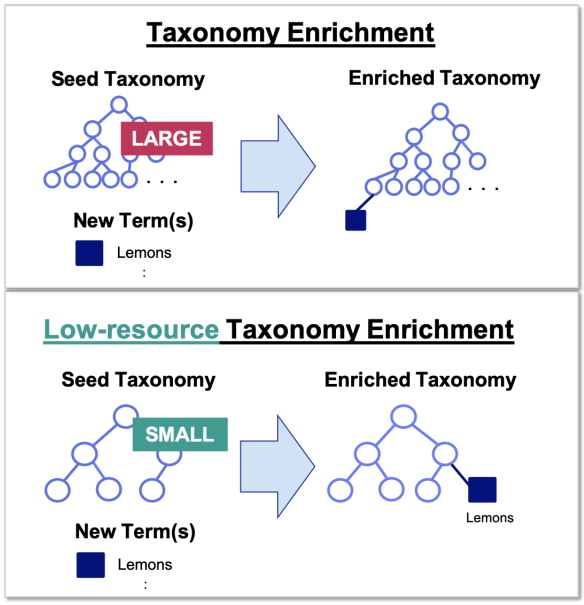

Figure 1: Enrichment of taxonomy with sufficient resources (top) and low resources (bottom). Conventional supervised methods for taxonomy enrichment fail to enrich small seed taxonomies because the number of hierarchical relations of term pairs used as training samples is limited.

The goal of a *taxonomy enrichment task* is to automate this costly maintenance (Jurgens and Pilehvar, 2016; Shen et al., 2018), as shown in Fig. 1. Taxonomy enrichment methods enrich a taxonomy that may be incomplete, i.e., a *seed taxonomy*, by predicting new hierarchical relationships between terms in the seed taxonomy and new terms. We focus on *low-resource taxonomy enrichment tasks*, in which there are few terms in the seed taxonomy, specifically less than 10,000. Previous studies on hypernymy detection used a similar definition for low-resource settings (Yu et al., 2020). Although conventional methods for taxonomy enrichment use large taxonomies (e.g., WordNet taxonomy (90,000 terms) (Miller, 1995), Microsoft Academic Graph (355,000 terms) (Sinha et al., 2015), most of the ones used in realistic situations (as seen in

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2747–2758
November 7–11, 2021.

§4) have around a thousand terms, so they are categorized as low-resource settings.

Existing supervised methods (Baroni et al., 2012; Shen et al., 2020; Manzoor et al., 2020) train their parameters with term pairs in a seed taxonomy to find the optimal parent of a new term. These methods do not work in low-resource settings (as we verify in §4) because they require many term pairs. Although Mao et al. (2020) partially tackled the problem of a small seed taxonomy, their proposed method requires additional information such as user search logs to improve the performance, which is not always available.

To handle low-resource taxonomy enrichment tasks, we propose **Musubu**, an efficient framework for such situations, with pretrained language models (LMs) to compensate for the shortage of information on hierarchical relationships in a seed taxonomy. Musubu leverages a pretrained LM-based classifier to determine whether term pairs have hierarchical relations for enriching taxonomies. The classifier can find the relationships accurately even if the training samples are limited because LMs contain real-world term relationships in their parameters (Petroni et al., 2019). Musubu also utilizes Hearst patterns (Hearst, 1992) for generating queries from term pairs to extract implicit knowledge related to them embedded in LMs efficiently.

We empirically demonstrate the effectiveness of our approach in the taxonomy enrichment task through extensive experiments on both the SemEval-2015 Task 17 (Bordea et al., 2015) taxonomies and real-world commerce taxonomies in Amazon and Walmart.

Our contributions are summarized as follows:

- We propose an approach for taxonomy enrichment that utilizes a pretrained LM as an implicit knowledge base and infers new hierarchical relationships.
- We leverage Hearst patterns to generate queries from term pairs to extract LM's knowledge effectively.
- We empirically demonstrate the effectiveness of our method for low-resource taxonomy enrichment through extensive experiments on real-world taxonomies.

2 Problem Statement

In this section, we describe the notations and the problem definition of this paper.

2.1 Notations

We denote a taxonomy $T = (V, E)$ as a tree-structured hierarchy with a term set V and a directed edge set E. A term $v \in V$ can consist of either a single word (e.g., "tea") or multiple words (e.g., "soy milk"). The edge set $E = \{(v, v_{par}); v \in V\} \subset V \times V$, where $v_{par} \in V$ is a parent of v, represents a set of hierarchical relationships between terms (e.g., "milk tea" $\rightarrow$ "tea").

2.2 Problem Definition

Given a seed taxonomy $T = (V, E)$ and a set of new terms V', *taxonomy enrichment* estimates an extended taxonomy $\tilde{T} = (\tilde{V}, \tilde{E})$ with $\tilde{V} = V \cup V'$ and $\tilde{E} = E \cup E'$ (Shen et al., 2018; Mao et al., 2020). Here, E' contains new edges $(v', v), v' \in V'$, and $v \in V$, where v is a parent term of v'. In this paper, we focus on a situation where the taxonomy $|V|$ is small such as $|V| < 10,000$ as mentioned in §1, hereafter referred to as a *low-resource taxonomy enrichment* problem.

3 Method

The goal of low-resource taxonomy enrichment is to find the optimal parent $v^\star \in V$ of each new term $v' \in V'$ in low-resource settings. We introduce a formulation of the taxonomy enrichment problem based on our probabilistic taxonomy model, then elaborate on our classification approach based on pretrained LMs and Hearst patterns.

3.1 Probabilistic Model on Taxonomy

To formulate the taxonomy enrichment problem, we model the entire taxonomy using the graphical model formulation by Bansal et al. (2014)[1]. Specifically, we define the likelihood of a taxonomy using a graphical model with a factor for each edge in the taxonomy:

$$p(T|\Theta) \propto \prod_{(v_1,v_2)\in E} \phi_E(v_1, v_2|\Theta), \qquad (1)$$

where Θ is a set of model parameters, v_2 denotes the parent of a term v_1 in taxonomy T, and ϕ_E is an associated scoring function of the term pair. By using the probabilistic model, we formulate the taxonomy enrichment problem to find the optimal

[1]While their study considered multiple types of factors, we only use edge factors.

enriched taxonomy $\tilde{T}^\star$ as the following optimization problem:

$$\begin{aligned}\tilde{T}^\star &= \arg\max_{\tilde{T}} p(\tilde{T}|\Theta) \\ &= \arg\max_{\tilde{T}} \prod_{(v_1,v_2)\in\tilde{E}} \phi_E(v_1, v_2|\Theta) \quad (2) \\ &= \arg\max_{\tilde{T}} \prod_{(v',v)\in E'} \phi_E(v', v|\Theta). \quad (3)\end{aligned}$$

Following the conventions of taxonomy enrichment, the low-resource taxonomy enrichment problem (§2.2) assumes that a seed taxonomy T never changes and a new term v' is always attached under the terms in a seed taxonomy T (in other words, it will not be attached under previously added terms). The assumptions allow us to ignore the scores of edges E in a seed taxonomy and the factors between new terms (Eq. (2) $\rightarrow$ Eq. (3)). Thus, we can cast the problem of finding the optimal parent term $v^\star(\in V)$ of a new term v' as the following optimization problem for each term $v' \in V'$:

$$v^\star = \arg\max_{v\in V} \phi_E(v', v|\Theta). \quad (4)$$

The optimization problem Eq. (4) can be regarded as a multiclass classification problem, the classes of which are V. However, each class, $v \in V$, has only a few or no positive training examples i.e., children of v. Thus, instead of directly solving this kind of problem, we treat it as a binary classification problem that classifies whether a term (v', v) has a hierarchical relationship.

3.2 Classifier Leveraging Pretrained Language Model and Hearst Patterns

To mitigate the shortage of information in low-resource taxonomy enrichment, our Musubu framework, a novel approach to taxonomy enrichment, leverages pretrained language models (LMs) as alternative information resources. Musubu consists of two main modules: an LM-based classifier p_{LM} and a query generator q, as shown in Fig. 2. The query generator generates a query text $q(v', v)$ from a given term pair (v', v). Then, given the query text $q(v', v)$, the LM-based classifier f_{LM} is used to classify whether the term pair has a hierarchical relationship. The LM-based classifier uses implicit knowledge embedded in the LM and the query generator utilizes Hearst patterns (Hearst, 1992) to generate queries to use the LM's knowledge efficiently for taxonomy enrichment. In addition, we fine-tune the LM-based classifier with hierarchical relationships in a seed taxonomy to adapt it to the taxonomy.

Language Models as Knowledge Bases. Musubu leverages a pretrained LM as an external knowledge base for enriching taxonomies. According to Petroni et al. (2019), large LMs (e.g., BERT (Devlin et al., 2019)) acquire term meanings and relationships as their weights by training on many documents. In low-resource settings, LMs can potentially improve the performance of taxonomy enrichment because relational knowledge learned in LMs can augment a limited number of available hierarchical relationships in seed taxonomies for training.

Classifier with Language Models. To leverage LMs in the taxonomy enrichment task, we take an LM-based text classification approach that uses an LM and a fully connected (FC) layer to classify texts[2]. LM-based text classifiers are more often used in few-shot classification than classifiers with word2vec/fasttext (Gupta et al., 2020). We input a query to the LM-based classifier, and then the classifier detects whether or not the term pair corresponding to the query has a hierarchical relationship. The classifier is fine-tuned with a seed taxonomy to adapt to the hierarchical relationships (see §3.3 for details).

Query Generation. To use pretrained LMs efficiently for taxonomy enrichment, Musubu generate queries from term pairs by using Hearst patterns (Hearst, 1992) and then input them into the LM-based classifier. These patterns are well-known lexical patterns used to represent hypernym-hyponym relationships (e.g., "Y such as X") as shown in Table 2. Normally, Hearst patterns are used for hypernymy detection from text corpora. We use Hearst patterns in a way that is different from the original approach to generate a query from a term pair for LMs, as shown in Fig. 2 (a). For instance, when we have a term pair "oranges" and "fruits" and choose a pattern "Y such as X," we generate the query "fruits such as oranges." We then input the generated query to the LM-based classifier to classify whether or not the corresponding term pair has a hierarchical relationship. While the conventional pattern-based approaches find hi-

[2]Although the masked language model scoring (Salazar et al., 2020) can be used for taxonomy enrichment by scoring queries of term pairs, this approach is ineffective because hierarchical relations in seed taxonomies are ignored as shown in the Musubu-noFT row in Table 3.

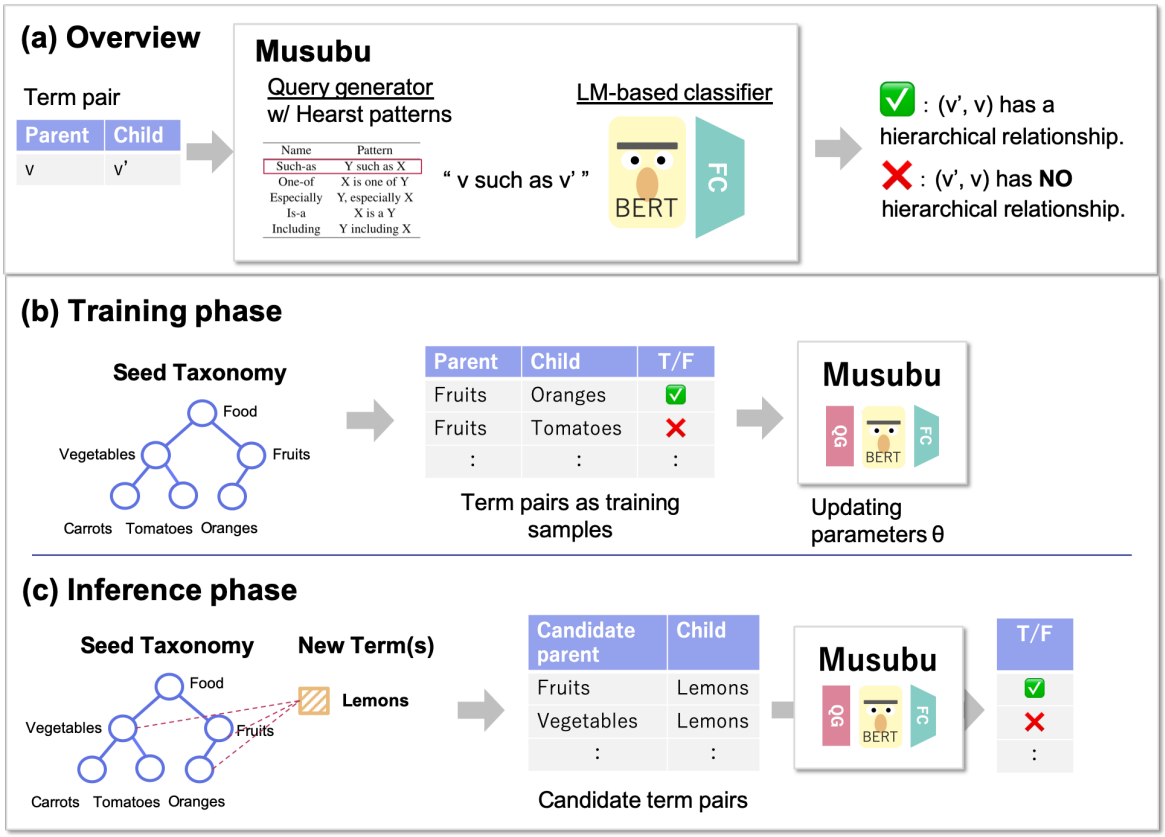

Figure 2: (a) Musubu architecture, (b) training, and (c) inference. Musubu transforms each term pair (v', v) into a Hearst pattern-based query "v such as v'," encodes the query, and then determines whether or not the term pair has a hierarchical relationship.

erarchical relationships from the text corpus by matching the patterns, our approach utilizes the LM's implicit knowledge using generated queries with patterns.

How does the query generator take advantage of Hearst patterns in generating queries in Musubu? Output sentences from the query generator are fed into LMs trained on a natural language corpus, so sentences from the generator should be naturally written. We found that Hearst patterns were used in the past to extract a hierarchical relationship of two terms from a natural language corpus, and the patterns were then used to generate naturally written sentences which imply hierarchies. We verified that the naturally written queries outperform awkward queries such as space-delimited terms as shown in §4.

3.3 Self-supervised Training and Inference

We create training data to enumerate all term pairs in a seed taxonomy $V \times V$ and fine-tune the LM-based classifier. As shown in Fig. 2(b), we add positive labels when term pairs have hierarchical relationships in the seed taxonomy; otherwise, we add negative labels. We fine-tune the model parameters Θ including the LM's parameters with the training data. We use a binary cross-entropy loss as the objective function and minimize it to find the optimal parameters $\Theta^\star = \min_\Theta \mathcal{L}(\Theta)$. We minimize the objective function for fine-tuning our model:

$$\mathcal{L}(\Theta) = -\sum_{(v_1^+, v_2^+) \in E} \log(f_{\mathrm{LM}}(q(v_1^+, v_2^+), \Theta)) - \sum_{(v_1^-, v_2^-) \in V \times V \setminus E} \log(1 - f_{\mathrm{LM}}(q(v_1^-, v_2^-), \Theta)). \quad (5)$$

To infer the optimal parent of each new term $v' \in V'$, we take every term in a seed taxonomy $v \in V$, and input a term pair (v', v) to the LM-based classifier to obtain a score $f_{\mathrm{LM}}(q(v', v), \Theta^\star)$ as shown in Fig 2 (c). Then, we output the term $v^\star$, which is the highest score, $v^\star = \arg\max_{v \in V} f_{\mathrm{LM}}(q(v', v), \Theta^\star)$.

4 Experiments

In this section, we describe how we studied the performance of Musubu on seven real-world tax-

Table 1: Statistics of taxonomies used in experiments. $|V|$ denotes the number of terms in the seed taxonomy and V_{nl} denotes its non-leaf terms. V_{tr}, V_{dev}, and V' denote the leaf terms used for training, development, and testing, respectively.

Dataset	Taxonomy	$\|V\|$	$\|V_{nl}\|$	$\|V_{tr}\|$	$\|V_{dev}\|$	$\|V'\|$
SemEval-2015 Task 17	Chemical	1146	294	682	170	273
	Equipment	406	122	226	57	71
	Food	1254	275	783	196	245
	Science	368	107	209	52	66
Commerce	Amazon Food	860	159	534	167	134
	Amazon Kitchen	1019	229	632	158	198
	Walmart	1085	376	567	142	178

Table 2: List of Hearst patterns used in experiments. Y denotes a parent term of a term X.

Name	Pattern
Such-as	Y such as X
One-of	X is one of Y
Especially	Y, especially X
Is-a	X is a Y
Including	Y including X

onomies.

4.1 Experimental Setup

Datasets. We used four SemEval taxonomies and three commerce taxonomies including various domains as shown in Table 1. SemEval-2015 Task 17 (Bordea et al., 2015) is a taxonomy extraction task, which contains four domains, chemical, equipment, food, and science. Each taxonomy is relatively small compared with the taxonomy used in SemEval-2016 Task 14 (Jurgens and Pilehvar, 2016), which has over 90,000 nodes[3]. We also used real commerce taxonomies from Amazon review data (Ni et al., 2019), Grocery & Gourmet Foods (Amazon Food), Home & Kitchen (Amazon Kitchen), and the Walmart taxonomy[4]. The commerce taxonomies contain more named entities than the SemEval taxonomies. For instance, the term "IPA & Pale Ale" appears in the Amazon Food taxonomy, but does not appear in the SemEval food taxonomy.

Considering a realistic situation, we held out 20% of leaf nodes as the new terms V'. Detailed statistics of the taxonomies are listed in Table 1.

Evaluation Metrics. We used two evaluation metrics, **hierarchical-F1 (H-F1)** and **edge-F1 (E-F1)** (Mao et al., 2020). Hierarchical-F1 is a commonly used measure for hierarchical classification tasks that compares the true path from the true parent to the root with the predicted path (Kiritchenko et al., 2005). Edge-F1 is the top-1 hit ratio of the predicted hypernyms, and is a more strict metric than H-F1.

Compared Methods. We compared Musubu with five baseline approaches:

1. **Random**: A simple baseline which randomly selects a parent term from V.

2. **Microsoft Concept Graph (MCG)**: Microsoft Concept Graph[5] (Wu et al., 2012; Wang et al., 2015) is an existing large-scale hypernymy knowledge base which is extracted from billions of web pages and consists of triplets (parent term, child term, frequency) which represent hierarchical relationships. We attach a given new term $v' \in V'$ to a term $v \in V$ if a hierarchical relation between v' and v exists in the knowledge base.[6]

3. **TaxoExpan**[7](Shen et al., 2020): A self-supervised method which leverages a position-enhanced graph neural network encoding the local structure in a seed taxonomy, and it uses a noise-robust training objective to learn the model.

4. **MSejrKu** (Schlichtkrull and Martínez Alonso, 2016): The winning method in SemEval-2016 Task 14 (Semantic Taxonomy Enrichment) (Jurgens and Pilehvar, 2016) which extracts semantic and lexical features and classifies them with support vector machines.

5. **Octet** (Mao et al., 2020): A self-supervised method which extracts semantic, lexical, and graph-based features and classifies hierarchical relationships between term pairs using a two-layer feed-forward neural network with dropout layers. In the original method, the graph-based features are extracted from e-commerce user queries. In our experiments,

[3]The task aims to enrich the WordNet taxonomy using new terms and their word sense. The number of nodes in the seed taxonomy is out of the scope of this paper due to its size.

[4]https://www.kaggle.com/promptcloud/walmart-product-data-2019

[5]https://concept.research.microsoft.com.

[6]If there is more than one such relation, we select the one with the highest frequency. If no such relation exists, we attach v' to a random term from V. We match terms by simple string-matching after lower-casing.

[7]https://github.com/mickeystroller/TaxoExpan

Table 3: Overall results for taxonomy enrichment on the SemEval and commerce taxonomies. Musubu-noFT denotes Musubu without fine-tuning on the seed taxonomy. The best scores in the columns are in bold.

	SemEval-2015 Taxonomies								Commerce Taxonomies					
	Chemical		Equipment		Food		Science		Amazon Food		Amazon Kitchen		Walmart	
Method	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1
Random	0.00	0.62	0.01	0.53	0.00	0.44	0.01	0.49	0.01	0.34	0.01	0.35	0.00	0.34
MCG	0.21	0.68	0.06	0.50	0.18	0.55	0.08	0.40	0.06	0.37	0.00	0.33	0.01	0.33
TaxoExpan	0.00	0.52	0.03	0.46	0.01	0.39	0.01	0.42	0.00	0.36	0.01	0.35	0.00	0.30
MSejrKu	0.26	0.77	0.13	0.63	0.20	0.62	0.24	0.66	0.18	0.55	0.21	0.53	0.15	0.51
Octet	0.31	0.76	0.33	0.71	0.24	0.60	0.36	0.70	0.36	0.64	0.28	0.61	0.24	0.60
Musubu-noFT	0.00	0.60	0.00	0.60	0.00	0.45	0.00	0.44	0.00	0.52	0.00	0.55	0.00	0.56
Musubu	**0.37**	**0.79**	**0.45**	**0.73**	**0.37**	**0.68**	**0.44**	**0.77**	**0.40**	**0.66**	**0.44**	**0.71**	**0.53**	**0.80**

we did not use the graph-based features because we had no user queries related to the seed taxonomies.

6. **Musubu**: Our method which leverages BERT (Devlin et al., 2019) as a pretrained LM, and fine-tunes a LM-based classifier with queries generated from the Such-as pattern. To analyze the effects of fine-tuning, we test a masked LM scoring method (Salazar et al., 2020) on the same queries without fine-tuning (Musubu-noFT).

Implementation Details. In our experiments, we used the public fasttext model (Bojanowski et al., 2017) trained on the Common Crawl corpus[8] to extract semantic features from term pairs in TaxoExpan, MSejrKu, and Octet. We also used the lexical features proposed by Bansal et al. (2014). During training, we randomly sampled nine negative term pairs for each positive pair. We implemented MSejrKu, Octet, and Musubu using PyTorch (Paszke et al., 2019), Transformers (Wolf et al., 2020), and Scikit-learn (Pedregosa et al., 2011). For both Musubu and the baseline methods, we tuned the hyperparameters including the optimizer, initial learning rate, dropout rate, and batch size, on the basis of the average performance of 20 random trials on the development set of the Amazon Food taxonomy. We used the Adam optimizer with a tuned learning rate of 8.8×10^{-4}, and a Tesla V100 GPU for training and inference. We used `bert-base-uncased` as a pretrained model in Musubu and limited the maximum length of tokens to 64, and longer queries were truncated. In addition, unless otherwise noted, we used the Such-as pattern to generate queries. Table 2 shows the patterns used in the experiments.

[8] https://fasttext.cc/docs/en/english-vectors.html

4.2 Evaluation Results

SemEval and Commerce Taxonomies. We evaluated the baselines and our method (Musubu) on the SemEval-2015 Task 17 and real commerce taxonomies shown in Table 3. The method of selecting parents randomly (Random) yielded edge-F1 scores of almost zero, which indicates the task's difficulty. TaxoExpan was not suitable for any taxonomy because the method assumes that the number of terms in a seed taxonomy is sufficiently large for extracting graph-based features with graph neural networks. Overall, Musubu performed most effectively across both metrics in various domains. The results show that our BERT-based approach outperformed Octet, which uses fasttext to extract semantic features. MCG was not effective on the SemEval and commerce taxonomies because the hierarchical relationships in the taxonomies were not always the same relations stored in the general is-a database. Finally, Musubu fine-tuned on the seed taxonomy was more effective than without that fine-tuning (Musubu-noFT). Although LMs generally have term relationships, the LM-based classifier needs to be fine-tuned to adapting to term relationships in the seed taxonomy.

Low-resource Settings. We evaluated the performance in more low-resource settings than that of the above experiments, as shown in Fig. 3. Because TaxoExpan was ineffective in the above settings, we used Octet and MSejrKu as baselines for comparison with Musubu in low-resource settings. The results show Musubu was more effective than the baselines, although the overall performance was declined when there was an insufficient number of training terms. Compared Musubu with Octet, the pretrained LM used in Musubu helped to estimate hierarchical relations accurately. The results of the experiment supported our hypothesis that pretrained LMs are useful for low-resource taxonomy enrichment.

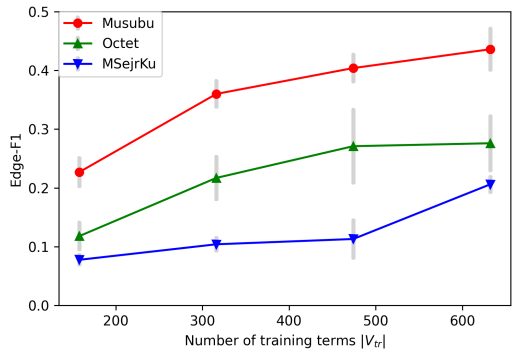

Figure 3: Performance comparison on edge-F1 for low-resource taxonomy enrichment on Amazon Kitchen taxonomy. Musubu outperformed the baselines, especially when the seed taxonomy was small.

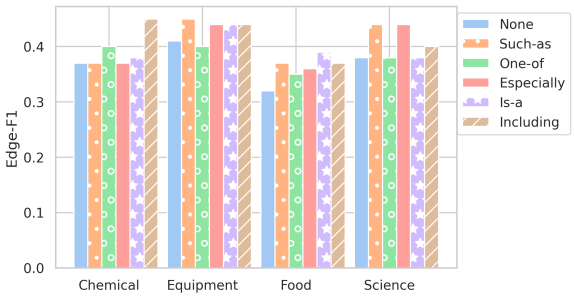

Figure 4: Pattern analysis of Musubu on SemEval taxonomies.

Pattern Analysis. We compared the performance of Musubu with several different Hearst patterns (Table 2) for generating queries. As a baseline pattern, we tested the None pattern, in which two terms are concatenated with a single space (e.g., "fruits oranges" for a term pair ("oranges", "fruits")). As shown in Fig. 4, the Such-as pattern obtained the highest score among the experimented patterns on two taxonomies. Using a Hearst pattern contributes to the performance of taxonomy enrichment, as shown in the compared results between the None pattern and the others. However, the scores on the chemical taxonomy show that the Including pattern is more effective than the Such-as pattern. The results indicate that we should choose the optimal pattern for each domain, although the Such-as pattern can be considered the first choice for a general case.

Case Studies. We analyzed the predictions to understand the model behavior of Musubu by comparing it and Octet on the Walmart taxonomy. As shown in Table 4, both methods predicted "food" correctly as the parent of "gluten-free foods," and Musubu also captured the lexical features. The "men's socks" row shows that both methods captured the semantic features for taxonomy enrichment. The "hair care" row exemplifies the difference between Musubu and Octet. Musubu predicted "hair care" as the parent of "dry shampoo" by extracting hierarchical relationships from the LM, while Octet predicted "skin care" because of the lack of training samples on the term "hair care."

5 Related Work

5.1 Hypernymy Detection

Hypernymy detection is a core natural language processing (NLP) task for estimating which hypernym term a query term corresponds to, which is a subtask of taxonomy construction or extraction (Wang et al., 2017). Both unsupervised and supervised methods have been proposed for this task. Unsupervised methods are categorized into pattern-based or distributional. Pattern-based approaches predict that the term pair (x, y) has an is-a relation if x and y satisfy syntactic patterns in given documents, for instance, Hearst patterns (Hearst, 1992) are as shown in Table 2. Distributional approaches use the distributional representations of term pairs to measure the strength of their Is-A relationships (Geffet and Dagan, 2005). These approaches cannot adapt to the domain-specific hierarchical relationships in the seed taxonomy. In contrast, the supervised methods follow a classification paradigm. In most of the supervised methods, each term pair transforms feature vectors constructed from word embeddings and identifies whether or not they have hypernymy relationships (Baroni et al., 2012; Roller et al., 2014; Shwartz et al., 2016). The methods fail to work in low-resource taxonomy enrichment because there are not enough training term pairs for learning models.

5.2 Taxonomy Construction and Enrichment

Taxonomy construction (taxonomy extraction) is an automatic task in which we obtain terms from a given corpus, construct a graph containing edges that represent hierarchical relationships, and reform the graph into a tree or directed acyclic graph (Wu et al., 2012; Bansal et al., 2014; Bordea et al., 2016). The second step, constructing a graph, is for finding a term pair's relationships from the given information, similarly to hypernymy detection.

Unlike taxonomy construction, *taxonomy enrichment* (Jurgens and Pilehvar, 2016; Shen et al., 2018)

Table 4: Case studies of taxonomy enrichment in Walmart taxonomy. Top-3 predicted terms of methods for an input term. The predicted terms in bold are the true parents.

New term	Method	Top 3 predictions
gluten-free foods	Octet	**food**; fresh food; baby food
	Musubu	**food**; snacks, cookies & chips; medical & dental
men's socks	Octet	**men's clothing**; women's socks, hosiery & tights; men's shoes
	Musubu	men's shoes; **men's clothing**; men's bags & accessories
dry shampoo	Octet	skin care; camping gear; shop by brand
	Musubu	**hair care**; medical & dental; bath safety

augments a seed taxonomy with new terms by finding the optimal hypernym term corresponding to a new term from the taxonomy. The main difference between taxonomy enrichment and hypernymy detection is whether or not the hierarchy of taxonomy can be used. While hypernymy detection is a task for extracting the aforementioned general hypernym-hyponym relations regardless of the domain, taxonomy enrichment strongly depends on the domain in that it is extended with reference to the hierarchical relationships of a given taxonomy. For instance, the SemEval-2016 Task 14 (Jurgens and Pilehvar, 2016) presents a semantic taxonomy enrichment task that extends the WordNet taxonomy with new terms and their definitions. The winning method in the task (Schlichtkrull and Martínez Alonso, 2016) used both the hand-crafted semantic and lexical features of term pairs to find hierarchical relationships.

More complicated approaches have been proposed for improving the performance of finding hierarchical relationships. Shen et al. (2018) devised an end-to-end pipeline for extracting new terms from documents and taxonomy enrichment, which integrates AutoPhrase (Shang et al., 2018) for term extraction and a distributional approach for finding siblings and hypernyms of terms. Other methods utilize self-supervision to enrich a seed taxonomy, but they require either large seed taxonomies and/or additional information about hierarchical relationships. TaxoExpan (Shen et al., 2020) is the first attempt to use a graph neural network to accurately predict hypernyms with self-supervision. Octet (Mao et al., 2020) utilizes the feature extractors proposed in (Schlichtkrull and Martínez Alonso, 2016) and user queries as additional information for taxonomy enrichment by means of graph neural networks and self-supervision. In contrast to these methods, including the winning method in the SemEval-2016 task, our method focuses on low-resource taxonomy enrichment, in which small taxonomies and no text corpus are available. The previous methods do not work on low-resource settings because conventional self-supervised approaches utilize the graph-based features of a large seed taxonomy.

5.3 Language Models as Knowledge

Pretrained LMs on large text corpora improve the performance on downstream NLP tasks such as text classification and question answering (Gupta et al., 2020; Su et al., 2019). However, an important question was raised about pretrained LMs: do the pretrained LMs have information about entities and the relationships between them? Petroni et al. (2019) showed that BERT contains relational knowledge as well as knowledge bases. The results indicate the weights of pretrained LMs containing relationships between terms, which also include hierarchical relationships. The fact that pretrained LMs contain the term relationships is used in entity set expansion (Zhang et al., 2020). The method utilizes the masked LMs to estimate similar entities using formatted queries. Although the method focuses on the entity set expansion task, our method tackles low-resource taxonomy enrichment tasks. The aforementioned papers suggest that our approach for taxonomy enrichment is reasonable.

6 Conclusion

We proposed an efficient self-supervised approach, Musubu, for low-resource taxonomy enrichment tasks. Musubu utilizes a novel classifier based on pretrained LMs and Hearst patterns for generating queries. Extensive experiments on taxonomy enrichment showed the effectiveness of Musubu over the conventional approaches on the SemEval and commerce taxonomies, especially in low-resource settings.

References

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, page 5–14. Association for Computing Machinery.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Rohit Babbar, Ioannis Partalas, Éric Gaussier, Massih-Reza Amini, and Cécile Amblard. 2016. Learning taxonomy adaptation in large-scale classification. *J. Mach. Learn. Res.*, 17:98:1–98:37.

Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1041–1051, Baltimore, Maryland. Association for Computational Linguistics.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the 9th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

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.

Yuqing Gao, Jisheng Liang, Benjamin Han, Mohamed Yakout, and Ahmed Mohamed. 2018. Building a large-scale, accurate and fresh knowledge graph. *the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Tutorial*.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Aakriti Gupta, Kapil Thadani, and Neil O'Hare. 2020. Effective few-shot classification with transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1061–1066, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 573–581. Association for Computing Machinery.

David Jurgens and Mohammad Taher Pilehvar. 2016. SemEval-2016 task 14: Semantic taxonomy enrichment. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1092–1102, San Diego, California. Association for Computational Linguistics.

Svetlana Kiritchenko, Stan Matwin, and A. Fazel Famili. 2005. Functional annotation of genes using hierarchical text categorization. In *Proceedings of the ACL workshop on linking biological literature, ontologies and databases: mining biological semantics*.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and J. Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*.

Yuning Mao, Tong Zhao, A. Kan, Chenwei Zhang, X. Dong, Christos Faloutsos, and J. Han. 2020. Octet: Online catalog taxonomy enrichment with self-supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

G. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38:39–41.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in*

Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Michael Schlichtkrull and Héctor Martínez Alonso. 2016. MSejrKu at SemEval-2016 task 14: Taxonomy enrichment by evidence ranking. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1337–1341, San Diego, California. Association for Computational Linguistics.

Jingbo Shang, Jialu Liu, Meng Jiang, X. Ren, Clare R. Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30:1825–1837.

J. Shen, Zhihong Shen, Chenyan Xiong, Chunxin Wang, Kuansan Wang, and Jiawei Han. 2020. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*.

J. Shen, Zeqiu Wu, Dongming Lei, C. Zhang, Xiang Ren, M. Vanni, Brian M. Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. WWW '15 Companion, page 243–246, New York, NY, USA. Association for Computing Machinery.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing question answering system with pre-trained language model fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 203–211, Hong Kong, China. Association for Computational Linguistics.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1203, Copenhagen, Denmark. Association for Computational Linguistics.

Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao. 2015. An inference approach to basic level of categorization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 653–662, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, page 481–492, New York, NY, USA. Association for Computing Machinery.

Changlong Yu, Jialong Han, Haisong Zhang, and Wilfred Ng. 2020. Hypernymy detection for low-resource languages via meta learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3656, Online. Association for Computational Linguistics.

Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via language model probing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8151–8160, Online. Association for Computational Linguistics.

A Detailed Experimental Settings

A.1 Data Preparation

To filter out noisy terms in the original commerce taxonomies, we removed infrequent terms that appeared fewer than five (Amazon Food) or 20 times (Amazon Kitchen and Walmart) in item categories of items and were shorter than ten words. We did not remove any terms from the SemEval taxonomies. We split the leaf terms into the training set and V' with a ratio of 80% / 20% and then split the training set into V_{tr} and V_{dev} at the same ratio. We set the non-leaf terms in the taxonomy as V_{nl}. The numbers of terms are listed in Table 1. The evaluation scores were the averages over the scores on three trials.

A.2 Models

Musubu. We used `BertForSequentialClassification`[9] as our LM-based classifier and initialized its parameters with `bert-base-uncased`. We fine-tuned the last layer in the BERT model and the classification layer (fully connected layer) and froze the parameters in the other layers to avoid overfitting the model. The number of parameters in Musubu is about 109M, and the number of trainable parameters is about 7M.

Musubu-noFT (Musubu without Fine-tuning). We also used `bert-base-uncased` as a pre-trained LM to calculate the likelihood of queries without fine-tuning on a seed taxonomy. Musubu-noFT generates Hearst pattern-based queries from term pairs, calculates the likelihoods of the queries, and then finds the optimal parent of each new term while maximizing the likelihood. To calculate the likelihood, we used the masked language model scoring (Salazar et al., 2020) with the public implementation provided by the authors[10]. There are no trainable parameters in the approach.

TaxoExpan (Shen et al., 2020). We used the authors' public implementation and set the hyperparameters by default in the original source code. To extract semantic features, we used fasttext as same as Octet and MSejrKu. The number of trainable parameters in TaxoExpan is 1.9M.

Octet (Mao et al., 2020). We extracted semantic and lexical features, input them into a two-layer feed-forward network with dropout layers, and used the output as the probability of terms. While the original method used graph-based features generated from user queries and the taxonomy in addition to the above features, we did not use them because we did not have any user queries. We constructed the lexical features following (Bansal et al., 2014) and the semantic features with fasttext (Bojanowski et al., 2017) trained by the Common Crawl dataset. To tune the model parameters, we used the same optimizer and the number of negative samples as that of our proposed method. The number of parameters in Octet is 503K.

MSejrKu (Schlichtkrull and Martínez Alonso, 2016). Although this method was the winner of the SemEval-2016 Task 14, the implementation is not public. According to (Mao et al., 2020), the features used in Octet are similar to those used in MSejrKu. We extracted the semantic and lexical features in Octet, and then input a support vector machine.

A.3 Hyperparameter tuning

We tuned the hyperparameters of our and that of the baselines for taxonomy enrichment using their H-F1 on the Amazon Food development dataset[11] by Optuna (Akiba et al., 2019). See Table 5 for the detailed ranges of tuned hyperparameters and the optimal values for our method and the baselines. We conducted 20 trials for each method to tune the hyperparameters.

[9] https://huggingface.co/transformers/model_doc/bert.html#bertforsequenceclassification

[10] https://github.com/awslabs/mlm-scoring

[11] We used the optimal hyperparameter settings for the other dataset.

Table 5: Ranges of tuned hyperparameters and hyperparameter configurations of Musubu and Octet.

Method	Hyperparameter	Range	Best value(s)
Musubu	batch size	$\{128, 256, 512, 1024, 2048\}$	2048
Musubu	learning rate	$[10^{-5}, 10^{-3}]$	8.8×10^{-4}
Octet	batch size	$\{128, 256, 512, 1024, 2048\}$	128
Octet	learning rate	$[10^{-5}, 10^{-3}]$	4.5×10^{-4}
Octet	dropout rates	$(0, 0.2)$	$(9.5 \times 10^{-3}, 0.5 \times 10^{-2}, 0.20)$
Octet	dimension sizes of hidden layers	$[256, 1024]$	$[366, 753]$

Table 6: Pattern analysis of Musubu on the SemEval taxonomies. Optimal scores in the columns are in bold.

	Chemical		Equipment		Food		Science	
Pattern	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1	E-F1	H-F1
None	0.37	0.78	0.41	0.73	0.32	0.65	0.38	0.74
Such-as	0.37	**0.79**	**0.45**	0.73	0.37	**0.68**	**0.44**	**0.77**
One-of	0.40	**0.79**	0.40	0.75	0.35	0.65	0.38	0.76
Especially	0.37	0.78	0.44	0.75	0.36	0.67	**0.44**	0.75
Is-a	0.38	**0.79**	0.44	**0.77**	**0.39**	**0.68**	0.38	0.75
Including	**0.45**	0.78	0.44	0.75	0.37	**0.68**	0.40	0.76

A.4 Computing Environment and Runtime

We used an Amazon EC2 instance “p3.2xlarge" as a computing infrastructure for training and inference. Musubu and TaxoExpan took about an hour and a half to learn the parameters, and Octet took about ten to 20 minutes, excluding feature extraction. The other methods took less than ten minutes.

B Experimental Results

We evaluated the performance of Musubu with several different Hearst patterns (Table 2) for generating queries. Table 6 shows the detailed data of Fig. 4.