TaxoSeq: Taxonomy Expansion with Entire Taxonomy Structure through Sequence-to-Sequence Model

Anonymous ACL submission

Abstract

Taxonomy is a knowledge graph of concept hierarchy which plays a significant role in semantics entailment and is widely used in many downstream natural language processing tasks. Distinct from building a taxonomy from scratch, the task of taxonomy expansion aims at enriching an existing taxonomy by adding new concepts. However, existing methods often employ only part of the structural information for representing the taxonomy, which may ignore sufficient features. Meanwhile, as many recent models usually take this task in *insertion only* manner, they preserve limitations when the new concept is not an insertion to taxonomy. Therefore, we propose TaxoSeq, a method that converts the task of taxonomy expansion into a sequence to sequence setting, thereby effectively exploiting the entire structural features and naturally dealing with more expansion cases. Empowered by pre-trained language models such as T5 (Raffel et al., 2020), our approach is shown to achieve significant progress over other methods in Semeval's three publicly benchmark datasets.

1 Introduction

Taxonomy is a tree structure or directed acyclic graph composed of "is-a" relationships, which constructs valuable knowledge connections between concepts. High-quality taxonomy is critical for many downstream tasks, such as product recommendation in e-commerce (Liu et al., 2019), information retrieval in web search (Yang et al., 2020), and question answering in education (Yu et al., 2020a). Inserting new emerging concepts into a taxonomy is the task of taxonomy expansion, and the key is to insert them into the right place to ensure consistency of concept relationships.

Recent works on taxonomy expansion can be divided into two technical lines: 1) *Concept Pair Matching* methods utilize information around concepts of taxonomy, e.g., GNN (Shen et al., 2020)

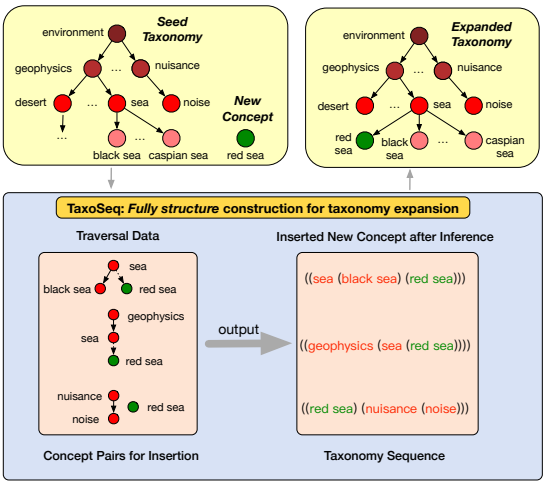

Figure 1: An illustration of a taxonomy expansion task. Modeling the entire structure of taxonomy with sequences.

for mapping the concepts into a high dimensional space or mini-paths (Yu et al., 2020b) for matching in local range of taxonomy concepts. 2) *Hierarchy Expansion* methods take taxonomy hierarchical information into account, constructing a subtree-like structure, e.g., ego-trees (Wang et al., 2021) and taxonomy-paths (Liu et al., 2021), to extract horizontal or vertical concept information of taxonomy.

Although these approaches utilize taxonomy structural knowledge, they do not explicitly establish the *entire structure* of local taxonomy, resulting in a discount in representing taxonomy. For example, in Figure 1, the "red sea" concept in expanded taxonomy forms structures that can be acute angle edges (semantic similarity), paths (hypernym relations) and individual points (no relation), where a fixed representation is only modeled from the side. Furthermore, they all insert concepts in a *"insertion only"* manner, which mainly consider semantic similarity and hypernym relations. The top-ranked concept in the seed taxonomy is chosen for

insertion and the new concept is appended below it. These methods do not explicitly distinguish no relation of new concept, which can cause unintuitive in the insertion position.

To address the issues above, we propose a novel taoxonomy expansion method by converting **Taxo**nomy into **Seq**uence (TaxoSeq) for concept insertion. In particular, we use brackets and concept position in brackets to represent an arbitrary tree structure taxonomy as a sequence, denoting various relationships in taxonomy. Then we insert the concept directly into the bracket sequence to obtain the expanded taxonomy of inserted concept by sequence-to-sequence (seq2seq) framework.

Based on our observation of the taxonomy structure, we found that taxonomy have two characteristics: layer factor and depth factor. The depth factor is created by hypernym-hyponym relationship, in which the hypernym has a semantic entailment relationship with the hyponym. The layer factor is made up of semantic similarities between concepts with the same hypernym, which we refer to as sibling concepts. We propose a bracket tree modeling approach in which concepts outside the brackets have a hypernym relationship with concepts inside the brackets, and concepts within the same bracket are sibling concepts. In this way, we converted taxonomy into a sequence regardless of its structure.

The sequence-to-sequence approach has been applied to various NLP tasks, such as Dialogue (Colombo et al., 2020) Question Answer (Saxena et al., 2022) and Translation (Wang et al., 2022). Drawing on this idea, we make a similar study on taxonomy expansion. During training, the sequence inputted to the model encodes both semantic and structural features of taxonomy. Then, combined with the generative T5 pre-train language model (PLM), we directly generate the sequence taxonomy inserted concept. This can break the case of insertion only, which identifies non-insertion and other insertion types. In addition, T5 model trained on large-scale corpora also benefits discovering the hypernym relationship.

In the experiment, we evaluate TaxoSeq in three public benchmarks in SemEval-task13. Our algorithm yields a maximum improvement of 22.1% accuracy over the state-of-the-art algorithm. Moreover, we conducted ablation experiments to verify the effectiveness of each part of TaxoSeq.

Our contributions are threefold: 1) We propose a bracket tree modeling method that can convert local tree taxonomy into a sequence regardless of its different structure. 2) We employ a new sequence-to-sequence concept insertion strategy to directly insert new concepts into the taxonomy, which can address the case of no insertion. 3) We conduct extensive evaluations on three benchmark datasets to validate the performance of TaxoSeq methods.

2 Related work

Taxonomy Construction. Automated taxonomy construction tasks have endured in academic research, which aims to construct a bunch of nodes into a directed acyclic graph or a tree from scratch. Further, it can be divided into two subfields. The first subfield clusters similar terms into a topic and constructs these topics into a taxonomy, which is called topic-level taxonomy construction (Shang et al., 2020; Downey et al., 2015). Another subfield is in term level, and the "is-a" relation extraction between terms utilizes either a pattern-based method (Panchenko et al., 2016; Chang et al., 2017) or a distributional method (Shi et al., 2019; Chang et al., 2017). For example, the Hearst pattern (Hearst, 1992) is a typical pattern-based method, and these methods have relatively high accuracy but low recall. The distribution method (Dash et al., 2020; Wang et al., 2019) can determine whether a concept pair has a hypernym relationship with each other by word embedding, but it requires a large amount of manual annotation, which affects the application in different domains. With the development of practical applications, many new concepts emerge, and the taxonomy construction will be laborious and time-consuming, thus it is critical to find a solid solution.

Taxonomy Expansion. Recent researches mainly addresses how to insert new concepts into the seed taxonomy. (Manzoor et al., 2020) learns the representation of the implicit semantic information of edges and the embedding of nodes to discover whether a node pair has a hypernym relationship. (Shen et al., 2020) proposes a location-enhanced graph neural network to encode the relative position of anchors and a noise-robust training strategy. STEAM (Yu et al., 2020b) offers a multi-view co-training procedure with the integration of multiple external sources to assist mini-path-based classification in finding the anchor node. HEF (Wang et al., 2021) constructs a subtree model of the hypernyms and sibling nodes and encodes the four representations of the subtree. Although existing

methods supplement the implicit information of the taxonomy structure with the help of auxiliary algorithms, they do not model the entire tree taxonomy structure. Moreover, they treat taxonomy expansion as a ranking task for nodes, which essentially defaults that nodes must be able to be inserted. In contrast, our algorithm can represent different taxonomy structures and flexibly insert concepts into the taxonomy with the help of pre-trained language models and the seq2seq framework.

3 Problem Formalization

In this section, we first define taxonomy, then formalize the task of taxonomy expansion.

Definition 3.1 (Taxonomy) *Taxonomy is a tree structure that can be represented by $\mathcal{T} = (\mathcal{C}, \mathcal{E})$, where $c \in \mathcal{C}$ represents the concept in taxonomy $\mathcal{T}$. $\langle c_i, c_j \rangle \in \mathcal{E}$ denotes the edge representing hypernym-hyponym relation between two concepts, where c_i is the hypernym, and c_j is the hyponym.*

For example, in the seed taxonomy in Figure 1, "sea", "black sea" and "caspian sea" are concepts. ⟨sea, black sea⟩ and ⟨sea, caspian sea⟩ are edges in the taxonomy, where "sea" is the hypernym of "black sea" and "caspian sea".

Definition 3.2 (Taxonomy Expansion) *Given a seed taxonomy $\mathcal{T}^0 = (\mathcal{C}^0, \mathcal{E}^0)$ and a set of new concepts $\mathcal{C}$, taxonomy expansion aims to form a new taxonomy $\mathcal{T} = (\mathcal{C}^0 \cup \mathcal{C}, \mathcal{E}^0 \cup \mathcal{E})$ by inserting the concepts $\mathcal{C}$ into taxonomy $\mathcal{T}^0$, where $\mathcal{E}$ is the set of new edges after insertion. Specially, for each given query concept $q \in \mathcal{C}$, the model utilizes the hypernym score function $f(q, c)$ formed with concept $c \in \mathcal{C}^0$ to determine the anchor concept $c_a = \arg\max_{c \in \mathcal{C}^0} f(q, c)$ and to form loacl taxonomy $\mathcal{T}^c$ containing q and c. If $c_a \in \mathcal{C}$,the edge $\langle c_a, q \rangle$ is added to $\mathcal{E}$.*

For the query concept "red sea" in Figure 1, the insertion function formed by "black sea" concept (this will be introduced in 4.2) has the highest ranking, the anchor concept is determined as "sea", and the local taxonomy sequence is formed as "(sea(black sea)(red sea)))". Both obtaining anchor concept and generating taxonomy inserted query in original taxonomy is very complex, so we consider pre-sorted nodes in practical application. Therefore, the key to inserting query concepts lies in how to construct the function f, which mainly solves two problems, **modeling taxonomy features** and **different insertion styles**. The following section presents our approach to solving these problems.

4 Method

In this section, we want to extract local entire taxonomy structure information and solve fixed insertion of query concepts, so we designed a bracket tree representation of taxonomy and used a pre-trained language model to generate the taxonomy after inserting query concept flexibly. As shown in Figure 2, TaxoSeq contains four key components: (1) **Bracket Tree Modeling**: given a tree taxonomy, TaxoSeq converts it into a sequence represented by a bracket tree. (2) **Concept Pair Extraction**: sampling the concept pairs from the seed taxonomy to form the training set. (3) **Concept Insertion**: generate sequences of inserted query concepts using T5 pre-trained language model. (4) **Model Inference**: determine the final anchor concept of the query concept using the complement module and pre-sorting. Next, we will introduce the details of four components.

4.1 Bracket Tree Modeling

In this section, we want to transform a given tree taxonomy into a bracket tree sequence, and the key is how to preserve the properties of the taxonomy. Therefore, we first analyse the characteristic of taxonomy, and then we give formal representation of bracket tree and the transformation rules.

Based on the observation of taxonomy, we summarize the characteristic of taxonomy structure contains three relations.

- **Hypernym Relationship**. Two concepts connected by the same edge in taxonomy have a hypernym relationship.
- **Similarity Relationship**. The hyponym concepts under the same hypernym have a semantic similarity relationship.
- **Semantic Granularity Distinction**. Concepts of different semantic granularity are located in different layers of taxonomy, where the more concrete concepts are in the lower of taxonomy and the more abstract concepts are in the upper of taxonomy.

The key to converting taxonomy into sequence is how to model these three relationships, which we designate as taxonomy relations in later section. Therefore, we design a bracket tree that nicely encompasses these taxonomy relationships.

Definition 4.1 (Bracket Tree) *Bracket tree is a sequence representing a tree-like taxonomy denoted by $\mathcal{S} = [\mathcal{C}, \mathcal{E}]$. $c \in \mathcal{C}$ is a concept in sequence.*

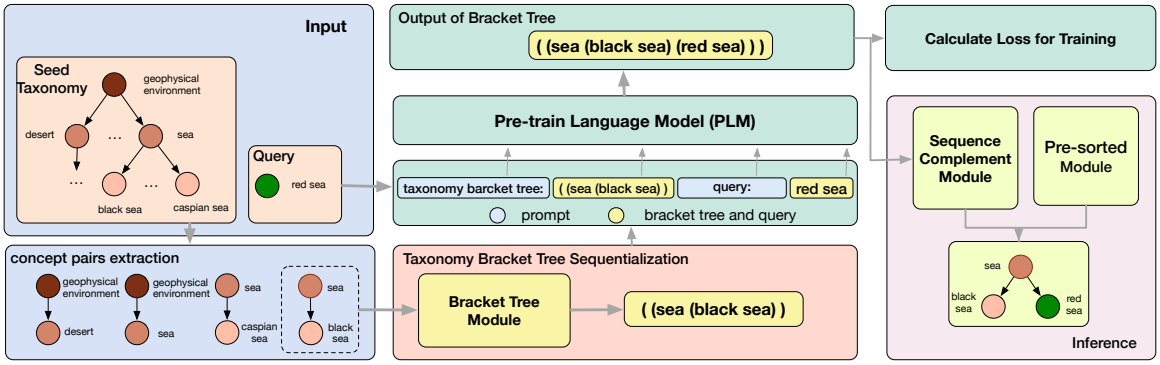

Figure 2: The illustration of the proposed taxonomy to sequence model. We give an example of a green concept as the query concept to be inserted. Each cycle represents the seed taxonomy concepts and a query concept.

$e \in \mathcal{E}$ is an edge in sequence, containing "(" and ")", denoted as $c_i(c_j)$, where the concept c_i outside the bracket is the hypernym of c_j inside the bracket.

For example, "sea", "black sea" and "red sea" in Figure 1 are transformed into a bracket tree as **((sea(black sea)(red sea)))**, where "sea" is the hypernym of "black sea" and "red sea", which is outside the brackets of "black sea" and "red sea". In this case, "black sea" and "red sea" are sibling concepts, which are inside the "sea" concept's bracket. To convert taxonomy to a bracket tree, we devise three transformational rules:

Concepts with Hypernym Relations. When two concepts have a hypernym-hyponym relation, add parentheses on both sides of the hyponym and place them on the right side of the hypernym.

Concepts with Sibling Relations. When concepts are siblings and are children of the same hypernym, they are placed to the right of that hypernym in alphabetical order.

Multiple Trees. We introduce a blank virtual concept to distinguish multiple trees, which means that the virtual concept is the hypernym of the root concept. Specifically, we enclose tree concepts with parentheses and add two parentheses to the outermost layer.

According to the above rules, the taxonomy is represented by sequence recursively. Bracket tree can allow the long-range dependencies of taxonomy and ensure its uniqueness.

4.2 Concept Pair Extraction

When given a bracket tree formed by taxonomy and a query concept q, our goal is to generate a new bracket tree after inserting query concept. However, direct operation on entire seed taxonomy does not work well because the taxonomy contains enormous amount taxonomy relationships. To distinguish different relations around concepts and identify the semantic granularity, we use taxonomy fragments to disassemble entire taxonomy. Therefore, we propose a concept pair sampling strategy. On the one hand, it can construct the entire taxonomy relations and be sampled systematically. On the other hand, concept pairs maintain the transferability of semantic information and also distinguish different levels of concepts. Specifically, we design positive and negative sampling methods:

Positive Sampling. We need the positive samples allow model to learn information around the anchor concepts, the hypernym relations and the semantic similarity relations of query concept. Through investigation, we find two characteristics. Firstly, the grandfather concept has a hypernym relation with the query concept, which can augment the anchor concept's hypernym finding. Therefore, we extract the anchor concept and its father, forming a grandfather-father concept pair $G(c)$. Secondly, the sibling concept has a similar semantic to the query concept, thereby strengthening the query structure hierarchy discovery in seed taxonomy. So we extract the anchor and the sibling of query concept forming a father-sibling concept pair $F(c)$.

For example, for the anchor concept "sea" of "red sea" in Figure 2, the grandfather-father concept pair is "geography environment - sea", and the father-sibling concept pair is "sea-black sea". Apparently, the grandfather-father concept pair has only one concept pair, while the father-sibling have multiple ones, which leads to data imbalance. Therefore, we duplicate the number of grandfather-sibling to half the number of father-sibling.

Negative Sampling. In a straightforward approach, negative concept pairs are randomly sampled $\tilde{R}(c)$, but this would make the insertion not distinguish the fine-grained hypernym relation because the ancestor concepts also have a hypernym relationship with query concepts. So we select the anchor's ancestor and the ancestor's father to form a negative concept pair $\tilde{A}(c)$, which can help finding the nearst-neighbor hypernym anchor at a fine-grained level. For example, for the "sea" anchor concept in Figure 1, we extract its ancestor concepts "environment" and "environment geography" as a negative sample concept pair.

The ratio of the number of positive samples to the number of negative samples can be tuned, and we will further analyze it during the experiment. Finally, the training data are summarized as:

$$S(c) = \underbrace{G(c) \cup F(c)}_{Positive\ Sampling} \cup \underbrace{\tilde{R}(c) \cup \tilde{A}(c)}_{Negative\ Sampling}$$

4.3 Concept Insertion as a Sequence to Sequence model

In this section, We design a sequence-to-sequence training method using the T5 pre-trained language model for concept insertion.

As mentioned of our goal in 4.2, we introduce prompts to glue the concept pair sequence and query together, forming an input as: "taxonomy bracket tree: t; query: q" in Figure 2, where t is a bracket concept pair and q is the query concept. For this sequence-to-sequence framework, which ask T5 to perform concept insertion learning during training so that the sequence inserted query concept can be generated during inference. Of course, it can be switched to any other prompt . The training loss $\mathcal{L}$ can be expressed as:

$$\mathcal{L}(\theta) = \sum_{t=1}^{k} -\log P_\theta\left(x_t | s_{token}, x_{<t}\right) \quad (1)$$

where θ is the optimization weight, s_{token} is the token of the sequence input to T5 model, x_t is the token of the sequence generated in decode, and $x_{<t}$ is the sequence token generated so far.

4.4 Inference

During inference, for the insertion of a new concept $q \in C$, we extract all concepts in the seed taxonomy $\mathcal{T}^0$ and their father to form the concept pair for input. Each concept has only one parent concept , so that all concept pairs can be included.

True : "((sea (black sea) (red sea)))"
Missing right bracket : "((sea (black sea) (red sea)))"
Word wrong : "((sea (black sea) (reptile)))"

Figure 3: Two errors – missing right bracket and word wrong.

We experimentally observe that the sentences generated by T5 have two errors, one is missing right bracket, and the other is that the words of the generated sentences are not the input words. For the first case in Figure 3, the closing parenthesis is only missing at the end of the sequence, so we can fill it according to the number of left parentheses. For the second case, "reptile" replacing "red sea" in Figure 3, the generated sentences are generally one word wrong. Due to we know the input taxonomy and query concept, we can amend the wrong word of the generated sentence to ensure word consistency. Finally, the resulting bracket tree can be equivalently converted to a taxonomy. We named the above operation as the sequence complement.

As described in Section 3, while we can design a sorting algorithm for all concepts, our method can also be used as a plug-in to obtain the final insertion position , which make further judgments on the results of other taxonomy expansion works. We utilize existing algorithms' result as prior ranking work. For example, the HEF algorithm scores the hypernym of all the concepts in seed taxonomy. Their models can be formulated as a score function $f_{pri}(c, q)$, concept c on the hypernym relation of q. The higher the score, the more hypernym relation is. Therefore, for each query concept q, all concepts c in the seed taxonomy C^0 are sorted according to their scores to obtain $R(C^0, q)$, which is the pre-sorted module in Figure 2.

We formulate query concept q inserted into concept pair formed by c as :

$$I(c, q) = \begin{cases} 1 & \text{inserted } q \\ 0 & \text{no inserted } q \end{cases}$$

The position selected for the query concept to be inserted is the highest-ranked sequence, i.e.,

$$parent(q) = \arg\max_{c \epsilon C^0} I(c, q) * f_{pri}(c, q)$$

5 Experiments

In this section, we first introduce the experiment settings, then report the overall comparison results, along with ablation studies to analyze the influence of different parameters of method components. Finally, we perform case studies and error analyses.

5.1 Experimental Setup

Datasets. We evaluate the performance of the TaxoSeq using three publicly benchmark datasets in SemEval 2016 task 13, including three taxos of the environment, science, and food. Table 1 shows the specific statistics of these three datasets. The taxonomy of these three datasets is a directed acyclic graph, which we convert into a tree referring to HEF. Following previous baseline methods, we extract 20% leaf concepts as the query concept. The number of the remaining concepts, which serve as the seed taxonomy, is displayed as N_S in Table 1.

Table 1: Statistics of SemEval datasets. $|N_O|$ and $|E_O|$ are the number of concepts and edges in the original taxonomy dataset, respectively. $|N_S|$ is the number of seed taxonomy concept in TaxoSeq, D is the depth of the taxonomy.

Dataset	D	$\|N_O\|$	$\|E_O\|$	$\|N_S\|$
Environment	6	261	261	193
Science	8	429	452	328
Food	8	1486	1576	1173

Baseline Methods. We mainly compare the following baseline methods:

- **BERT+MLP** uses BERT to determine whether two concepts merged into a sequence have a hypernym-hyponym relation, which can be regarded as a binary classification task.
- **HypeNet** (Shwartz et al., 2016) encodes concept dependency path by recurrent neural network and combines distributional signals to enhance detecting hypernym relations.
- **TaxoExpan** (Shen et al., 2020) is a self-supervised taxonomy expansion method, which uses graph neural network to integrate the information around the concept of seed taxonomy to strengthen the anchor search.
- **STEAM** (Yu et al., 2020b) learns feature representations from multiple views and performs co-training to formulate a concept attachment prediction task between anchor mini-paths and query terms.
- **HEF** (Wang et al., 2021) is the state-of-the-art taxonomy expansion model, which constructs several tree-exclusive features to enhance hypernymy relation detection.
- **TEMP** (Liu et al., 2021) uses a merging sequence formed by the path of root concept to the concept in seed taxonomy and the query concept with an explanation to find hypernym.

BERT and HypeNet are the classical hypernym relationship classification model. TaxoExpan and STEAM construct local concept information, which we compare on the performance of extracting different taxonomy feature approaches. In addition, we experiment with no complement module of TaxoSeq and BART-large for replacing T5 to verify the effectiveness of each module.

Metric. We denote the real anchor concept in the test data as $c_{a1}, c_{a2}, \ldots, c_{an}$, and the anchor concepts predicted by the model as $\hat{c}_{a1}, \hat{c}_{a2}, \ldots, \hat{c}_{an}$. Based on previous work (Wang et al., 2021; Yu et al., 2020b), we use the following metrics:

Accuracy (ACC) measures the exact proportion that predicted anchor concept inserted query concept matches the true anchor concept.

Mean reciprocal rank (MRR) measures average recipocal rank of real anchor concept formed by inserting query concept.

Wu & Palmer similarity (Wu&P) measures the distance in taxonomy between the predicted anchor concept and the true anchor concept. This metric is often used to evaluate the quality of the taxonomy structure after inserting query concept:

$$WU\&P = \frac{1}{n}\sum_{i=1}^{n}\frac{2 * depth(LCA(c_{ai}, \hat{c}_{ai}))}{depth(c_{ai}) + depth(\hat{c}_{ai})}$$

where $depth()$ is the depth of the concept in the taxonomy, and $LCA(,)$ is the nearest common ancestor of the two concepts in the taxonomy.

5.2 Implementation Details

The above baseline methods, except for BERT-MLP, are obtained from the source code published by the author. We perform 5 experiments on each baseline and select the best result, which are compared with the result of TaxoSeq. Additionally, we choose an equal number of negative concept pairs as positive concept pair at random. For each query concept during inference, we first use the HEF to obtain a prior ranking of the seed taxonomy concept for no complement and BART-large experiments. Then we chose T5-large as our primary experimental for HEF prior sorted concepts and TEMP prior sorted concepts. We adopt AdamW to optimize the parameters and employ a linear warm-up that accelerates the learning rate from 0 to a maximum value (9e-4) from an initial 10 percent step, then drops to 0 in subsequent steps.

Table 2: Comparison of TaxoSeq and other baselines. All metrics are presented in percentages (%). The best results are marked in bold. No Complement is that the TaxoSeq method does not add the sequence complement module during inference. T5-large replaced with BART-large in HEF pre-sorted module denoted by BART.

Dataset	*SemEval16-Env*			*SemEval16-Sci*			*SemEval16-Food*		
Metric	Acc	MRR	Wu&P	Acc	MRR	Wu&P	Acc	MRR	Wu&P
BERT+MLP	11.1	21.5	47.9	11.5	15.7	43.6	10.5	14.9	47.0
HypeNet	16.7	23.7	55.8	15.4	22.6	50.7	20.5	27.3	63.2
TaxoExpan	11.1	32.3	54.8	27.8	44.8	57.6	27.6	40.5	54.2
STEAM	36.1	46.9	69.6	36.5	48.3	68.2	34.2	43.4	67.0
HEF	50.0	60.1	69.3	51.6	61.5	73.5	36.6	48.3	68.8
TEMP	51.9	65.9	**79.4**	54.1	64.6	83.0	47.8	**60.6**	**80.9**
No Complement	56.8	63.6	68.2	58.2	66.7	82.0	42.5	52.8	63.9
BART	56.8	64.1	68.9	59.3	60.0	80.5	53.1	58.6	76.8
HEF+TaxoSeq	58.6	**66.7**	73.3	59.3	**67.7**	**83.8**	44.5	53.9	70.3
TEMP+TaxoSeq	**63.4**	66.4	67.3	**60.0**	62.7	71.3	**53.8**	55.4	70.1

5.3 Experimental Results

As shown in Table 2, our proposed TaxoSeq outperform state-of-the-art HEF and TEMP with the greatest improvement of 21.5% and 22.1% in ACC, respectively. TaxoExpan, STEAM, HEF and TEMP construct the information around anchor concept, and the performance is improved dramatically over the other two baselines, which also validates that our bracket tree can improve the performance. In particular, firstly, ACC has the largest improvement rate among the three metrics, showing that TaxoSeq is more helpful in finding sorted first-anchor concept. Secondly, the performance of the food dataset is worse than the other two datasets, indicating that the semantic information of concepts from different domains impacts concept insertion. When we perform experiments with no complement module, the overall performance is 2% to 5% lower to HEF+TaxoSeq. This phenomenon is because the pre-trained language model does not have prior knowledge of the bracket tree, so two types of errors occur to reduce the performance. Bart-large also outperforms HEF on all three datasets, but it outperforms T5 on the food dataset. This indicates that different pre-training language models influence the improvement in different domains.

5.4 Ablation Experiments

To further explore different components of TaxoSeq on performance, we also performed two experiments. One is control the number of negative of $\tilde{A}(c)$, the other is to investigate the ratio of negative samples to positive samples. We perform experiments on the science dataset and apply the optimized parameters to the other two datasets.

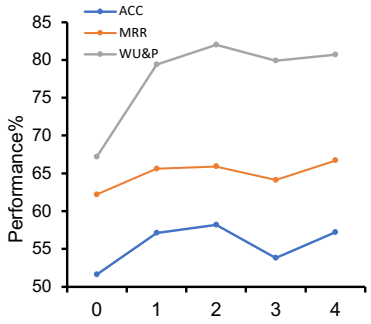

Figure 4: The performance on different number of negative ancestor concepts.

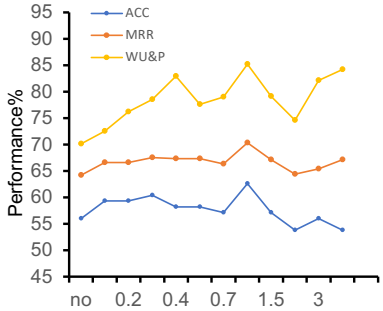

Figure 5: The performance on different ratios of negative sample to positive sample.

Negative Ancestor. We repeat the number of negative ancestor concept pair from none to 5 times and observe the changes in Figure 4. The worst experimental results were obtained when no ancestor concept pair were added as negative samples. This is because the ancestor concept has a hypernym relationship with the query concept. Without the ancestor concept as a negative sample, the model will find the wrong hypernym word. In addition, the experimental results are the best at 2 times the number of negative ancestor concept pair samples.

Ratios of Negative to Positive. We also exper-

Table 3: The analysis of 3 error types is based on the failure of TaxoSeq cases in science dataset. "#" refers to the number of error types, and the red concept is the query concept inserted into taxonomy.

Error Type	#	Example
Bypass Anchor Insertion	12	"predict sentence": "((physics(optics)(holography)))" "true sentence": "((physics(optics(holography))))"
No Insertion	15	"predict sentence": "((geostrategy)(politics(geopolitics)))" "true sentence": "((politics(geopolitics(geostrategy))))"
Wrong Ranking	10	"anchor concept": "chemistry" "inital rank":10 "predict sentence": "((chemistry(thermochemistry)(femtochemistry)))" "true sentence": "((chemistry(thermochemistry)(femtochemistry)))" "inital ranking": 5

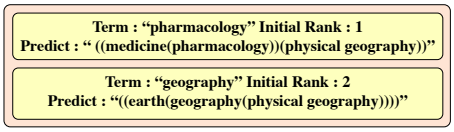

Figure 6: A case study on the concept of 'physical geography' inserted into concept pair and improving the ranking of anchor concept.

iment with different negative sample to positive sample ratios from none to all in Figure 5. We can see the curve first increase and then decrease as the negative samples increase. In the absence of negative sample, TaxoSeq is equivalent to sorting all concepts. Even so, our experimental results are better than baselines. This can be attributed to the design of concept pair that will extend the search of anchor concepts. The best result is at the same proportion of positive and negative samples. However, as negative cases continue to increase, the accuracy rate decreases, whereas too many negative samples will overfit the model to misclassification.

5.5 Case Studies and Error Analysis

Case Studies. Figure 6 shows a case study on the query concept of "physical geography" inserted into seed taxonomy. In the first bracket tree, although the concept "pharmacology" was initially ranked first, the query concept is not inserted, so "pharmacology" is eliminated. "geography" is chosen as the anchor concept in the second bracket tree and directly formed the result after insertion. This example shows that TaxoSeq can eliminate over-ranked non-anchor nodes, and the insertion method is very straightforward and can continue to be extended in future work.

Error Analysis. To further explore the bottlenecks of TaxoSeq, we analyze all 37 errors resulted in the science dataset and classify them into three categories in Table 3: (1) **Bypass anchor insertion**. Query concept bypasses the anchor concept and inserts directly into the grandfather concept, e.g., the anchor for "holography" is "optics", but "physics" is selected. Ten of these errors had only one grandfather-father concept pair inserted (single-edge) in a way that does not have sibling concept, accounting for 83% of these errors. (2) **No insertion**. The query concept isn't inserted into the anchor concept, e.g., the "geostrategy" concept should be inserted under "geopolitics". Although there are many concept pair containing anchor for each query concept in this type of error, none of them have been identified. (3) **Wrong ranking** fails to rank the anchor concept first, but it does move it forward, e.g.,the ranking of the anchor for "femtochemistry" is lifted from 10 to 5.

In future work, we focus on addressing single-edge hypernym concept discovery and allowing the model to identify no finding anchor concept with a view to being able to resolve bypass anchor insertion and no insertion.

6 Conclusion

We propose a TaxoSeq model that converts taxonomy into sequence and flexibly inserts query concepts into taxonomy using the Seq2Seq architecture. We use the bracketed tree serialization method and propose three conversion rules to transform any tree taxonomy into its corresponding sequence. We systematically design a strategy for sampling the training set from the seed taxonomy, considering the hypernym relations of ancestor nodes and the semantic similarity of sibling nodes during insertion, and also finding the most fine-grained hypernym. The experimental results in SemEval-task 13 show that we outperform state-of-the-art algorithms.

Ethic Considerations

In this paper, we conduct an exploration of converting the taxonomy expansion task into seq2seq paradigm and exploiting pretrained language models (PLMs) in lifting performance. However, we only provide primary results on english datasets, which limits the use of this method in other language scenarios. Meanwhile, we just provide the insight of this strategy by one example implementation as bracket trees. As there are plenty of other approaches for converting the taxonomy into sequence, we hope our work can call for more technical attempts via such ideas.

References

Haw-Shiuan Chang, Ziyun Wang, Luke Vilnis, and Andrew McCallum. 2017. Distributional inclusion vector embedding for unsupervised hypernymy detection. *arXiv preprint arXiv:1710.00880*.

Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7594–7601.

Sarthak Dash, Md Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, and Nicolas Rodolfo Fauceglia. 2020. Hypernym detection using strict partial order networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7626–7633.

Doug Downey, Chandra Bhagavatula, and Yi Yang. 2015. Efficient methods for inferring large sparse topic hierarchies. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 774–784.

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

Bang Liu, Weidong Guo, Di Niu, Chaoyue Wang, Shunnan Xu, Jinghong Lin, Kunfeng Lai, and Yu Xu. 2019. A user-centered concept mining system for query and document understanding at tencent. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1831–1841.

Zichen Liu, Hongyuan Xu, Yanlong Wen, Ning Jiang, Haiying Wu, and Xiaojie Yuan. 2021. Temp: Taxonomy expansion with dynamic margin loss through taxonomy-paths. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3854–3863.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*, pages 2044–2054.

Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. *arXiv preprint arXiv:2203.10321*.

Jingbo Shang, Xinyang Zhang, Liyuan Liu, Sha Li, and Jiawei Han. 2020. Nettaxo: Automated topic taxonomy construction from text-rich network. In *Proceedings of The Web Conference 2020*, pages 1908–1919.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi 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*, pages 486–497.

Yu Shi, Jiaming Shen, Yuchen Li, Naijing Zhang, Xinwei He, Zhengzhi Lou, Qi Zhu, Matthew Walker, Myunghwan Kim, and Jiawei Han. 2019. Discovering hypernymy in text-rich heterogeneous information network by exploiting context granularity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 599–608.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076*.

Chengyu Wang, Yan Fan, Xiaofeng He, and Aoying Zhou. 2019. A family of fuzzy orthogonal projection models for monolingual and cross-lingual hypernymy prediction. In *The World Wide Web Conference*, pages 1965–1976.

Suyuchen Wang, Ruihui Zhao, Xi Chen, Yefeng Zheng, and Bang Liu. 2021. Enquire one's parent and child before decision: Fully exploit hierarchical structure for self-supervised taxonomy expansion. In *Proceedings of the Web Conference 2021*, pages 3291–3304.

Wenxuan Wang, Wenxiang Jiao, Yongchang Hao, Xing Wang, Shuming Shi, Zhaopeng Tu, and Michael Lyu. 2022. Understanding and improving sequence-to-sequence pretraining for neural machine translation. *arXiv preprint arXiv:2203.08442*.

Carl Yang, Jieyu Zhang, and Jiawei Han. 2020. Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In *ICDM*.

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, et al. 2020a. Mooccube: a large-scale data repository for nlp applications in moocs. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3135–3142.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020b. Steam: Self-supervised taxonomy expansion with mini-paths. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.