

Graph Neural Networks for Automatic Term Generation: Defining and Exploring Terminological Graph

Toshihide Fujii and Amir Hazem and Kyo Kageura

Library and Information Science Laboratory,

Graduate School of Education. The University of Tokyo.

fujii1234@gmail.com and amir.hazem@gmail.com and kyo@p.u-tokyo.ac.jp

Abstract

Despite the advance in automatic term acquisition (ATA) research, the use of ATA in terminology work is limited. One of the main reasons is that most ATA research focuses on automatic term extraction (ATE) from textual corpora and does not consider terminologies. This paper introduces automatic term generation (ATG), a new method of ATA, which explores existing terminologies, in which timeline information is also considered. We propose a new way of representing the structure of a terminology as a graph, which consists of nodes that represent constituent elements of terms and two types of links that represent intra-term dependency between constituent elements and term closure between the first and the last constituent element of terms. ATG is then defined as a set of link prediction tasks, i.e. predicting two types of links. Cliques of the graph represent the new generated terms. They are determined in a backward fashion by first selecting nodes for which term closure links are predicted and then by walking backward to term nodes for which intra-term dependency links exist. Experiments showed that our approach gives high performance in predicting new terms.

1 Introduction

Despite the development of automatic term acquisition (ATA) research (Heylen and Hertog, 2015; Tran et al., 2023; Xu et al., 2025), advanced methods are not used much in practical terminology work. One of the main reasons for this is that ATA research has mostly focused on automatic term extraction (ATE) from corpora. Correspondingly, the structure of terminologies, although it binds generation of new terms, has not been explored sufficiently. To address the limits of ATE and in order to explore the structure of terminologies, we introduce a new approach to automatic term acquisition (ATA), i.e. automatic term generation (ATG), which makes use of existing terminologies. About

80 percent of terms in most languages in most domains are complex (Cerbah, 2000). Complex term formation is the main way of generating new terms (Sager, 1990), and tends to be systematic (Felber, 1984; Kageura, 2022). It is thus expected that the generation of new terms can be predicted by exploring the structure of terminologies. Our approach starts from defining the terminological graph based on a given terminology, which has nodes that consist of constituent elements of terms and two types of links¹, i.e. intra-term dependency links between constituent elements of a term and term closure links that connect the first and the last element of a term. ATG is then defined as a link prediction task on the terminological graph. Finally, the cliques in the graph represent the terms.

The main contributions of this work are: (a) it proposes ATG, a new approach to ATA, which complements existing ATE; (b) it proposes terminological graph that incorporates essential information of terminologies by means of two types of links and applies advanced graph processing methods; (c) it evaluates the results by taking into account timeline perspective; (d) it gives comparative evaluation with baselines including LLMs. With respect to practical terminology work, it explored the information contained in concrete terminologies, the compilation and revision of which is claimed to be the major objective of ATA research. This work provides a point of reference for new ATG framework of ATA research.

2 Related Work

2.1 Automatic Term Acquisition

Research on monolingual ATE has employed a range of approaches, including linguistic patterns, corpus statistics, and supervised or unsupervised learning, to identify term candidates (Kageura and Umio, 1996; Heylen and Hertog, 2015; Tran et al.,

¹We use links and nodes for edges and vertices.

2023; Xu et al., 2025). More recent work has further incorporated pretrained embeddings and LLMs (Al-Thubaity, 2025; Tran et al., 2025). Despite these advances, relatively few methods explicitly make use of the target terminology (Bordea et al., 2013). This is an important limitation, as candidates extracted from corpora do not necessarily correspond to the terms required in practical terminology work (Kageura, 2024).

A different approach has been proposed as a part of bilingual terminology acquisition task. Noting the fact that the many terms are complex, Sato et al. (2013) defined a head-modifier bipartite graph using a given terminology and generated new complex term candidates by interpolating missing links. Iwai et al. (2016) improved this approach by using the terminology graph introduced in Kageura (2012). Their work, which adopted ATG approach, showed promising performance but remained embryonic. So the full exploration of terminological structure for monolingual ATA is yet to be done.

2.2 Graph Processing for Link Prediction

As we will see in section 3, our approach defines automatic term generation as a link prediction task. The terminological graph we define consists of two types of links or multi-relational settings. We review here main work for link prediction tasks.

Within this link prediction formulation, graph-based methods provide a natural framework for modeling the compositional structure of terminologies, as in representing knowledge graphs (Hogan et al., 2022; Jiang et al., 2025). Among the various methods developed for graph learning, Graph Neural Networks (GNNs) have achieved high performance across a wide range of applications, including link prediction. Representative approaches include Graph Auto-Encoders (GAEs) and Variational Graph Auto-Encoders (VGAEs), which use a Graph Convolutional Network (GCN) (Kipf and Welling, 2017) as an encoder to learn node representations and reconstruct links between them (Kipf and Welling, 2016).

Many real-world graphs, however, contain multiple types of relations. GNNs that treat all links uniformly cannot adequately capture interactions specific to each relation type. Multi-relational GNNs address this limitation by performing message passing separately for different relations. Relational Graph Convolutional Networks (R-GCNs) (Schlichtkrull et al., 2018) are a representative example of this approach. For link prediction on

such graphs, an encoder such as R-GCN can be combined with a relation-aware decoder such as DistMult (Yang et al., 2015), allowing the model to predict both whether a link exists and which relation it represents.

In our setting, we consider two graph neural network architectures as encoders. Graph Attention Networks (GATs) aggregate information from neighboring nodes through attention mechanisms (Velickovi et al., 2017) and serve as our baseline for the single-relation setting. R-GCNs, by contrast, naturally accommodate our two-relation representation through relation-specific message passing. We therefore combine an R-GCN encoder with a ComplEx decoder (Trouillon et al., 2016) to score directed, relation-specific links.

3 Proposed Approach to ATG

Taking as a point of departure the ATA work that used the graph representation of terminologies (Sato et al., 2013; Iwai et al., 2016), we introduce a new method for monolingual term acquisition, i.e. prediction of new terms through generative terminology modeling starting from given terminologies. It starts from constructing a terminological graph from an existing terminology. New term prediction task is then defined as a link prediction task².

3.1 Defining Terminological Graphs

The terminological graph is defined as follows:

- Nodes: constituent elements of terms, with either pretrained FastText features or one-hot identity features;
- Links: two types of directed links i.e. (a) type 1 links, which represent intra-term dependency relations between constituent elements, directed from modifier to head, and (b) type 2 links, which represent term closure that connects between the first constituent element and the final constituent element of a term. As of now, the links and nodes are not weighted.

Figure 1 shows an example of a terminological graph constructed from a small set of terms.

Graphs have been widely used for representing knowledge structures (Hogan et al., 2022; Jiang et al., 2025), but have been underexplored in representing and processing terminologies. As a few

²We provide our code on GitHub: <https://github.com/utlis/two-types-link-gnn>

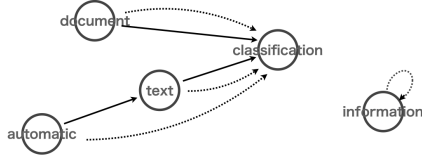


Figure 1: A terminological graph made from four terms: “information,” “text classification,” “automatic text classification,” and “document classification.” Solid arrows show type 1 links and dotted ones show type 2 links.

existing studies that used graph representation remained rudimentary in terms of both the representation of terminological structure and the processing methods (Kageura, 2012; Iwai et al., 2016), it is the first graph representation that incorporates all the essential features of terminological structures.

3.2 Exploring terminological graphs

Automatic term generation is defined as link prediction tasks for the terminological graph.

The following three tasks are defined:

- (a) Detecting type 1 links or intra-term dependency links between two nodes;
- (b) Detecting type 2 links or term closure links;
- (c) Detecting terms (or term reconstruction from cliques), which corresponds to integrating the predictions of type 2 and type 1 links.

Once ATG is defined as a set of link prediction tasks over the terminological graphs, we can apply a series of link prediction methods to the task (Smida et al., 2025). Some important methods were briefly reviewed in Section 2.2. The methods we adopted will be given in Section 4.3.

4 Experimental Setup

4.1 Dataset

To construct our dataset, we used as a starting point lists of terminologies in four domains: Agriculture, Computer Science, Dentistry and Physics³. The first column of Table 1 illustrates the size of these lists. To build a coherent list of terms that is used in a controlled scenario with reasonable size, we refine these lists by projecting them in an existing corpus. As a target corpus for projection, we used the English part of the ASPEC-JE Parallel corpus⁴ (Nakazawa et al., 2016), which contains

³See the appendix E for references.

⁴<https://jipsti.jst.go.jp/aspec>

Domain	Terms	Terms	
		$\cap$ Aspec	$\cap$ Google Ngrams
Agriculture	15,618	5,921	5,263
Computer	14,186	3,708	3,175
Dentistry	14,480	6,316	5,641
Physics	9,984	5,533	4,845

Table 1 Statistics of the four terminologies and their intersection with ASPEC and Google Ngram datasets. Table 8 in the appendix gives statistics per ngram size.

abstracts of 24 scientific domains including the four domains that we focus on in this work. The size of the resulting lists is reflected in the second column of Table 1. Finally, as our ultimate goal includes the study of the growth of terminologies from a diachronic perspective, we used Google Ngrams dataset (Lin et al., 2012), which provides timestamp information of ngrams. The size of the corresponding termlists is illustrated in the third column of Table 1.

4.1.1 Timeline Information

Based on Google Ngrams dataset (Lin et al., 2012), Norman and Aizawa (2015) used neology as a way to extract terms. They have shown that there is a certain point in time at which terms gain momentum⁵. The peak value reflects the period of time in which the term was adopted. Following Norman and Aizawa (2015), we adopt this indicator and use it to timestamp our dataset⁶. To determine the first appearance of an ngram as a term⁷, we compute the mean frequency and standard deviation of a term over the timeline⁸. We order the terms based on the mean frequency of the first appearance of terms and split our data accordingly. Terms in the dev and test sets thus appear later in the timeline compared to the terms used in the training.

4.1.2 Term Distribution

Figure 2 illustrates the distribution of terms per ngram size. Unigrams and bigrams are the most frequent in all domains. Trigrams are of a relatively small size. Finally, fourgrams are almost null and thus are not addressed in our experiments.

Figure 3 illustrates the estimation of the first appearance of terms per period of time. The year 1850 for instance, indicates the terms that appeared before 1850 while the year 1875 indicates the terms

⁵A surge or a peak in frequency.

⁶See Table 7 in the appendix

⁷For terms, we use ‘ngram’ as terms with n constituents.

⁸The latest available version of the Google Ngrams dataset until 2020 (Version 20200217).

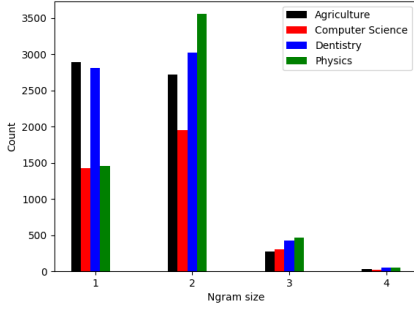


Figure 2: Ngram size distribution of terms ($n = [1,4]$) in the four studied domains.

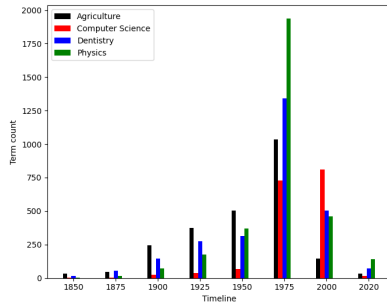


Figure 3: Term distribution of bigrams based on the estimated first time appearance in google book ngrams.

that appear between 1850 and 1875 and so on. We see a large number of new terms that appeared around 1975. This is particularly noticeable for the domain of Physics. A large number of new terms has also been introduced around the year 2000 in the computer science domain.

Given this timeline, we can reasonably evaluate term prediction in a real word scenario. We first order terms according to their estimated time of first appearance, assigning earlier terms to the training set and later terms to the development and test sets. This temporal split allows us to examine how term prediction performance changes compared with a conventional random split of the dataset. More importantly, the temporal setting naturally corresponds to the prediction of new terms, i.e. given a set of terms available at a certain point in time, the task is to predict terms that appear subsequently. These new terms may be units that have not yet been adopted as terms, or longer terms formed by combining existing terms as constituents. To the best of our knowledge, this is the first work to address term prediction under such a temporally structured scenario.

4.2 Constructing Terminological Graphs

To construct the terminological graphs, terms were lowercased and tokenized using NLTK library (Bird and Loper, 2004). The tokenized units constitute the nodes. We used two different settings for node features, i.e. with and without embeddings. In the setting with embeddings, we assign to each node a pre-trained FastText word vector (Bojanowski et al., 2016). In the setting without embeddings, we use one-hot vectors derived from the identity matrix as node features. The latter setting would show how much surface structure of terminologies themselves bind individual terms.

The terminological graph also incorporates nodes which are not terms. These ngrams are extracted from corresponding in-domain corpus. For training we consider two scenarios: i) in the first one (noted exp1), we train the graph only on nodes which are term constituents. ii) in the second one (noted exp2), we train the graph using nodes which can be terms or non terms⁹. For testing, in both scenarios the positive candidates are terms and the negative candidates are non terms that appear in the corpus. As defined in Section 3.1, two types of links exist in the terminological graph. Type 1 links are introduced between adjacent elements within a term. Type 2 links are made between the first and the last element of a term, thereby representing the term closure information.

4.3 Link and Term Prediction Setups

To perform link prediction with a GNN, it is necessary to learn the graph structure using an encoder and to compute link existence probabilities using a decoder. We use R-GCNs (Schlichtkrull et al., 2018) as the encoder. R-GCNs learn distinct weight matrices for each relation type and capture diverse relational patterns in heterogeneous graphs. As the decoder, we choose ComplEx (Trouillon et al., 2016). ComplEx represents node and relation embeddings in a complex vector space and assigns a score to a triple (s, r, o) by computing $\text{Re}(\langle s, r, \bar{o} \rangle)$. Here s , r and o represents subject or source node, relation type and object or target node, respectively. This scoring function is well suited to modeling asymmetric relations, including those arising in directed graphs considered in this study¹⁰.

⁹Non-term ngrams constitute our negative sample.

¹⁰While Graph transformers or global attention GNNs tend to show better performance in many applications that require long-range dependencies (Smida et al., 2025), the terminological graph does not require explorations of long-range dependencies.

In the training phase, we use the binary cross-entropy (BCE) as the loss function. We apply ReLU activations, dropout, and label smoothing to mitigate overfitting. We also introduce link dropout during message passing to prevent memorization of training links. For each positive link (s, r, o) , we generate negative examples by randomly replacing the source or target node with a sampled node (s') or (o') , yielding (s', r, o) or (s, r, o') . Replacement nodes are sampled from a mixture distribution, with 50% drawn uniformly at random and 50% drawn from a degree-proportional distribution, yielding negative samples of varying difficulty. We ensure that the generated negatives do not appear in the positive sets of the training, validation, or test splits, preventing information leakage. We set the positive-to-negative ratio to 1:1, and to reduce memorization of negative samples, we refresh the negative set every five epochs. At inference time, we search for the threshold that maximizes the F1 score on the validation set and apply this threshold to binary classification on the test set.

For term reconstruction, we first use each predicted type 2 link to identify the boundary of a candidate term, with its source and target nodes specifying the start and end points, respectively. Given these boundary nodes, we then search for a simple path composed of predicted type 1 links that connects the source to the target. To deal with computational constraints, we impose a cutoff length and discard overly long paths. For each discovered path, we map the sequence of nodes to their corresponding constituent elements and concatenate them to generate a candidate term. This procedure enables the systematic extraction of novel domain-term candidates based on the learned compositional relations between constituent elements.

5 Evaluations

We use two types of baselines: the first one is graph-based and the second one is LLM-based.

5.1 Graph Attention Networks

As the first baseline, we apply a Graph Attention Network (GAT) model (Velikovi et al., 2017) using only one type links. This baseline is designed to evaluate the effect of introducing a second link type by comparing the single-relation and two-relation settings. In the single-relation setting, no closing link is considered. This introduces ambiguity for

term reconstruction when the number of term constituents is greater than 2. Hence, this method can only be applied for bigram evaluation¹¹. See Section C.2 for the hyperparameter details.

5.2 LLM

As the second baseline, we use LLMs because they have shown great capabilities in many downstream applications. LLMs are by essence for generation and thus match our term generation task. To keep the level playing field, we provide LLM with a training list of nodes. To evaluate LLMs, we consider two prompting scenarios: 1) Open question and 2) Terminology list expansion. In the open question scenario, two cases are considered. In the first case, we only ask the LLM to provide a list of terms by specifying the domain name. In the second case, we ask the LLM to generate terms by specifying the name of the domain and a list of term examples that we select from the training set. In the terminology list expansion scenario, we provide examples of lists of tokens and their corresponding terms. This scenario is the closest to the graph-based approach¹². This evaluation underlines two sub-tasks: The first one is recognizing the unigram terms from a list containing unigram terms. The second task consists of generating terms out of a list of constituents which may not exist as terms on their own (evaluation of bigram and trigram terms). We provide the LLM with a list of shuffled unigram constituents and we expect the LLM to generate (reconstruct) bigrams and trigrams based on the input list¹³. This second subtask is also a new contribution as no previous work to our knowledge has evaluated LLMs under these configurations. We focus on evaluating llama3.3 and Qwen2 (Yang et al., 2024) in the next section. Additional results for llama3.1:8b, llama3.1:70b, Gemma2 and Mistral are reported in Table 17 in the appendix.

6 Results

6.1 Graph-based Approaches

Table 2 shows the results of bigram term generation, while Table 3 gives the results of term generation for all ngrams (1 to 3). Both show random and timeline setups. As can be seen in Table 2, our proposed method (exp2 (embed)) outperformed both

¹¹To our knowledge, GAT is applied for the first time to term generation. It can be considered as additional contribution.

¹²Corresponds to the vertices used in the graph.

¹³See Table 19 for complete prompt examples. The open question scenario is given for complementary observation.

Methods	Bigram Evaluation (Random list)											
	Agriculture			Computer			Dentistry			Physics		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
llama3.3	3.6	6.42	4.61	4.75	12.28	6.85	41.18	1.16	2.26	24.06	11.67	15.72
Qwen2	17.65	0.55	1.07	13.16	8.95	10.65	5.26	0.17	0.33	23.43	14.21	17.69
GAT	55.95	66.01	60.56	57.36	57.36	57.36	52.33	61.61	56.59	59.87	74.29	66.31
GAT (embed)	63.91	90.47	74.91	66.47	83.54	74.03	62.33	89.68	73.54	61.55	92.19	73.81
exp1	54.87	67.15	60.39	52.14	93.35	66.91	57.23	71.23	63.47	72.74	70.18	71.43
exp1 (embed)	87.90	54.67	67.42	76.39	76.98	76.68	84.94	49.42	62.48	77.86	82.13	79.94
exp2	88.02	76.88	82.07	85.10	75.95	80.27	89.18	72.23	79.81	87.89	81.71	84.69
exp2 (embed)	87.99	86.05	87.01	87.60	79.53	83.37	87.70	89.58	88.63	87.18	88.04	87.61

Methods	Bigram Evaluation (Timeline list)											
	Agriculture			Computer			Dentistry			Physics		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
llama3.3	5.78	8.14	6.76	30.17	16.51	21.34	14.29	0.57	1.10	16.53	13.21	14.68
Qwen2	4.76	0.21	0.40	16.40	12.54	14.21	21.43	0.57	1.11	25.08	12.40	16.60
GAT	51.50	44.11	47.52	56.15	44.65	49.74	46.85	41.14	43.81	55.49	60.20	57.75
GAT (embed)	63.24	82.87	71.73	60.24	75.54	67.03	67.08	72.95	69.89	57.98	84.18	68.66
exp1	70.68	35.11	46.92	49.05	87.76	62.93	71.82	39.80	51.22	54.44	69.98	61.24
exp1 (embed)	79.32	35.33	48.88	64.19	89.90	74.90	85.14	40.38	54.78	85.65	66.23	74.70
exp2	82.69	60.38	69.80	83.88	70.03	76.33	88.76	60.19	71.73	87.77	77.32	82.22
exp2 (embed)	83.10	79.01	81.00	80.68	79.20	79.93	86.86	88.19	87.52	88.40	83.36	85.81

Table 2 Results for bigrams term prediction in four domains (Agriculture, Computer Science, Dentistry and Physics). Results are presented based on random and timeline splitting.

Methods	All ngram Evaluation (Random list)											
	Agriculture			Computer			Dentistry			Physics		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
llama3.3	8,50	15,45	10,96	40,10	30,99	34,96	18,77	16,91	17,79	48,50	30,64	37,55
Qwen2	15,19	1,27	2,34	44,86	37,95	41,11	30,79	2,17	4,05	43,81	39,50	41,54
exp1	61.53	62.73	62.12	57.17	72.35	63.87	80.34	30.00	43.69	61.04	80.73	69.52
exp1 (embed)	75.71	44.73	56.24	79.34	45.79	58.07	77.26	46.92	58.39	78.29	50.41	61.33
exp2	81.01	50.00	61.83	84.19	50.54	63.16	85.01	45.73	59.47	82.24	67.67	74.24
exp2 (embed)	86.76	47.28	61.20	87.26	51.08	64.44	89.21	48.20	62.59	87.30	66.57	75.54

Methods	All ngram Evaluation (Timeline list)											
	Agriculture			Computer			Dentistry			Physics		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
llama3.3	20.39	19.70	19.06	55.07	33.76	41.85	51.60	7.15	12.56	43.22	43.67	43.44
Qwen2	27.08	1.89	3.53	43.92	37.64	40.53	19.36	1.28	2.40	47.63	38.17	42.37
exp1	59.28	83.09	69.19	53.15	78.26	63.31	66.03	80.24	72.45	59.27	87.62	70.71
exp1 (embed)	60.93	22.22	32.56	65.06	29.92	40.99	64.71	27.45	38.55	71.65	32.06	44.30
exp2	78.31	38.74	51.84	77.14	46.77	58.23	87.65	86.80	87.22	81.85	59.07	68.62
exp2 (embed)	83.53	32.76	47.06	81.69	45.66	58.58	84.71	87.86	86.26	87.24	57.11	69.03

Table 3 Results for all ngrams term prediction in four domains (Agriculture, Computer Science, Dentistry and Physics). Results are presented based on random and timeline splitting.

baselines GAT and the LLM-based approaches in all settings reaching the highest F1-score. Our proposed method exp1, where training excludes out-of-domain ngrams, outperformed GAT baseline in two out of four domains (Physics and Computer) suggesting that including external data in the train-

ing setup is not always necessary to build efficient models for ATG. Table 3, where the test set contains terms of different size (1 to 3 grams), shows differences between exp1 and exp2. In the random setting, exp2 outperformed exp1 in terms of F1 score in three out of four domains, while in

the timeline setting the contrary happened. If the random and timeline experiments can not be directly compared, as the test sets are different, we can nonetheless remark that including or excluding general words (ngrams) while training the graph impacts the performance of ATG.

LLM-based approaches perform poorly, in terms of both recall and precision for all the domains, although the overall performance for all ngram term prediction by LLMs shows much improvement compared to bigram predictions. Incidentally, we also note that the results of LLMs depend highly on the domain. The results of GAT and the results of bigram link predictions using our methods in Table 15, on the other hand, are more or less comparable. This indicates that the key factor of the high performance of our method resides in the exploration of the two types of links.

The effect of embeddings in term prediction task in our method is inconclusive as shown in Tables 2 and 3, while GAT produces consistently better results with embeddings than without them both in terms of precision and recall. The use of embeddings in our method improves precision but generally deteriorates the recall. As for the difference between random split and timeline split setups, our method produces comparable results in both setups. This contrasts with the results of GAT, for which timeline split produces consistently lower results than random split. LLMs tend to perform better with the setup corresponding to timeline split. Interestingly, Table 15 shows that link prediction by our method have the same tendencies as GAT. Table 16 shows link prediction results by link type.

6.2 LLMs

Table 4 shows the results of term generation for open-question (0 and n-shot respectively noted OpenQ-0 and OpenQ-n) and list expansion prompting (Expan-n) for llama3.3. As can be seen, the results for open question scenario are very low although few-shot learning improves the performance. Expan-n shows better performance but the results are still very low compared to graph-based approaches.

Table 5 illustrates the impact a few-shot learning on LLM performance. Although the number of shots impacts the performance, this remains dependent on the model and on the settings. llama3.3 for instance obtains the best performance with 3-shot on the random test while with 2-shot in the timeline test. Other factors may impact the results such as

llama3.3	Computer		
	P	R	F1
OpenQ-0	0.96	4.88	1.6
OpenQ-n	5.77	3.23	4.14
Expan-n	29.73	11.83	16.93

Table 4 Comparison of the performance of llama3.3 using open question prompting with 0-shot (OpenQ-0), few-shot (OpenQ-n) and expansion prompting (Expan-n) with $n = 3$ in the Random test setting for trigrams.

the examples given in the prompt.

	Computer					
	Random			Timeline		
	P	R	F1	P	R	F1
llama3.3						
1-shot	7.14	4.92	5.83	30.0	21.95	25.35
2-shot	19.44	11.48	14.44	29.63	39.02	33.68
3-shot	18.33	18.03	18.18	27.66	31.71	29.55
Qwen2						
1-shot	25.0	1.64	3.08	15.91	17.07	16.47
2-shot	21.43	4.92	8.0	8.41	21.95	12.16
3-shot	6.15	13.11	8.37	50.0	4.88	8.89

Table 5 Results for llama3.3 and Qwen2 using expansion prompting (Expan-n) with 1, 2 and 3-shot prompting. The results are shown for trigrams in the random and timeline test settings.

llama3.3	Computer		
	P	R	F1
Expan-n	15.00	43.90	22.36
+ Syst	36.59	36.59	36.59
+ CoT	33.33	31.71	32.50
+Syst+CoT	19.23	24.39	21.50
Qwen2	P	R	F1
Expan-n	17.24	12.2	14.29
+Syst	12.0	14.63	13.19
+CoT	50.0	7.32	12.77
+Syst+CoT	7.35	12.2	9.17

Table 6 Comparison of the performance ATG using expansion prompting (Expan-n) and varying the prompt content by using a system prompt (+syst) or Chain of Thoughts (+CoT) or both (+syst + CoT). The results are shown for trigrams (timeline test).

As a final experiment, we explore the impact of the prompt based on its structure. We contrast the usage of the standard prompt versus using a system prompt which contains system instructions that are separated from the main prompt. We also use chain-of-thoughts (CoT) prompting as de-

scribed in (Kojima et al., 2022) and which consists in adding the sentence: "*Let's think step by step.*". Finally, we combine system and CoT parameters (+Syst+Cot). The results are reported in Table 6. Overall, the structure of the prompt does impact the performance, however, here also it depends on the model and on the settings. Using system prompting improves F1 score for llama3.3, while F1 drops slightly for Qwen2. CoT often improves the precision (See Table 18 for a detailed comparison for all domains).

7 Discussions

7.1 Graph-based Approaches

We can observe the effect of introducing two types of links. In the term prediction task, precision is greatly improved without affecting much the recall. This reflects the fact that integrating the predictions of type 1 and type 2 links cancels out their errors. The key point here is that the dual link types can eliminate random false positives (FPs), as indicated by the precisions in Tables 2 and 3. In the present experiment, link prediction is performed under a transductive setting; it would be worthwhile to conduct in the future term acquisition experiments under an inductive setting, that is, targeting a completely unseen set of words. In such a scenario, link predictions for type 1 and type 2 links – though produced by the same underlying model – are expected to yield different false-positive patterns, because the two link types play distinct roles in the graph and are evaluated in different structural contexts. Combining them as a filter is thus likely to suppress wrong links. In this respect, the proposed method can be regarded as a robust and powerful approach.

Let us now examine contrasted factors. Firstly, as shown in Tables 2 and 3, using FastText does not change the term prediction performance, while it improves link prediction performance as shown in Table 15. Across all settings, link prediction performance improves when FastText embeddings are used. The same trend is also observed for GAT. These results suggest that using word embeddings as node features is effective, especially when exploring terminological graph structure. Secondly, comparing the results of experiments using bigrams only with those using all ngrams up to 3, the results with all ngrams tend to be slightly lower at the term level but generally improve at the link level. This is due to the fact that constructing terms from the

graph becomes more difficult as the diversity of ngrams increases. The improvement in link-level performance is likely to be due to the increase in the amount of training data. Finally, we discuss terminology acquisition, which is the major theme of this study. Timeline splitting in our experiments is expected to be more representative of real-world term acquisition than random splitting. For the baseline GAT, performance decreases under timeline splitting in all settings except for precision in Dentistry. In contrast, our proposed method yields stable performance regardless of the splitting methods. Our method is robust for term acquisition in real-world situation. The link-level results show that timeline splitting produces lower performance, which indicates that simple graph-based link prediction is not sufficient for term acquisition.

7.2 LLMs

It has been shown that prompt instruction design, as well as the chosen few-shot examples and their order highly impact LLM performance (Gao et al., 2021; Liu et al., 2022; Min et al., 2022; Lu et al., 2022). If these findings mainly concern classification tasks (Lu et al., 2022), they may also apply to ATG task¹⁴. Independently from the prompt design, paraphrasing or using automatic prompt generation methods (Shin et al., 2020; Kong et al., 2024; Deng et al., 2022), the current performance limitations mainly come from the task itself. ATG assumes a list of terms that at least reflects the standard of a given domain. Unless finetuned for term generation, and as shown in our experiments, LLMs tend to always generate a much smaller subset of terms, or a large subset but mainly not specific to the terminology domain. Our LLM evaluations which covered open question and constituent-based few-shot, still exhibit lower results than our proposed GNN approach. Overall, GNN is much faster to train and less costly for generation and it is thus more suitable than LLMs for ATG.

8 Conclusions

This paper introduced a new approach, i.e. automatic term generation (ATG), to the automatic term acquisition (ATA) task. Theoretically, it is understood that term formation is bound by the structure of terminologies. Practically, one of the major merit of ATA is to assist terminologists in augment-

¹⁴See Table 18 for a comparison between several prompt structures.

ing terminologies. It is thus essential to explore concrete terminologies to model the generation of new terms, which has been addressed little so far.

The main methodological contributions of our work are three. First, we defined the terminological graph with two types of links and embeddings attached to the nodes. This enables us to fully explore the terminological structure for ATG as a link prediction. Second, we applied an advanced graph neural network methods to the ATG task over the graph and showed high performance. Third, we introduced timeline information in the experiments, the setup which is closer to the real-world scenario, and showed that our approach is robust in this scenario. Our work, which explored the full structure contained in terminologies, provides the basic point of reference for further advancement of ATA research.

Limitations

Some technical issues and practical factors are left unaddressed in this work. In relation to the construction of terminological graphs, two main issues were left unaddressed in the future. First, we have not introduced the weight to links or allowed multiple links of the same type between the nodes. As this is the first full exploration of the structure of terminologies via graph representation, we retained the essential information and safely left the examination of these possibilities to our future work. Second, we did not apply ambiguity resolution in identifying the intra-term dependency links for complex terms with more than three constituents. Here, we simply made links between adjacent constituents. The resolution routine will be incorporated at the next stage.

As for the evaluations, we could not report the detailed diagnosis of the performance in terms of such factors as frequency of occurrences of constituent elements (nodes) or of such graph features as centrality of nodes or links. Also, it would be important to establish a standard framework for evaluation. From the practical point of view, we assumed the existence of initial terminologies with reasonable size. For minor domains, we would have to start from first collecting an initial set of terms. For smaller languages for which not many terminologies exist, we would have to extend the bilingual or multilingual extension task, which will be a different task from our work here.

As for LLMs evaluation, we only evaluated open-

source models under prompt engineering method. While few-shot learning has shown substantial improvement in classification tasks, in our term generation task however, the results were not satisfactory. We however, expect better results if LLMs are fine-tuned for term generation. We let this experiment for our future work.

Acknowledgments

This work is partly supported by JSPS S20110, JSPS 24H00736 and by Cistate Solutions. We would like to thank Dr. Rei Miyata of the University of Tokyo and Dr. Shuntaro Yada of Tsukuba University for fruitful discussions.

References

- Abdulmohsen Al-Thubaity. 2025. A Novel Dataset for Arabic Domain Specific Term Extraction and Comparative Evaluation of BERT-Based Models for Arabic Term Extraction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(9):1–12.
- Steven Bird and Edward Loper. 2004. [NLTK: The Natural Language Toolkit](#). In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.
- Georgeta Bordea, Paul Buitelaar, and Tamara Polajnar. 2013. Domain-Independent Term Extraction through Domain Modelling. In *Terminology and Artificial Intelligence*, pages 61–68, Paris.
- Farid Cerbah. 2000. Exogeneous and Endogeneous Approaches to Semantic Categorization of Unknown Technical Terms. In *COLING 2000*, pages 145–151, Saarbrücken.
- Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. [RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Helmut Felber. 1984. *Terminology Manual*. Unesco and Inforterm, Paris.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. [Making Pre-trained Language Models Better Few-shot Learners](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*

- and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3816–3830, Online. Association for Computational Linguistics.
- Kris Heylen and Dirk De Hertog. 2015. Methods of Automatic Term Extraction. In *Handbook of Terminology*, pages 203–221, Amsterdam. John Benjamins.
- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D’Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axcil Polleres, Sabbir M. Rashid, Anisa Rula, and Lukas Schmelzeisen. 2022. Knowledge Graphs. *ACM Computing Surveys*, 54(4):1–37.
- Miki Iwai, Koichi Takeuchi, Kyo Kageura, and Kazuya Ishibashi. 2016. A Method of Augmenting Bilingual Terminology by Taking Advantage of the Conceptual Systematicity of Terminologies. In *Computerm 2016*, pages 30–40, Osaka.
- Xuhui Jiang, Chengjin Xu, Yinghan Shen, Xun Sun, Luminyuan Tang, Saizhuo Wang, Zhichao Shi, Baozhu Shang, Zhongwu Chen, Yuanzhuo Wang, and Jian Guo. 2025. [On the Evolution of Knowledge Graphs: A Survey and Perspective](#). ArXiv:2310.04835v3 [cs.AI].
- Kyo Kageura. 2012. *The Quantitative Analysis of the Dynamics and Structure of Terminologies*. John Benjamins, Amsterdam.
- Kyo Kageura. 2022. Terminological Growth. In *Theoretical Perspectives on Terminology*, pages 457–475, Amsterdam. John Benjamins.
- Kyo Kageura. 2024. Automatic term processing in the context of translation: Theoretical and practical issues and prospects. In *Frontiers in Terminology*, pages 337–368, Harbin. Heilongjiang University Press.
- Kyo Kageura and Bin Umino. 1996. Methods of automatic term extraction: A review. *Terminology*, 3(2):259–289.
- Thomas N. Kipf and Max Welling. 2016. [Variational Graph Auto-Encoders](#). arXiv:1611.07308 [stat.ML].
- Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large Language Models are Zero-Shot Reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Weize Kong, Spurthi Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. [PRewrite: Prompt Rewriting with Reinforcement Learning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–601, Bangkok, Thailand. Association for Computational Linguistics.
- Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. [Syntactic Annotations for the Google Books NGram Corpus](#). In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174, Jeju Island, Korea. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. [What Makes Good In-Context Examples for GPT-3?](#) In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. [Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. *Aspec: Asian scientific paper excerpt corpus*. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Christopher Norman and Akiko Aizawa. 2015. [Technical term extraction using measures of neology](#). In *Proceedings of the ACL 2015 Workshop on Novel Computational Approaches to Keyphrase Extraction*, pages 2–9, Beijing, China. Association for Computational Linguistics.
- Juan Carlos Sager. 1990. *A Practical Course in Terminology Processing*. John Benjamins, Amsterdam.
- Koichi Sato, Koichi Takeuchi, and Kyo Kageura. 2013. Terminology-driven augmentation of bilingual terminologies. In *MT Summit XIV*, pages 3–10, Nice.
- Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607. Springer International Publishing.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Takoua Ben Smida, Riadh Bouslimi, and Hadhemi Achour. 2025. A comprehensive survey on link prediction: From heuristics to graph transformers. *The Journal of Supercomputing*, 81(1):1–42.

Hanh Thi-Hong Tran, Carlos-Emiliano González Gallardo, Antoine Doucet, and Senja Pollak. 2025. Llamate: Automated terminology extraction using large-scale generative language models. *Terminology*, 31(1):5–36.

Hanh Thi Hong Tran, Matej Martinc, Jaya Caporusso, Antoine Doucet, and Senja Pollak. 2023. [The recent advances in automatic term extraction: A survey](#). *Computer Science: Computation and Language*, arXiv:2301.06767v1.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. [Complex embeddings for simple link prediction](#). *Preprint*, arXiv:1606.06357.

Petar Velickovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *6th International Conference on Learning Representations*.

Kang Xu, Yifan Feng, Qiandi Li, Zhenjiang Dong, and Jianxiang Wei. 2025. Survey on terminology extraction from texts. *Journal of Big Data*, 12(29):1–40.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. [Embedding entities and relations for learning and inference in knowledge bases](#). ArXiv:1412.6575 [cs.CL].

A Motivation behind using Google book ngrams

Google book ngrams is a very large ngram dataset (See Table 7) which was extracted from a massive

collection of digital books. Covering a period of time ranging from the 17 century to July 2020¹⁵, It is a valuable resource for studying word frequency evolution over time. By using this dataset, however, it is not possible to find with certitude the first appearance of words. Nonetheless, time stamp information can serve to estimate the periods in which they were the most popular. We consider the first peak in usage of terms as the period in which a term is effectively adopted as such (Norman and Aizawa, 2015). To estimate the first appearance of a term, we first calculate the mean frequency of terms of the domain, and then we select the year in which the frequency of a term is the closest to the mean. Our search interval ranges from the beginning of the period until the year of the first peak (popularity).

Figures 4 and 5 show the distribution of unigrams and trigrams according to the timeline. Table 8 shows the basic statistics of the four refined terminologies by using ASPEC Corpus, and those that are further filtered by matching Google Books Ngram to assign time stamps.

ngram	#files	size
n = 1	24	13G
n = 2	589	302G
n = 3	6,881	2T
n = 4	6,668	2T
n = 5	19,423	10T

Table 7 Google Books Ngram Statistics for English. The size is given for compressed files in gzip.

B Information About Data Split

Tables 9 to 14 show additional statistics about the data split adopted in our experiments.

C Hyperparameters

C.1 R-GCNs (Ours)

We set the hyperparameters of the R-GCNs as follows: dropout to 0.2. We used a hidden dimension of 64 and an embedding dimension of 32, and trained the model with a learning rate of 0.01. We trained the model using full-batch learning for up to 1,000 epochs.

C.2 GAT Baseline

We set the hyperparameters of GAT as follow: hidden dimension to 256, embedding dimension to 64,

¹⁵The latest version up to date.

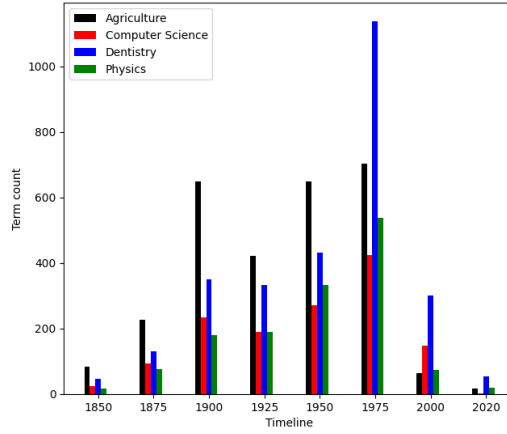


Figure 4: Term distribution of unigrams based on their first time appearance in google book ngrams for: Agriculture, Computer Science, Dentistry and Physics.

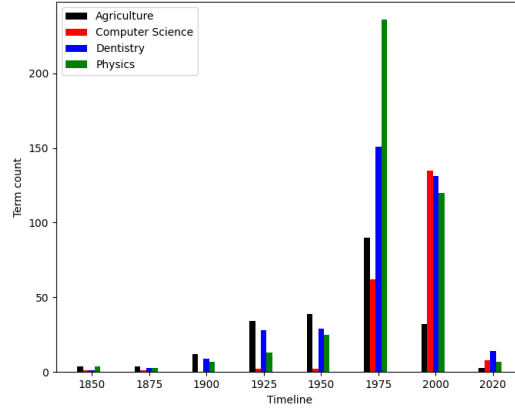


Figure 5: Term distribution of trigrams based on their first time appearance in google book ngrams for: Agriculture, Computer Science, Dentistry and Physics.

	ASPEC Corpus		$\cap$ Termbank (#Terms)					$\cap$ Google Books Ngram				
Domain	#sentence	#Token	1	2	3	4	all	1	2	3	4	all
Agr	56,722	1,353,956	2,896	2,725	270	30	5,921	2,718	2,333	212	-	5,263
CS	38,659	837,476	1,457	1,952	307	20	3,736	1,334	1,636	205	-	3,175
Den	284,277	6,250,158	2,816	3,022	427	51	6,316	2,673	2,623	345	-	5,641
Phy	149,406	3,287,778	1,457	3,558	466	52	5,533	1,379	3,064	402	-	4,845

Table 8 Dataset statistics (English) based on four terminology domains (Agr: Agriculture, CS: Computer Science, Den: Dentistry and Phy: Physics).

number of attention heads to 4 and dropout to 0.2. We run the training for 300 epochs using a learning rate of 0.001 and a batch size of 8.

D Additional Results

Tables 15 and 16 show the results of experiments for different setups with different sets of evaluation metrics. These tables show additional information

about the performance of our proposed methods.

E Terminologies of the Four Domains

Japanese Ministry of Education, and Japanese Institute of Landscape Architecture. 1986. *Japanese Scientific Terms: Agricultural Sciences*. JSPS, Tokyo.

Aiso, Hideo. 1993. *Terminological Lexicon of Information Processing*. Tokyo: Ohm.

ngram	Agriculture				Computer				Dentistry				Physics			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
1	2,896	2,027	290	579	1,429	1,000	143	286	2,816	1,971	282	563	1,457	1,020	146	291
2	2,725	1,908	272	545	1,952	1,366	195	391	3,022	2,115	302	605	3,558	2,491	356	711
3	270	189	27	54	307	215	31	61	427	299	43	85	466	326	47	93
4	30	21	3	6	20	14	2	4	51	36	5	10	52	36	5	11
all	5,921	4,145	592	1,184	3,708	2,595	371	742	6,316	4,421	632	1,263	5,533	3,873	554	1,106

Table 9 Train, validation and test statistics for term-level (Random Split) in four domains (Agriculture, Computer Science, Dentistry and Physics).

ngram	Agriculture				Computer				Dentistry				Physics			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
1	2,718	1,903	272	543	1,334	934	133	267	2,673	1,871	267	535	1,379	965	138	276
2	2,333	1,633	233	467	1,636	1,145	164	327	2,623	1,836	262	525	3,064	2,145	306	613
3	215	149	22	44	208	145	21	42	348	243	35	70	405	282	41	82
all	5,266	3,685	527	1,054	3,178	2,224	318	564	5,644	3,950	564	1,130	4,848	3,392	485	971

Table 10 Train, validation and test statistics for term-level (Timeline Split) in four domains (Agriculture, Computer Science, Dentistry and Physics).

link type	Agriculture				Computer				Dentistry				Physics			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
type 1	2764	1938	276	550	1974	1380	198	396	3085	2165	305	615	3680	2584	369	727
type 2	2723	1908	272	543	1948	1363	195	390	3021	2114	302	605	3544	2482	355	707
all	5487	3846	548	1093	3922	2743	393	786	6106	4279	607	1220	7224	5066	724	1434

Table 11 Train, validation and test statistics for link-level (Bigram/Random Split) in four domains (Agriculture, Computer Science, Dentistry and Physics).

link type	Agriculture				Computer				Dentistry				Physics			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
type 1	3181	2248	311	622	2362	1697	233	432	3698	2630	361	707	4289	3060	417	812
type 2	5867	4115	586	1166	3630	2551	360	719	6230	4366	621	1243	5426	3809	542	1075
all	9048	6363	897	1788	5992	4248	593	1151	9928	6996	982	1950	9715	6869	959	1887

Table 12 Train, validation and test statistics for link-level (All/Random Split) in four domains (Agriculture, Computer Science, Dentistry and Physics).

link type	Agriculture				Computer				Dentistry				Physics			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
type 1	2333	1633	233	467	1639	1148	164	327	2625	1838	262	525	3066	2147	306	613
type 2	2333	1633	233	467	1636	1145	164	327	2623	1836	262	525	3064	2145	306	613
all	4666	3266	466	934	3275	2293	328	654	5248	3674	524	1050	6130	4292	612	1226

Table 13 Train, validation and test statistics for link-level (Bigram/Timeline Split) in four domains (Agriculture, Computer Science, Dentistry and Physics).

Japanese Ministry of Education, and Japanese Association for Dental Science. 1992. *Japanese Scientific Terms: Odontology*. Japanese Association for Dental Science, Tokyo.

Japanese Ministry of Education, and the Physical Society of Japan. 1990. *Japanese Scientific Terms: Physics*. Baifukan, Tokyo.

link type	Agriculture				Computer				Dentistry				Physics			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
type 1	2634	1847	261	526	1883	1328	183	372	3104	2195	305	604	3567	2508	354	705
type 2	5251	3676	525	1050	3144	2202	315	627	5617	3935	561	1121	4810	3380	478	952
all	7885	5523	786	1576	5027	3530	498	999	8721	6130	866	1725	8377	5888	832	1657

Table 14 Train, validation and test statistics for link-level (All/Timeline Split) in four domains (Agriculture, Computer Science, Dentistry and Physics).

Methods	Proposed-Method (RGCN)											
	Random list											
	Agriculture			Computer			Dentistry			Physics		
	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc
exp1 bigram	66.79	75.05	53.68	65.62	72.79	52.74	65.04	75.52	54.32	75.37	80.52	68.30
exp1 bigram (embed)	79.18	83.37	72.59	80.25	80.61	75.00	75.70	80.60	66.78	84.46	84.31	78.96
exp2 bigram	90.90	89.71	83.34	89.07	87.68	81.23	88.78	90.48	80.22	92.81	92.78	86.20
exp2 bigram (embed)	94.47	94.61	87.14	91.18	89.29	83.90	94.01	94.21	89.00	94.50	94.77	88.74
exp1 all	58.50	68.83	54.01	67.97	73.24	55.76	62.93	75.31	59.15	74.98	81.00	62.69
exp1 all (embed)	69.51	74.76	62.48	76.63	77.04	70.39	70.43	78.30	62.44	76.23	79.74	70.41
exp2 all	84.87	86.99	73.05	85.69	86.15	77.12	89.11	91.03	73.30	90.21	91.03	81.72
exp2 all (embed)	85.73	88.42	73.86	87.72	87.80	78.35	90.48	92.05	74.84	91.81	93.06	84.47
	Timeline list											
exp1 bigram	52.57	62.90	60.15	56.88	62.91	48.24	60.61	67.99	62.16	64.37	71.18	52.37
exp1 bigram (embed)	66.92	69.79	62.76	77.65	70.74	67.20	76.46	77.87	64.71	83.04	81.21	76.34
exp2 bigram	86.58	83.66	73.45	85.62	82.57	78.44	86.98	86.84	74.62	90.37	89.32	82.00
exp2 bigram (embed)	90.13	88.11	81.05	89.45	84.81	80.50	92.81	91.53	87.13	93.24	91.68	85.43
exp1	57.44	64.29	53.67	60.83	62.27	49.64	60.38	68.99	59.54	71.29	75.02	57.72
exp1 (embed)	57.15	61.70	51.98	64.36	64.47	60.05	59.34	66.27	52.89	68.68	69.09	60.66
exp2	80.41	81.47	66.56	82.92	80.19	72.09	88.94	89.96	86.41	87.83	87.49	76.88
exp2 (embed)	84.00	83.47	66.00	85.31	81.60	73.21	89.45	89.80	85.90	90.06	90.12	79.11

Table 15 Results for link-level evaluation (AUC, AP, Acc) in four domains (Agriculture, Computer Science, Dentistry and Physics). Results presented above are based on random split, results presented below are based on temporal split.

Method	Type	Agriculture				
		P	R	F1	AUC	AP
exp1	overall	59.52	66.78	62.94	58.50	68.83
	Type 1	54.74	60.29	57.38	58.64	68.14
	Type 2	62.00	70.24	65.86	57.49	68.99
exp2	overall	83.38	67.34	74.50	84.87	86.99
	Type 1	83.46	88.42	85.87	91.50	92.08
	Type 2	83.31	56.09	67.04	82.33	85.25
exp1 (Embed)	overall	76.16	52.18	61.93	69.51	74.76
	Type 1	75.49	55.47	63.95	69.24	73.35
	Type 2	76.56	50.43	60.81	69.24	75.61
exp2 (Embed)	overall	88.72	63.37	73.93	85.73	88.42
	Type 1	88.87	82.15	85.38	91.00	91.51
	Type 2	88.60	53.34	66.60	83.36	87.26

Method	Type	Computer				
		P	R	F1	AUC	AP
exp1	overall	56.38	83.32	67.25	67.97	73.24
	Type 1	53.20	75.00	62.25	66.58	71.58
	Type 2	58.15	88.32	70.13	67.45	73.79
exp2	overall	85.31	70.11	76.97	85.69	86.15
	Type 1	84.63	82.87	83.74	90.06	88.33
	Type 2	85.85	62.45	72.30	83.75	85.34
exp1 (embed)	overall	79.82	61.16	69.26	76.63	77.04
	Type 1	77.41	65.05	70.69	75.29	73.36
	Type 2	81.50	58.83	68.34	77.24	79.01
exp2 (embed)	overall	88.13	69.68	77.83	87.72	87.80
	Type 1	87.37	78.47	82.68	90.05	87.33
	Type 2	88.70	64.39	74.62	86.27	88.28

Method	Type	Dentistry				
		P	R	F1	AUC	AP
exp1	overall	81.60	43.44	56.69	62.93	75.31
	Type 1	81.64	53.47	64.62	66.31	76.79
	Type 2	81.57	37.73	51.60	62.72	75.49
exp2	overall	86.65	66.92	75.52	89.11	91.03
	Type 1	86.63	87.98	87.30	91.50	92.30
	Type 2	86.68	54.95	67.26	87.76	90.44
exp1 (embed)	overall	76.87	55.74	64.63	70.43	78.30
	Type 1	76.23	57.14	65.32	71.03	79.42
	Type 2	77.26	54.95	64.22	69.71	78.02
exp2 (embed)	overall	90.46	66.10	76.39	90.48	92.05
	Type 1	90.69	79.92	84.96	93.05	93.36
	Type 2	90.27	58.25	70.81	89.11	91.47

Method	Type	Physics				
		P	R	F1	AUC	AP
exp1	overall	62.00	93.48	74.56	74.98	81.00
	Type 1	60.83	90.27	72.68	73.46	80.69
	Type 2	62.87	95.91	75.95	75.34	81.09
exp2	overall	82.77	86.80	84.74	90.21	91.03
	Type 1	81.64	95.81	88.16	92.73	92.89
	Type 2	83.82	80.00	81.87	89.14	90.02
exp1 (embed)	overall	80.98	64.55	71.84	76.23	79.74
	Type 1	82.62	68.47	74.88	78.00	82.12
	Type 2	79.66	61.58	69.46	74.92	78.13
exp2 (embed)	overall	87.75	85.37	86.54	91.81	93.06
	Type 1	86.41	93.97	90.03	93.86	94.47
	Type 2	88.98	78.88	83.63	90.76	92.29

Table 16 Results by type for link-level evaluation (P, R, F1, AUC, AP) in four domains (Agriculture, Computer Science, Dentistry and Physics). Results presented above are based on random split for all ngram evaluation.

Model	Agriculture								
	Unigram			Bigram			Trigram		
	P	R	F1	P	R	F1	P	R	F1
llama3.1:8b	99.46	34.07	50.75	0.0	0.0	0.0	7.14	2.33	3.51
llama3.1:70b	18.01	41.99	25.21	4.0	5.57	4.66	15.05	32.56	20.59
llama3.3	40.83	32.78	36.36	5.78	8.14	6.76	14.58	16.28	15.38
Qwen2	65.38	3.13	5.97	4.76	0.21	0.4	11.11	2.33	3.85
Gemma2	99.7	61.88	76.36	25.0	0.21	0.42	1.41	2.33	1.76
Mistral	99.75	72.38	83.89	7.04	1.07	1.86	16.67	4.65	7.27

Model	Computer								
	Unigram			Bigram			Trigram		
	P	R	F1	P	R	F1	P	R	F1
llama3.1	99.61	96.63	98.1	14.36	8.56	10.73	20.0	26.83	22.92
llama3.1:70b	92.02	73.41	81.67	12.66	21.71	15.99	32.35	26.83	29.33
llama3.3	95.51	55.81	70.45	30.17	16.51	21.34	38.71	29.27	33.33
Qwen2	100.0	85.77	92.34	16.4	12.54	14.21	15.38	14.63	15.0
Gemma2	99.62	98.88	99.25	46.15	3.67	6.8	27.78	12.2	16.95
Mistral	100.0	99.25	99.62	16.91	14.37	15.54	80.0	9.76	17.4

Model	Dentistry								
	Unigram			Bigram			Trigram		
	P	R	F1	P	R	F1	P	R	F1
llama3.1	100.0	58.13	73.52	7.73	3.43	4.75	6.45	5.8	6.11
llama3.1:70b	54.76	4.3	7.97	12.77	1.14	2.09	60.0	8.7	15.2
llama3.3	97.67	7.85	14.53	14.29	0.57	1.1	42.86	13.04	20.0
Qwen2	20.0	0.37	0.73	21.43	0.57	1.11	16.67	2.9	4.94
Gemma2	89.47	3.18	6.14	18.75	0.57	1.11	5.88	1.45	2.33
Mistral	57.14	6.73	12.04	14.52	1.71	3.06	12.5	4.35	6.45

Model	Physics								
	Unigram			Bigram			Trigram		
	P	R	F1	P	R	F1	P	R	F1
llama3.1	100.0	99.28	99.64	0.0	0.0	0.0	25.0	3.7	6.45
llama3.1:70b	99.64	99.28	99.46	14.92	15.99	15.44	17.86	6.17	9.17
llama3.3	97.51	99.28	98.39	16.53	13.21	14.68	15.62	18.52	16.95
Qwen2	99.64	99.64	99.64	25.08	12.4	16.6	18.18	2.47	4.35
Gemma2	99.63	97.1	98.35	32.57	16.15	21.59	50.0	2.47	4.71
Mistral	99.64	99.64	99.64	30.53	13.05	18.28	5.41	2.47	3.39

Table 17 Automatic term generation results for LLMs using **Expansion Prompting** in the Timeline test setting with few-shot.

	Agriculture			Computer			Dentistry			Physics		
	Unigram											
llama3.3	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Expan-n	7.37	24.01	11.28	92.92	68.88	79.11	25.46	39.61	31.0	90.91	3.44	6.63
+Syst	7.56	23.83	11.48	96.12	69.23	80.49	36.75	40.14	38.37	100.0	3.78	7.28
+CoT	8.88	26.94	13.36	87.39	72.73	79.39	31.23	30.73	30.98	92.89	67.35	78.08
+Syst+CoT	12.08	28.84	17.03	91.28	69.58	78.97	19.17	38.37	25.57	97.37	50.86	66.82
	Bigram											
llama3.3	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Expan-n	1.12	0.73	0.88	4.75	12.28	6.85	0.0	0.0	0.0	16.51	9.7	12.22
+Syst	3.6	6.42	4.61	4.18	2.56	3.18	0.0	0.0	0.0	24.06	11.67	15.72
+CoT	3.22	5.69	4.11	4.55	2.81	3.47	0.0	0.0	0.0	36.36	4.5	8.01
+Syst+CoT	3.63	6.06	4.54	4.18	4.6	4.38	0.0	0.0	0.0	14.29	9.56	11.46
	Trigram											
llama3.3	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Expan-n	12.12	7.41	9.2	50.0	3.28	6.16	5.0	1.18	1.91	28.57	12.9	17.77
+Syst	9.84	11.11	10.44	57.14	6.56	11.77	0.88	1.18	1.01	25.53	12.9	17.14
+CoT	7.02	7.41	7.21	30.0	4.92	8.45	16.67	2.35	4.12	57.14	4.3	8.0
+Syst+CoT	4.0	5.56	4.65	19.44	11.48	14.44	19.57	10.59	13.74	20.41	10.75	14.08
	Agriculture			Computer			Dentistry			Physics		
	Unigram											
Qwen2	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Expan-n	10.67	1.38	2.44	28.0	2.45	4.51	31.25	0.89	1.73	81.82	6.19	11.51
+Syst	36.84	1.21	2.34	33.33	2.1	3.95	85.71	5.33	10.04	78.26	6.19	11.47
+CoT	27.94	3.28	5.87	100.0	100.0	100.0	0.0	0.0	0.0	100.0	100.0	100.0
+Syst+CoT	20.0	0.35	0.69	99.65	99.65	99.65	0.0	0.0	0.0	100.0	100.0	100.0
	Bigram											
Qwen2	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Expan-n	10.53	0.37	0.71	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
+Syst	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.25	0.14	0.27
+CoT	8.0	0.37	0.71	0.0	0.0	0.0	0.0	0.0	0.0	23.43	14.21	17.69
+Syst+CoT	17.65	0.55	1.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Trigram											
Qwen2	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Expan-n	0.0	0.0	0	14.29	1.64	2.94	4.55	1.18	1.87	0.0	0.0	0.0
+Syst	0.0	0.0	0	8.33	1.64	2.74	6.67	1.18	2.01	0.0	0.0	0.0
+CoT	0.0	0.0	0	21.43	4.92	8.0	0.0	0.0	0.0	0.0	0.0	0.0
+Syst+CoT	0.0	0.0	0	15.38	3.28	5.41	0.0	0.0	0.0	8.0	4.3	5.59

Table 18 Comparison of the performance of ATG for llama3.3 and Qwen2 using expansion prompting (**Expan-n**) and varying the prompt content by adding a system prompt (+Syst) or Chain-of-Thoughts (+CoT) or both (+Syst+CoT). The results are shown for trigrams (Random test).

OpenQ-0	Prompt: Generate a list of trigram terms in the Computer Science domain ranging from 50 to 500 terms. The list of terms should be separated by a comma, nothing else print only the terms. Answer:
OpenQ-0 (+Syst)	System prompt: You are an expert in term generation. Your task is to generate a comprehensive list of terms in a specific domain. The list of terms should be separated by a comma, nothing else print only the terms. Prompt Generate a list of trigram terms in the Computer Science domain ranging from 50 to 500 terms. The list of terms should be separated by a comma, nothing else print only the terms. Answer:
OpenQ-0 (+CoT)	Prompt: Generate a list of trigram terms in the Computer Science domain ranging from 50 to 500 terms. The list of terms should be separated by a comma, nothing else print only the terms. Answer: Let's think step by step.
OpenQ-0 (+Syst+CoT)	System prompt: You are an expert in term generation. Your task is to generate a comprehensive list of terms in a specific domain. The list of terms should be separated by a comma, nothing else print only the terms. Prompt: Generate a list of trigram terms in the Computer Science domain ranging from 50 to 500 terms. The list of terms should be separated by a comma, nothing else print only the terms. Answer: Let's think step by step.
OpenQ-n	Prompt: Based on a list of term examples of the Computer Science domain, generate a list of trigram terms ranging from 50 to 500 terms that are similar to the list of examples. The list of terms should be separated by a comma, nothing else print only the terms. List of term examples: standard data format, shared memory type, higher order logic, random access machine, remote procedure call, data link layer, liquid crystal device, random number generation, virtual memory management, statistical pattern recognition, parallel processing algorithm, remote sensing image, parallel computation model, round robin scheduling, pseudo random number, massively parallel computer, adaptive digital filter, electronic bulletin board, network management function, pyramid data structure. Answer:
Expan-1	Prompt: Given a list of words, generate the corresponding list of terms that belong to the Computer Science domain. The list of terms should be separated by a comma, nothing else print only the terms. Here are some examples of a List of words and their corresponding Terms. List: machine, access, computation, pyramid, model, random, type, pseudo, data, memory, management, structure, function, shared, parallel, number, network. Answer: network management function, parallel computation model, pseudo random number, pyramid data structure, random access machine, shared memory type. Generate the terms of the Computer Science domain of the following list. List: auditor, filter, descent, line, wide, transfer, speech, computer, connected, extraction, distributed, search, design, diagnosis, architecture, interface, stereo, server, supply, estimation, data, power, language, genome, set, development, access, maximum, markov, polynomial, feature, bulletin, servo, binocular, histogram, structure, mode, band, image, gray, mail, natural, user, least, resource, management, flow, area, system, hidden, description, support, standard, asynchronous, recognition, vision, component, information, level, small, project, hardware, herbrand, distribution, right, steepest, normal, graph, document, likelihood, service, human, continuous, process, parallel, software, board, intellectual, model, business, property, pass, uninterruptible, relational, time, mechanism, visual, positioning, instruction, analysis, method, network, electronic, processing, control, algorithm, client, graphical, query, global, strongly. Answer:

Table 19 Examples for open question and expansion based prompting. **Prompt** indicates the main prompt while **System Prompt** indicates the system instruction provided before the main prompt. We illustrate both settings in OpenQ-0 and OpenQ-0 (+Syst). For chain of thoughts, the sentence "*Let's think step by step.*" is added at the end of the prompt. This is shown in OpenQ-0(+CoT). Combining system prompt and chain of thought is given in OpenQ-0 (+Syst+CoT). OpenQ-n is n-shot example learning. Finally, for expansion prompting, we report Expan-1 which refers to expansion with 1-shot example represented by **List**. For n-shot, we provide n lists and their corresponding Answers before asking the LLM on the test list. We do not report OpenQ-n and Expan-1 with system prompt (+Syst) and chain of thoughts (+CoT) but the format is similar to the illustrations in OpenQ-0 with +Syst and +CoT.